OMNIDISK Unit

Multi-Function DMA Disk Controller

for IEEE-696/S-100

Technical Reference Manual

January 1986

Revised November 2022

----------------------------------------------------------------

Copyright (C) 1985 by Fulcrum Computer Products

All rights reserved.

----------------------------------------------------------------

No part of this publication may be reproduced, transmitted,

transcribed, stored in a retrieval system, or translated into

any language, in any form or by any means, electronic, mechanical,

magnetic, optical, chemical, without the prior written permission

of Fulcrum Computer Products.

-----------------------------------------------------------------

DISCLAIMER

This manual contains information which is subject to change without

notice. Fulcrum Computers, Inc. assumes no responsibility for any

errors or omissions which may be contained within this manual.

-----------------------------------------------------------------

OMNIDISK Board

Technical Reference Manual

Table of Contents

1. INTRODUCTION

1.1. Conventions ......................................... 1

1.2. General Theory ...................................... 1

1.3. Hardware Overview / Features ........................ 5

1.4. Quick Integration Steps ............................. 7

1.5. Skew / Interleave ................................... 8

1.6. Track Buffering......................................11

1.7. Disk Drives Floppy and Winchester ...................12

1.8. Data Transfer - Processor vs. DMA ...................12

2. INSTALLATION

2.0. Preparation for use / conventions ..................14

2.1. Unpacking and Inspection ...........................14

2.2 Seating the Board ..................................15

2.3. Hardware Options ...................................16

2.3.1. Factory Jumpering .............................16

2.3.2. Boot Drive Type ...............(S1)............16

2.3.3. Port Address Selection ........(S2-3,J8).......17

2.3.4. Extended TMA Addressing .......(J6)............18

2.3.5. Interrupts ....................(J7)............18

2.3.6. Boot PROM Enable ..............(J9 - upper)....19

2.3.7. Write Precompensation .........(J9 - lower)....20

2.4. Port Definition and Use ............................20

2.5. DMA usage ..........................................20

2.6. Cabling ............................................20

3. FIRMWARE FEATURES AND FACILITIES

3.0. Abstract ............................................27

3.1. Versions and History ................................27

3.2. Concept of Logical Device Sets ......................27

3.3. Command - types and usage ...........................28

3.4. Booting - via onboard PROM bootstrap ................??

3.5. Boot Interface with FULCRUM MPU-Z ...................??

3.6. Booting - other cpu considerations ..................??

3.7. Known Bugs ..........................................??

3.8. Writing your own drivers ............................??

4. HOST OPERATING SYSTEM ENVIRONMENT INTERFACE

4.0. Preface .............................................64

4.1. CP/M-80 .............................................65

4.2 CP/M Plus (CP/M 3.1) ................................69

4.3. MP/M-80 .............................................69

4.4. CP/M-86 .............................................69

4.5. MP/M-86 .............................................69

4.6. MS-DOS ..............................................70

4.7. Concurrent PC-DOS or CCP/M ..........................74

4.8. CP/M-68k ............................................74

4.9. 68000 XENIX .........................................74

5. ERROR MESSAGE INTERPRETATION

5.0. Abstract.............................................75

5.1. Floppy Drive.........................................75

5.2. Winchester Hard Disk ................................78

5.3. Other ...............................................79

6. DIAGNOSTICS/MAINTENANCE

6.0. Abstract ............................................80

6.1. Hardware Diagnostics, a Flow-charted Summary ........80

7. ENGINEERING SPECIFICATIONS

7.1. Physical Characteristics ............................81

7.2. Environmental Characteristics .......................81

7.3. Power Requirements ..................................81

7.4. Bus interface specifications (IEEE/696)..............81

7.5. Winchester Interface specifications .................86

7.6. 5.25" Floppy Interface Specifications ...............90

7.7. 8" Floppy Interface Specifications ..................91

7.8. Theory of Operation .................................92

7.8.1. Overview ..........................................92

7.8.2. Onboard MicroProcessor ............................92

7.8.3. Floppy Interface Operation ........................95

7.8.4. Hard Disk Interface Operation .....................95

7.8.5. DMA operation .....................................95

7.8.6. IEEE-696 Bus interface ............................96

8. SELECTED DRIVE SPECIFICATIONS

8.0. Abstract ............................................98

8.1. 5.25" Floppy Drive Configuration ....................98

8.2. 8" Floppy Drive Configuration .......................99

8.3. Winchester Drive Configuration .....................105

9. UTILITY PROGRAMS

9.2. SSKEW - A skew change facility ....................107

9.3. BOOT - Simulate COLD boot ........................107

9.4. TMOV - Verify and test correct operation of TMA ..107

9.5. BPROM - Operation of Boot PROM firmware ...........108

9.6. COPY - Disk to disk copy program .................109

9.7. MCCONV - Provides access to MSDOS files from CPM ...109

9.8. FMT3 - Format device .............................110

9.9. SYSGEN - A system track read/write utility .........110

9.10 DPB - Examine a drive DPB .......................110

9.11 FORMAT - Format for MS-DOS .........................110

9.1. LU - A logical device reassignment facility ....106

Index - cross referencing ...............................111

List of Tables

1-1 Sector Skew, Standard IBM 8 inch ....................10

1-2 Bit Cell Times ...................................... 4

1-3 DMA vs CPU bus bandwidth ............................13

2-1 Initial Boot Device Type ............................16

2-3 Sample Port Addressing ..............................17

2-6 Interrupt Jumpering .................................18

2-7 Jumper Area J9 ......................................19

3-1 Logical to Physical Mapping, Initial ................21

3-2 Summary of Firmware Commands ........................22

3-3 Defined Physical Formats ............................27

4-1 CP/M Disk Parameter Block ...........................65

4-2 CP/M BSH/BSM/BLS Relations ..........................66

4-3 CP/M Supported Logical Formats - 5 inch .............67

4-4 CP/M Supported Logical Formats - 8 inch .............68

4-5 Sector Allocation in MS-DOS .........................70

4-6 MS-DOS DPB ..........................................71

4-7 MS-DOS Supported Logical Formats - 5 inch ...........72

4-8 MS-DOS Supported Logical Formats - 8 inch ...........73

5-1 General Status Result Description ...................75

7-4 S-100 Buss Pin Designations .........................82

Appendices

A. SCHEMATICS...........................................???

B. Component designations ..............................112

C. Silk Screen Mask for Component Location .............???

D. DMA State Prom Listing ..............................114

E. Firmware, Format, and other sources .................115

F. 8085/Z80 Drivers ....................................118

G. Acknowledgements ....................................???

H. Glossary ............................................???

I. Boot Prom Listings...................................119

1 - INTRODUCTION

This manual contains the information necessary to install and operate the OMNIDISK Multi-Function disk controller in an IEEE 696 environment. Currently available driver software includes MS-DOS, CP/M-80, CP/M+ (CP/M 3), CP/M-86 and MP/M-86. Future releases for CCP/M 4.1 (MS-DOS 2.0 compatible) and XENIX (68000 SCO release) are planned.

1.1 Conventions used through-out the manual

Some of the OMNIDISK signals described in this manual are active-low

(ground true); others are active-high (positive true).

The active-low signals are shown terminated with an asterisk (\*).

The absence of an asterisk indicates that the condition is true when the logic level is high.

Also refer to the glossary for term definitions, found in APPENDIX K.

1.2 General Theory

A floppy/hard disk controller/drive system allows for the storage and retrieval blocks of data between the main system memory and a storage media.

The data on a disk can be conceived of as a circular ring with its center at the physical center of the diskette.

If the read/write head is located a "distance" N from the center of the diskette, then Nth track is defined as the area passing under the head in one complete revolution of the disk.

A floppy disk may have commonly 40 to 80 tracks. Each track consists of a number of "sectors". A sector consists of an ID field, a gap, and a data field.

Several sectors can be contained on a single track. The length of a sector is fixed at the format time to be either 128, 256, 512, or 1024 bytes long.

The number of sectors per track is determined by the length of the sector, mode of recording, the storage media, and the speed of disk rotation.

Data from program files is stored on the disk in as many sectors as is needed to completely store the file, and thus can spread over several tracks.

This is all kept track of by the operating system.

When a diskette is first used, it must be formatted.

All microcomputer systems should have a program named something like "FORMAT" that performs this necessary operation. This operation is necessary because a diskette right out of the box has no meaningful magnetic recording on it at all.

If you buy pre-formatted diskettes you are paying someone else to perform the format for you.

The format operation writes sector ID's and dummy data over the entire floppy disk surface.

All other data transferring operations require that an ID be read before any user data is transferred.

The ID portion of a sector is placed on the diskette during the format phase of diskette preparation and precedes each sector.

The ID field serves to identify the sectors position on the diskette and contains:

1. A fixed pattern to indicate the start of a sector called an address mark

2. The track address

3. The sector address

4. The number of bytes/sector

The data field contains the actual information to be stored.

Also included in the data field are sync bytes (a train of digital zeroes), address marks, and CRC check bytes for error checking.

An inter-sector gap separates the ID field from the data field.

Placing ID fields on the diskette is known as "soft-sectoring".

Most floppy disks used in microcomputer systems are soft-sectored.

Soft-sectored implies that the design of the diskette itself does not force one to use a particular number of sectors or a particular sector length.

Soft-sectored floppies have one index hole.

Contrast this with hard-sectored floppies that have, perhaps, 16 sector holes, evenly spaced, with an extra index hole between two of the sector holes.

These extra holes are detected by the drive/controller electronics to produce sector address information.

In soft sectoring the sectoring information is written magnetically.

This formatting data remains permanently on the diskette or until it is reformatted.

A more exact description of the ID field contents is as follows:

1: A train of sync bytes, all zeroes.

2: An address mark indicating the start of a sector.

3: The track number.

4: The side number.

5: The sector number.

6: A coded value indicating the data field length.

7: Two bytes of CRC data.

General Write Process

Assume that a block of data is present in the OmniDisk to be stored on a disk.

(Previously transfered by the host system)

1) First the correct drive unit is selected, and checked for ready.

2) The head is position over the correct track.

3) ID fields are read until the proper sector ID is found.

4) A gap time is waited.

5) The entire data field is written to the disk.

Encoding methods - Single and Double Density

All information is recorded on the floppy disk medium by means of

magnetic flux reversals. Several techniques are possible for encoding

clock and data information for magnetic media.

The most commonly used for floppy discs are MFM (double density) and

FM (single density).

Both FM and MFM techniques record a data bit 1 as a flux reversal at the center of the "bit cell" time. A data bit of zero is indicated by no flux reversal.

Clock bits are recorded by a flux reversal at the beginning of the bit cell.

FM and MFM techniques differ primarily in their definitions of when the clock bits are to be written.

If we consider that the data on the disk is divided down into bit cells, then the rules for FM or Frequency Modulation encoding are:

1. Write data bits at the center of the bit cell when data bit = 1

2. Always write clock bits at the leading edge of the bit cell.

Examination of the rules show that a penalty is paid in using clock bits with each data bit, and that if the code were made more efficient by elimination of the clocking bits then higher information densities may could be achieved.

And so we get to MFM.

One such technique is called MFM or Modified FM.

This encoding scheme uses a rather clever idea to reduce the clocking bits and increase storage density.

Since there are often long streams of logic 1's in the data stream, it is possible to use data bits themselves to supply clocking.

During the times when there are no logic 1's we can insert a clock bit to maintain synchronization.

Again when we consider the data divided into bit cells on the floppy disk we have the following rules for MFM data encoding:

1. Write data bits at the center of the bit cell

2. Write clock bits at the leading edge of a bit cell if:

2.1. NO data has been written in the previous bit cell and

2.2. NO data bit will be written in the present cell

Note that although the bit cells are twice as small as in the single density FM encoding, the maximum density of the flux reversals is identical to that achieved by FM encoding, since now we write more real data per unit time.

In the single density FM mode, the recording frequency is 125Khz or

8us per bit cell.

In the double density MFM mode, the recording frequency is 250Khz or

4us per bit cell.

Table 1-2: Bit Cell Times

Disk size Density Bit Cell Time

(Inch) (micro-seconds)

--------- ------- ---------------

8 Double 2

8 Single 4

5 1/4 Double 4

5 1/4 Single 8

HARD n/a (double) 0.2

The conventions used in encoding address marks in the ID field differ from the encoding methods of data. Address marks are encoded by deleting various clock bits in the byte which would normally be present. This is called missing clocks.

For example in the FM mode the ID address mark FE drops certain clock bits to produce the clock pattern C7 in place of the usual FF clock pattern.

1.3 Brief Hardware Overview

The OMNIDISK is a fast, intelligent controller designed to simultaneously interface the S-100 bus to 5.25 inch and 8 inch floppy disk drives.

In addition, circuitry is provided to interface the Western Digital 1000 family of Winchester hard disk controllers.

The OMNIDISK is designed to meet the requirements of the IEEE-696.2 standard.

High speed is achieved by reading a full track in one floppy disk revolution and buffering the data within RAM on the OMNIDISK.

Data is then given to the system in logical sectors as requested.

No additional disk reads are required until a new track is requested.

All deblocking and sector buffering is performed on the OMNIDISK for CP/M systems.

By removing sector buffering and deblocking from the CP/M BIOS, memory available to the user is increased.

This enables the CP/M-80 system to fit on two 8 inch single density tracks;

Thus the CP/M system can be booted from either single or double density 8 inch diskettes. Likewise for other operating system environments.

Data to or from the OMNIDISK may be either transferred through port mapped registers or via DMA. This option is software selectable.

The OMNIDISK will DMA to/from any address location within the 16 megabyte (24-bit) memory space defined by the IEEE-696 standard.

There are no address boundary limitations; i.e., the DMA data block addresses may cross 64K boundaries, unlike many other 8 and 16 bit systems.

The OMNIDISK may be operated at system clock speeds up to 12 MHz.

DMA transfers are made at full system clock speed without the need for host processor intervention.

Four (4) 5 1/4 drives, four (4) 8 inch drives, and four (4) different Winchester drives may be utilized at any one time.

The logical drive identifiers are user selectable through software; i.e., any physical drive number may be chosen to be any logical drive number.

The assignments may be changed at any time.

Some of the features offered by the OMNIDISK are:

o Meets or exceeds IEEE 696.2 specification. (S-100 buss)

o Supports both DMA (TMA) mode or Port oriented data

o 12 Mhz operation

o Operates with 8-bit or 16-bit processors, using 8-bit transfers

o Single or double or quad density Floppy drives

o Soft sectoring

o Any combination of 5.25 and 8 inch for a total of four each.

o Status register or vector interrupt driven

o Onboard 4 mhz 8085 processor

o Typical Board operating power consumption of ?? watts.

o Full track reads with onboard buffering

o 8k Firmware EPROM, 10k Buffer (16k optional)

o 12 Mhz DMA operation without wait states

o Wait states allowed during DMA

o Onboard CP/M sector buffering

o CP/M sector deblocking performed onboard

o Automatic Media density determination

o 8 or 16 bit port mapped

o Media-changed status available

o Supports MS-DOS FAT

o Western Digital 100X Winchester interface

o Supports fast RAM disk via DMA

1.4 Quick Integration Procedure

1.5 SKEW

Skew or sector interleaving occurs when logical sectors are in non-sequential order.

In some implementations this is determined during format time, in others it is done by software.

The advantage of skewing comes from a delay between logically sequential sectors.

This delay can be used for data processing and then deciding if the next sector should be read.

Without interleaving, the next sector could pass by the drive head, imposing a one revolution delay before any operation to it can be performed.

This delay varies as a function of revolution period, information density, and interleave factor. It is defined below.

An additional benefit from this delay is that bus utilization is reduced by spreading the data transfer over a greater amount of time, thereby reducing overall bus bandwidth requirements.

Skew usage applies to any media that involves such a cyclical delay.

For example: hard disks, floppies.

Thus, delay can be defined as:

1 revolution period

Delay : ------------------------ X ( interleave factor - 1 )

Sectors/Track

The following is a sample implementation of skew on a 8-inch single-sided,

single density, so-called standard 8" format.

Skew can apply to hard-sectored disks, other size floppies, and hard disks,

but the intent of the examples is the same.

The standard 8-inch format uses 26 128-byte sectors.

The formatting program writes a series of sector ID's.

Starting from the INDEX HOLE the ID's are written 1, 2, 3,,, 25, 26.

If one were to read sector 1 and then sector 2 and then sector 3, something bad would be seen: The process would take 26 times as long as might be expected.

To read each sector in order, one might expect 167 ms (one revolution at 360 RPM) but instead the process takes 167 ms for EACH sector, more than 4 seconds.

Clearly this is undesired. Why it takes this long will lead us to understand how this time can be reduced.

After reading data from sector 1, the ID for sector 2 is quite close to

passing under the read head. The read command for sector 2 must be issued

within microseconds or we will have to wait 167 ms for it to come around again.

Without some special "read multiple" command the normal sequencing commands between records would take too long for the floppy disk controller chip.

This problem has been solved in three ways.

Each of these solutions has found a place in current microcomputer systems.

Solution number one is just about the only way this problem is handled on

floppy disks. Assume that we were willing to read the odd numbered records first, followed by the even numbered records.

This would take 333 ms because the disk would have to rotate twice.

There would be enough time (in some systems) to get everything done in time to read the next odd or even ID. Early microcomputers were not very fast.

Therefore instead of reading every other sector, Digital Research, the makers of CP/M, decided to read every SIXTH sector.

This is the only standard floppy disk skew factor in all of the microcomputer industry.

Our time to read the track has grown to one second, but even that time is a lot better than the 4 seconds discussed earlier.

The second way the problem can be solved does the same thing in a different way.

If the ID's were FORMATTED in the proper out-of-sequence order, then reading records in order still would take only as long as discussed in the preceding paragraph.

This method has charm, but is not widely used in the microcomputer industry for floppy disks. This method IS common on HARD disks.

The third way this problem is solved involves "smart" hardware and considerable software effort.

An entire track can be read in 167 ms by advanced FD controller chips that have commands that access multiple records. "Smart chips" such as the NEC 765 and the INTEL 8272 can transfer as much as 16K bytes with one command. (16K comes from double-sided, double-density diskettes formatted with 8 1024-byte sectors.)

Clearly, in order to take advantage of non-skewed data transfer, one must have a large amount of memory to assign to a track buffer.

In addition, to make sense of a skewed disk, one must be adept in buffer management.

In this next section, we look at the CP/M single-density skew factor 6, and a general method of generating skew tables.

Here is how the makers of CP/M use the disk track:

Table 1-1: Skew on Standard 8-inch

1 2 Physical

12345678901234567890123456 Sector

-------------------------- ----------------

A.....B.....C.....D.....E. The letters A to Z

....F.....G.....H.....I... represent logical

..J.....K.....L.....M..... sectors 1 to 26.

.N.....O.....P.....Q.....R

.....S.....T.....U.....V.. In CP/M sector B

...W.....X.....Y.....Z.... follows A etc.

The algorithm for generating the logical to physical mapping is usually implemented by indexing into a table in memory.

The table in memory usually follows the following rule:

"Start at 1.

Add the skew factor.

If the sum is greater than the number of sectors on the track, then subtract the

number of sectors.

If the resulting number has been used before, add one and check again.

Continue until all sectors have been mapped."

This is easier to understand in C

j = 1 + nsec - iskewf; /\* start at a strange looking value \*/

/\* if to which is added iskewf and \*/

/\* MODed gives us j = 1 \*/

/\* this is a fix for first time \*/

for ( i = 1; i <= nsec; i++ ) /\* repeat for all logical secs \*/

{

j += iskewf; /\* add skew to current sector \*/

j %= nsec; /\* keep within range \*/

while(sector[j]) /\* find first free slot \*/

j++; /\* IF we didn't land on one \*/

sector[j] = i; /\* and assign value \*/

}

See also "Firmware Features & Facilities, Command Types and usage : SETSKEW"

for an sample usage, and command definition.

1.6 Onboard Track Buffering via an intelligent controller.

The use of a track buffer completely eliminates the need for skew with the exception of backward compatibility and this is implemented by logically skewing the track buffer.

The OMNIDISK uses an 8085 microprocessor to perform most of the disk tasks normally found in MS-DOS device drivers or a CP/M bios. When a read is requested from a track, the OMNIDISK reads the entire track and buffers the data in 10K of onboard RAM. The track read requires only one (1) revolution of the disk; therefore using about the same amount of time as a normal physical sector read. Once the track's data is in the OMNIDISK's memory, additional requests for data from that track do not require a disk access -- data is immediately available to the system. A significant increase in speed is realized by using this technique.

Sector requests to the OMNIDISK are by logical or physical sectors. The onboard firmware automatically deblocks the physical sectors. This greatly simplifies the software overhead on the operating system interface. Removing the deblocking requirement from the software reduces its memory requirements. The shorter BIOS will allow CP/M to fit onto two single density tracks rather than the usual combination of one single and one double density track. Now the user may SYSGEN either single or double density diskettes.

In a CP/M-80 environment additional user memory is made available due to the shorter BIOS and the onboard data buffering. CP/M may now be started 2K bytes higher in memory. User memory is therefore increased by 2K. This can be significant when running large programs on 8-bit processors. Whereas, for CP/M-plus the boot area is reduced to one single track.

1.7 Disk drives, Winchester and floppy.

The OMNIDISK will control both 5 1/4 inch and 8 inch floppy drives.

Any mixture of drive sizes may be used at any one time as long as no more than 4 drives of any one size are used. Thus eight (8) floppy drives may be used at any one time. Drive specific interface specifications are given in section 8.

The interface to a Western Digital 100X Winchester controller is provided. Either a CRC or ECC type is supported via a switch set in software. Up to four (4) hard disks may be used in addition to four (4) each 5 1/4 inch and 8 inch floppy drives.

With the newly released firmware (version 6.0 or newer), the Winchester drives may be all of different types. There are specific integration examples

in section 3.

1.8 Data Transfer - Processor vs. DMA

The OmniDisk performs I/O to the host system through port mapped data registers or by DMA in 128 byte bursts directly to/from system memory. A status register is used to tell the system CPU that a data byte has been accepted or is ready to be transmitted. There are no timing requirements when operating in this mode as the data is buffered and may be transferred at any rate. Interrupts may be generated on completion of DMA.

Data transfers may be made using DMA to improve data throughput. OmniDisk DMA is implemented per the IEEE-696 standard. DMA arbitration is provided as well as all status and control signals. One data byte is transferred for every four (4) clock cycles regardless of the clock frequency (up to 12 MHz). A status bit is loaded at the end of the DMA cycle. The status register may be checked by the system CPU or a vector interrupt may be generated upon the loading of the status register.

DMA transfers may be made from/to any RAM location within the 16M byte (24-bit) memory space defined in the IEEE-696 standard. Memory boundaries, such as multiples of 64K bytes, can be crossed during DMA transfers.

Although the OMNIDISK does not require any wait time during DMA transfers, it does respond to both ready lines per the IEEE-696 standard.

Table 1-3: Buss Bandwidth Requirements - CPU vs. DMA

Device Address Width Bandwidth

Bits Mbits/sec/Mhz

------------- ------------- -------------

LDIR, Z80 16 0.313

TMA, OmniDisk 24 0.263

LD, Z80 16 0.127

NOTE:

Since both DMA and processor data transfers are dependent on host CPU speed, the "bus-bandwidth" or data transfer rate is quoted "per Mhz".

2.0 PREPARATION FOR USE / CONVENTIONS.

This portion of the manual provides instructions for preparing and installing your OMNIDISK. Unpacking and inspection instructions are included, as are instructions for setting up the jumper options and switch options.

NOTE:

All references to left, right, up, down, etc. are assuming that one is holding the board with the component side up, and with the S-100 edge connector nearest.

2.1 Unpacking and Inspection.

All computer cards must be handled with care, since the components on them may be damaged by bending or bumping. Also the chips may come loose if the board is mishandled.

You should be especially careful of static electricity when you handle the OMNIDISK board, since the chips are susceptible to damage from static discharge. Any static electricity that has built up on your body should be discharged by touching an electrically grounded piece of metal (such as other grounded equipment, or a metal desk) before handling the board. For added safety keep the board in it's conductive envelope during transportation or handling.

Upon receipt of your OMNIDISK, immediately inspect the shipping carton and the board itself for evidence of mishandling or damage during transit. If the shipping container is severely damaged or water stained, contact the carrier and request that his agent be present when any additional cartons are opened. If the carton is opened and the carrier's agent is not present, save the carton and all shipping materials for the agent's inspection.

The shipping carton and packing material have been carefully designed to protect the OMNIDISK during shipment. If it becomes necessary to return a board, it should be repacked in its original shipping carton with its original packing material.

Check that all chips are seated in their sockets. If a chip is not fully seated in its socket, be sure that all of the pins of the chip are above the holes in its socket. Push gently on the end that is sticking up from the board until the chip is evenly flat against the socket.

Also, be sure that the black rectangular shunts on pin type connectors that stand above the board are pushed all the way down. If any of the jumpers have fallen off the board, read the next section and replace them on the correct pins.

2.2. Seating the Board.

Once you are certain that the jumpers are set correctly, install the board in your system. Make sure that the computer is turned off, and the power supply capacitors drained. Do not install or handle this board with the system on, as this may cause damage to the board components, traces, and other boards in your system.

Slide the board in any free slot on the motherboard, making sure the component side is oriented correctly. Gently push the top of the board until the board is seated in the motherboard. If you need to remove the board, pull gently on the top of board and rock it to loosen it from the edge connector, then simply pull up.

2.3 SETTING HARDWARE OPTIONS.

2.3.1 Factory Jumpers settings.

As the board is shipped from Fulcrum, unless otherwise requested, it is configured as follows:

Enable Boot from 8 inch floppy, Use port 00A0h, Disable Power-On-Boot PROM, No-precomp.

The above may be changed and is described in the following section. All else is configurable by software, and is described in the section "Firmware Features and Facilities".

2.3.2 Initial Boot Drive Type.

There are four choices - boot from 8" floppy, boot from 5.25" floppy, boot from Winchester, or boot from RAM-disk. The switch settings for each are given in table 2-1, below.

NOTE:

Only switch positions 4, 5, and 6 are relevant here. They are numbered 1 to 8, with 1 towards the top of the card.

Table 2-1: Boot Drive type, S1 options

Initial Boot Device Type S1 - position

4 5 6

8" floppy disk drive ON ON ON

5" floppy disk drive OFF ON ON

Winchester hard drive ON OFF OFF

Memory "Disk" ON OFF ON \*\*

\*\* NOTE:

Booting from memory disk requires use of battery back-up capable RAM such as the OMNI256.

2.3.3.1 IEEE 696 Port Address Selection : S1, S2, and J8.

Switches S1, and S2 settings determine the address space that the board resides at. The board requires two ports, consecutively. Factory setting is for port A0h data (and A1h status). Extended port addressing should be Disabled for 8-bit systems, Enabled for 16-bit systems.

Disable/Enable Extended Port Addressing (J8)

TO DISABLE SIXTEEN TO ENABLE SIXTEEN

BIT PORT ADDRESSING BIT PORT ADDRESSING

\*=\* \* \* \*=\*

J8 J8

Switch S2, located towards the left bottom of the card, determines the lower 7 address bits used to qualify a board select during an input or output cycle. Switch position 1 is located towards the upper end of S2, and corresponds with A8. Likewise switch position 7, next-to-last from the bottom, corresponds with A1.

(NOTE: A0 is not relevant here, and there is no switch for defined for it.)

Switch sense is inverted, that is, an ON corresponds to an address value of 0.

See table 2-3, below for some sample S2 settings.

Table 2-3: Sample Port addressing.

Address selected for S2 setting

Status Data (hex) 1 2 3 4 5 6 7 8

\*\* A1 A0 OFF ON OFF ON ON ON ON ON

A9 A8 OFF ON OFF ON OFF ON ON ON

51 50 ON OFF ON OFF ON ON ON ON

\*\* The default setting is A1/A0

2.3.3.2 IEEE-696 Port Address Selection - High Byte : S3.

Switch S3 is for setting the upper port address on sixteen bit port addressing. The conventions for S3 usage are similar to S2, except that S3 is for A9-15. Jumper J8 must be enabled for it's setting to be useful.

2.3.4. Disable/Enable Extended TMA Addressing : J6.

If your system does not allow extended dma addressing, this feature should be disabled. Extended DMA addressing is necessary to utilize the ram disk feature of the BIOS.

TO DISABLE EXTENDED TO ENABLE EXTENDED

DMA ADDRESSING DMA ADDRESSING

J6 \* J6 \*

|

\* \* \*=\*

2.3.5. Interrupts : J7.

The OMNIDISK may be set up to signal completion of its assigned task by interrupting the CPU. The OMNIDISK interrupts may be jumpered to INT\*, NMI\*, or any of the Vectored Interrupts.

The OMNIDISK does not provide a RESTART instruction when J7 is jumpered

to INT\* or NMI\*.

Table 2-6. J7: Interrupt Jumpering.

DATA RDY INT\*

\*----\*----\*----\* \*----\*----\*----\*----\* \*

\* \* \* \* \* \* \* \* \* \*

VI0\* VI1\* VI2\* VI3\* VI4\* VI5\* VI6\* VI7\* NMI\* (NC)

2.3.6. Boot PROM enable : Upper 3 pins of J9.

This hardware option enables a 512x8 PROM on reset. The PROM is used to store a minimal boot program for bootstrapping a larger boot program or even loading the operating system. It is enabled for every read, while writes are kept transparent "through" to underlying system RAM. Since only the lower 9 - bits (A0-8), are decoded the PROM is cloned throughout all the address space. Therefore a power-on-jump to anywhere on a 512 byte boundary would result in a valid boot method. There are some CPU's that require special considerations and some examples may be found in section 3.6.???? Under "Booting - other CPU considerations". This OmniDisk asserts S-100 - PHANTOM\* (69) on selection of the boot PROM.

See the file BPROM.ASM, the boot PROM source, for a typical implementation. Usually the code in the PROM would replicate itself in RAM then issue a command to the OMNIDISK to disable the boot PROM and then jump to the replicated code.

Table 2-7: Jumper Area J9.

7) \* \*

| DISABLE BOOT PROM

6) \* \*

| ENABLE BOOT PROM

5) \* \*

4) \* \*

3) \* \*

| PRECOMP INSIDE 38 TRKS.

2) \* \*

| PRECOMP ALL TRKS.

1) \* \*

J9 J9

2.3.8. Floppy Disk Write Precompensation : Lower 3 pins of J9.

Precompensation on the inner tracks is achieved by jumpering J9 1 to 2.

This provides 250ns precomp in MFM mode for 5.25" floppy disks and 125ns precomp in MFM mode for 8" floppy disks. The normal mode is to jumper J9 2 to 3.

Refer to table 2-7 above.

Refer to section 8.1-2 for disk drive types, and section 7.8 for a discussion of write precompensation.

2.4. Port Definition and Usage

Status Port:

Bit 0 : 1: OmniDisk has data to transfer via data port.

0: No data available.

Bit 7 : 0: OmniDisk is ready to accept next data or command via data port.

1: OmniDisk is busy.

2.5. DMA usage.

DMA speeds disk access by 10-30%, therefore it is desirable to implement. Although most all static RAM cards will work w/DMA, many DRAM (Dynamic) cards are not designed for DMA access and fail. In either case, a determination should be made if your system hardware will work w/DMA.

A utility is provided for this purpose: TMOV, See section 9.4.

2.6. Cabling.

OmniDisk <------> 8" Floppy Disk

IDS, 50 pin EDGE, 50 pin

OmniDisk <------> 5.25" Floppy Disk

IDS, 34 pin EDGE, 34 pin

OmniDisk <------> WD100X Controller

IDS, 40 pin IDS, 40 pin

3 - FIRMWARE - FEATURES & FACILITIES

3.1 Versions and Revision History.

Version 1.3 is the current release. The following command reference guide assumes you are running 1.3 or better.

3.2 Concept of Logical Device

The SET DEVICE command specifies the logical device for use by the following READ or WRITE commands. Right after power up or firmware reset one of the following devices is defined as Logical Device 0, as specified by the switch setting of S1.

Table 3-1: Initial Logical to Physical Device Mapping.

PHYSICAL- OMNIDISK- LOGICAL- CP/M

DEVICE DEVICE DEVICE DEVICE DRIVE

NUMBER CODE NUMBER LETTER

8" floppy 0 10h 00h A:

8" floppy 1 -- 01h B:

8" floppy 2 -- 02h C:

8" floppy 3 -- 03h D:

5" floppy 0 14h 04h E:

5" floppy 1 -- 05h F:

5" floppy 2 -- 06h G:

5" floppy 3 -- 07h H:

Hard disk 0 18h 08h I:

Hard disk 1 -- 09h J:

Hard disk 2 -- 0Ah K:

Hard disk 3 -- 0Bh L:

RAMdisk - 1CH 0CH M: of course.

Physical devices are rarely referenced directly, but when they are, they are always referenced via the OMNIDISK device codes given in the above table. Logical device numbers and CP/M device codes are always related as given above.

Normal reference to a device is through a logical device number. Logical devices are numbered 00H to 0FH. The mapping from logical device to physical device may be changed.

3.3 - Commands - Types, syntax, and usage.

Commands and parameters are sent to OmniDisk data port, while observing ready status on the status port. Refer to appendix F for listings of software used to send and receive lists of data from the OmniDisk. Parameters are specified as BYTE, WORD or binary fields in a BYTE or WORD.

Table 3-2: Summary Of Available Commands

Mnemonic Code Op Description

Hex Dec

oc$boot 00h 0 Boot system

oc$swrt 01h 1 Write system

oc$mode 02h 2 \* DMA or I/O mode

oc$fmtf 03h 3 Format floppy track

oc$dldm 04h 4 \* Define logical device mapping

oc$gldm 05h 5 Get logical device mapping

oc$fwrt 06h 6 Force write of modified buffers

oc$swap 07h 7 Turn Off Boot Prom Mapping

oc$skew 08h 8 Set non-standard skew factor

oc$unit 09h 9 \* Select (logical) unit

oc$trak 0Ah 10 \* Select track

oc$recd 0Bh 11 Select (logical) record

oc$dadr 0Ch 12 \* Set DMA address

oc$read 0Dh 13 Read

oc$writ 0Eh 14 Set head (HD only)

oc$movm 10h 16 Move block of memory with DMA

oc$gdpb 11h 17 Get DPB (CP/M 2.0)

oc$dfmd 12h 18 \* Define memory disk

oc$ssek 13h 19 Non-implied seek

oc$tiow 14h 20 Debug WRITE to OmniDisk

oc$tior 15h 21 Debug READS from OmniDisk

oc$gens 16h 22 Get general status

oc$exts 17h 23 Get extended status (clears general status)

oc$stry 18h 24 \* Set counters for (floppy disk) retry logic

oc$dpbx 19h 25 Get extended DPB (CP/M 3.0)

oc$gbpb 1Ah 26 Get MS-DOS "bios parameter block"

oc$gmcs 1Bh 27 Get MS-DOS "media change status"

oc$sfdp 1Ch 28 \* Set floppy disk parameters

oc$prea 1Dh 29 Physical sector read

oc$pwrt 1Eh 30 Physical sector write

oc$hdpr 1Fh 31 Set hard disk parms

(#heads, precomp, step rate, controller)

oc$hdtb 20h 32 Set hard disk tables

(Device selct, physical offset, O/S table)

oc$fdie 21h 33 Ignore floppy disk error & mark buffer valid

oc$sihd 22h 34 \* Set individual HD parameters

oc$gihd 23h 35 \* Get individual HD parameters and tables

oc$sfpb 24h 36 Set floppy DPB and PARAMETERS

oc$gfpb 25h 37 Get floppy DPB and PARAMETERS

oc$drid 26h 38 Read I.D. info to extended status

oc$sind 27h 39 Set individual step rates

oc$stpi 28h 40 Set track density status (TPI)

oc$gtpi 29h 41 Inverse of above

oc$vers 2Ah 42 Return version number

oc$tags 2Bh 43 Return buffer tags to host

oc$init FFh 255 Initialize OmniDisk to defaults

\*\* NOTE: Commands marked with a \* simply store parameter in OmniDisk internal RAM. This ram survives a hardware reset.

MNEMONIC and CODE:

oc$BOOT - Read System Tracks

00h 00-dec

PARAMETERS:

p1, BYTE : NSEC

Where NSEC defines the number of 128-byte blocks to read. A count of 1 transfers 128 bytes, a count of zero means 256 blocks or 32K bytes.

DESCRIPTION:

Transfer data to host system from selected logical disk track zero, sector 1 until all blocks have been transferred. Seeks to tracks one and following are performed as required to satisfy the block count. In common with the READ command, a GET GENERAL STATUS command should be issued before any data is read. The signal that this command has completed is the port flag indicating that the GET GENERAL STATUS command has been accepted and another byte may be sent to the OMNIDISK. The next byte read from the data port is the completion status. If this byte is zero then the boot command ran ok. If this byte is non-zero then see the section on error codes for reason. This command runs in byte-transfer mode even if DMA mode is in effect.

NOTES:

This command (with a count of one) is used in the boot PROM to boot the bootstrap sector which uses this command to load the rest of the system. The source code for the bootstrap sector is in the BIOS which is often distributed with the OMNIDISK controller. SYSGEN uses the boot command to read the system from the system tracks. The boot command works even if density and physical sector size is different on each of the tracks it reads.

MNEMONIC and CODE:

oc$SYSWRT - Write System Tracks

01h 01-dec

PARAMETERS:

p1, BYTE : NSEC

Where NSEC defines the number of 128-byte blocks to write. A count of 1 transfers 128 bytes, a count of zero means 256 blocks or 32K bytes.

DESCRIPTION:

Transfer data from host system to selected logical disk track zero, sector 1 until all blocks have been transferred. Seeks to tracks one and following are performed as required to satisfy the block count. After issuing the SYSWRT command and transfering the data, a GET GENERAL STATUS command should be issued and interpreted as follows:

If zero all went well, otherwise the OmniDisk was not able to

succesfully complete the SYSWRT command.

MNEMONIC and CODE:

oc$MODE - set DMA mode

02h 02-dec

PARAMETERS:

p1, BYTE : DBXXPPPP

Where bits PPPP is the IEEE-696 S-100 Bus DMA priority assignment.

Bit ‘D’ is defined 0 for no DMA usage, 1 use DMA.

Bit ‘B’ is defined 1 for BOOT/SYSWRT in progress.

"TMA" is the official IEEE-696 name for DMA.

DESCRIPTION:

Stored for later use by the OMNIDISK. Changes READ and WRITE to indicated DMA usage mode. OMNIDISK uses internal buffers to store disk data and therefore will not drop any data even if it runs at low priority and is often delayed by DMA contention from other devices.

The default power-on setting of this parameter is for no DMA. When used, it expected that SET DMA MODE will be used just before or just after system boot and left alone forevermore.

n some S-100 systems DMA will never work. If you have such a system you can still use the other powerful features of the Fulcrum OMNIDISK without fear that it will attempt a DMA operation and hang. As long as you do not specify DMA with this command, avoid defining a "memory disk" and do not issue the DMA block memory move command, no DMA operations will be done.

NOTES:

The BIOS often distributed with the OMNIDISK controller uses this command as a part of its initialzation. The BIOS DMA assembly option controls the setting of

DMA flag in this command. Even if host <==> controller data transfers do not use DMA, the DMA priority is needed for the memory disk and for the DMA move command.

MNEMONIC and CODE:

oc$FMTF - Format Disk Track (Floppy, Hard, or RAM-Disk)

03h 04-dec

PARAMETERS:

p1, BYTE : Format #, as defined by internal FDREL tables

p2, BYTE : Format control byte

0010 0000 - read list of available densities

0001 0000 - check # of sides, non-destructive

0001 0001 - check # of sides, destructive

0000 XXVF - Format/Verify

V - set V=1 for verify

F - set F=1 for format

XX - 00 : normal, use 2 heads

- 01 : use head 0 only

- 10 : use head 1 only

- 11 : use head 0 only, normal

DESCRIPTION:

Format, verify, or read physical-format-types list. If format or verify then UNIT and TRACK must have been previously issued to the OmniDisk. The first parameter specifies which physical format to use from the format list. This parameter is used in format and verify, but not read list. Second parameter specifies operation, either format, verify or read physical-format-types list. Status of operation is left in GENSTAT and EXT-STAT.

Table 3-3: Supported Physical Formats.

Fmt Type Dens Sec/ Bytes/ Notes

Code 8”/5" S/D/Q Track Sector --------

00h 0 8" S 26 128 Standard 8"

01h 1 8" S 15 256

02h 2 8" S 8 512

03h 3 8" S 4 1024

04h 4 8" D 26 256

05h 5 8" D 15 512

06h 6 8" D 8 1024 W/W Components

07h 7 5" S 16 128

08h 8 5" S 17 128

09h 9 5" S 18 128 Xerox

0Ah 10 5" S 8 256 Osborne

0Bh 11 5" S 10 256

0Ch 12 5" S 4 512

0Dh 13 5" D 16 256 Heath/NEC

0Eh 14 5" D 17 256

0Fh 15 5" D 18 256

10h 16 5" D 8 512 IBM PC/XT - 8 sector

11h 17 5" D 9 512 IBM PC/XT - 9 sector

12h 18 5" D 10 512

13h 19 5" D 4 1024

14h 20 5" D 5 1024

Transfer Rates

Density 5” Drive 8” Drive

S (FM) 250Kb/s 500Kb/s

D (MFM) 500Kb/s 1000Kb/s

Logical formats are defined relative the host O/S environment, and are listed in that section.

MNEMONIC and CODE:

oc$DLDM - Define Logical Device Mapping

04h 04-dec

PARAMETERS:

p1, BYTE : Logical device number 00h-0Fh

p2, BYTE : Physical device number 10h-1Ch

See Table 3-1 for a list of physical devices.

RETURNS:

None.

DESCRIPTION:

The mapping from logical device to physical device may be changed. The command DLDM or "Define Logical Device-Mapping" undefines the last defined physical member of a logical device, and defines the new physical member. The following commands configure a system with two 5.25" floppies which are physically wired as addresses 0 and 2:

OP p1 p1

-- -- --

04 00 14 5" unit 0 as CP/M A:

04 01 16 5" unit 2 as CP/M B:

04 02 00 define away logical unit 02

04 03 00 define away logical unit 03

04 04 00 define away logical unit 04 \* see NOTE

04 05 00 define away logical unit 05

04 06 00 define away logical unit 06 \* see NOTE

04 07 00 define away logical unit 07

04 08 00 define away logical unit 08

04 09 00 define away logical unit 09

04 0A 00 define away logical unit 0A

04 0B 00 define away logical unit 0B

NOTES:

In the above sequence, 04 04 00 and 04 06 00 are unnecessary. The first two commands define away the previous logical assignments of OMNIDISK devices 04 and 06.

The relation from logical device to physical device may be read by the host system. The command READ LOGICAL DEVICE MAPPING returns the physical device number for given logical device. Using the above 5" floppy as an example, 05 01 would

return 16.

Or in words:

Q: What device is logical 01?

A: Device 16, the 5" floppy wired as address 2.

MNEMONIC and CODE:

oc$GLDM - Get Logical Device Mapping

05h 05-dec

PARAMETERS:

p1, BYTE : Logical device number 00h-0Fh

RETURNS:

r1, BYTE : Physical device number

DESCRIPTION:

Returns current assignment of physical device to the specific logical device requested in accordance with the physical device definitions in table 3-1.

NOTES:

This command is not used in the BIOS, but is used in FORMAT and the logical unit swapper.

MNEMONIC and CODE:

oc$FWRT - Force Write of Modified Buffers

06h 06-dec

PARAMETERS:

p1, BYTE : (flush$mode)

Set 0 - Write pending

1 - Avoid problem sector ??

2 - Throw away buffer.

RETURNS:

None.

DESCRIPTION:

Flush altered buffers present in the OMNIDISK RAM. Begins by zeroing GENERAL STATUS which, if non-zero, would cause this command to be ignored. If the code is zero, then the normal write of modified buffers is done. If the code is one, then the write to the last sector giving a write error is ignored (ie "let's not, but say we did".) Following this action the normal write as in code zero is done. If the code is two, then all outstanding writes are ignored. Used to insure that no dirty buffers remain unwritten after this is executed. Otherwise the OMNIDISK does this all by itself, except as delayed by the write-delay" set by oc$SFDP.

NOTES:

At the time of this writing only the CP/M plus BIOS utilizes this command. In general this command is provided for paranoid writers of special-purpose software. One would do well to follow this command with a GET GENERAL STATUS command. One could then know when and with what success the FORCE WRITE command completed.

MNEMONIC and CODE:

oc$SWAP - Turn Off Boot Prom Mapping

07h 07-dec

PARAMETERS:

None.

RETURNS:

None.

DESCRIPTION:

This is used to release the address space used up by the boot PROM and free it for normal system use. It can only be mapped in again by a cold system reset.

NOTES:

Used by the bootstrap PROM after it has copied itself to RAM. Once a system is running, this command would make a dandy NOP command. Feel free....

MNEMONIC and CODE:

oc$SKEW - Set Sector Skew

08h 08-dec

PARAMETERS:

p1, BYTE : Logical or Physical device identifier

p2, BYTE : Sector length and Density [ 0D00LLLL ]

D : 1 for Double Density, 0 for Single Density

LLLL bytes/sector

0000 128

0001 256

0010 512

0011 1024

0100 2048

0101 4096

0110 8192

0111 16384

p3, BYTE : Number of sectors (or zero to accept first size match)

p4, BYTE : Desired skew factor (or zero to restore default)

RETURN STATUS:

Response to the SKEW command is left in GENERAL-STATUS. It is one byte long and defined among the following:

0: good

1: bad device (not floppy disk)

2: unknown format

3: bad skew factor

Note that the default skew values for the various floppy disk formats used by the OMNIDISK are determined by the FDRELTAB table in the onboard firmware. These default values may be redefined "on the fly" by using the OMNI controller SKEW command by means of the SSKEW program or by adding code to the CBOOT routines in the BIOS.

For example...

The default skew for the 8" double density 1024x8 disk is 2.

Several other floppy disk controller board manufacturers (Compupro and Morrow) typically use a skew of 3 with this format.

To alter your system to permanently use a skew of 3 with the 8" 1024x8 format without the need to revise the firmware, simply add the following code to your BIOS - CBOOT routines following the "define logical-to-physical mappings used by current system" section and immediately prior to the "define hard disk" sections.

; redefine 8" dd 1024x8 disks to have a skew of 3

;

DB OC$SKEW ;omni command: set

; non-standard skew factor

DB 00H ;logical device 00

DB 43H ;sector length code (3)

; + double density (40H)

DB 08H ;number of sectors

DB 03H ;desired skew factor

;

DB OC$SKEW ;omni command: set

; non-standard skew factor

DB 01H ;logical device 01

DB 43H ;sector length code (3)

; + double density (40H)

DB 08H ;number of sectors

DB 03H ;desired skew factor

The above example assumes that you have two 8" floppy drives defined as drives A: and B: in your system. If your 8" drives are defined otherwise, the logical device numbers in the above code will need to be changed. The SETSKEW command could also be used to set the skew of drive A: to one value and drive B: to another value; thus, for example, allowing data to be transferred between a Fulcrum format disk in one drive and a Compupro format disk in the second drive.

Unless again changed with SSKEW.COM or a comparable user program, the altered skew values will remain unchanged until a system reset or until the system is powered down.

MNEMONIC and CODE:

oc$UNIT - Select Logical or Physical Unit

09h 09-dec

PARAMETERS:

p1, BYTE : Logical drive unit 00-0Fh or

Physical drive unit 10-1Ch

RETURNS:

None.

DESCRIPTION:

Forces p1 to become the current unit. Actual select of physical unit is deferred until a READ or WRITE command is issued.

TRACK

MNEMONIC and CODE:

oc$TRAK - Select Track (Logical Seek)

0Ah 10-dec

PARAMETERS:

p1, WORD : Track

Low-order byte followed by high-order byte.

RETURNS:

None.

DESCRIPTION:

Selects track on current unit on which the next

read/write operation will be from.

Deffers seek (if needed) until the actual read or

write is performed.

NOTES:

All seeks to track zero on floppy disks are done using an

internal recallibrate command.

RECD

MNEMONIC and CODE:

oc$RECD - Select Logical Record

0Bh 11-dec

PARAMETERS:

p1, BYTE : Logical record number 00h-0FFh (0-255)

RESULTS:

None.

DESCRIPTION:

Selects logical record on current track of current

unit. Actual read/write operation deffered until

that command is issued.

NOTES:

This is LOGICAL not PHYSICAL sector. The OMNIDISK

controller performs all required skew arithmetic and

record deblocking.

DADR

MNEMONIC and CODE:

oc$DADR - Set DMA (TMA) transfer address

0Ch 12-dec

PARAMETERS:

p1, WORD : DMA address for data transfer - 24 bit

Low byte of 16 bit address followed by

high byte of 16 bit address.

p2, BYTE : Upper 8-bits of 24-bit address space.

RESULTS:

None.

DESCRIPTION:

No data transfer when received.

Set data address for next read/write operation.

P2 must contain typically 0FFh or 000h for systems

not utilizing extended address.

NOTES:

See SET DMA MODE command. SET DMA ADDRESS is dumb,

but quite legal, if one is operating in NON-DMA mode.

Also see notes in BIOS about setting the extended address

byte.

READ

MNEMONIC and CODE:

oc$READ - Read logical sector

0Dh 13-dec

INPUT PARAMETERS:

None.

RETURN PARAMETERS:

Status is left in GENERAL-STATUS.

DESCRIPTION:

Read data from current disk, current track, current

record to internal buffer. If in DMA mode,

then transfer is automatic to address set by DADR

command. Otherwise, requires 128 successive reads

to transfer data.

NOTES:

Device switch, seek, track selection, writing of modified

sectors and error recovery are all performed as required

to simplify the task of using the OMNIDISK controller.

The READ command does NOT provide an ending status.

It is therefore, almost essential that this command be

followed by a GET GENERAL STATUS command. When this is

done, the status bit that says that the OMNIDISK controller

can accept another byte from the host also says that the

next byte to be sent to the host is GENERAL-STATUS.

READ refuses to operate if GENERAL-STATUS is non-zero.

Unrecovered errors in the writing of modified sectors

(perhaps even on some other physical disk) also cause

READ to refuse to operate.

Such errors need to be at least acknowledged by the host.

Use GET EXTENDED STATUS and then retry the READ command.

See CPM 2.2 BIOS or CP/M Plus BIOS for examples.

WRITE

MNEMONIC and CODE:

oc$WRIT - Write logical sector

0Eh 14-dec

PARAMETERS:

p1, BYTE : Write Mode

0 - deferred write (NORMAL)

1 - no deferred write (DIRECTORY)

2 - deferred write to the first

sector of an unallocated data

block, no pre-read (SEQUENTIAL)

RESULTS:

Status is left in GENSTAT.

DESCRIPTION:

Write to current disk, current track, current record.

If in DMA mode, then transfer is automatic from address

set by DADR command. Otherwise,

requires 128 successive writes to dataport.

P1 is the CP/M write mode.

If zero, then normal rewrite in place.

If one, then directory write.

If two, then write at end of file,

( i.e. sequential write or padding. )

NOTES:

Device switch, seek, track selection, writing of previosly

modified sectors and error recovery are all performed

as required to simplify the task of using the OMNIDISK

controller.

The WRITE command DOES provide GENERAL-STATUS.

Should an attempted buffer flush or pre-read fail,

GENERAL-STATUS will be non-zero. The point at which this

status is provided depends on the operational mode defined

with SET DMA MODE.

In DMA mode, status is provided AFTER the data is

(I hate to say "data are") accepted from the host.

In program I/O mode, status is provided BEFORE the

data is accepted from the host.

(BEFORE because interrupt driven NON-DMA systems need the

interrupt-producing "data ready" before the write data may

be sent.)

NON-DMA systems NEED to check this status for non-zero

because the OMNIDISK is done with the write command when

it sends non-zero status. An attempt to send the write

data after non-zero status would be interpreted as a string

of OMNIDISK commands!

SETHEAD

MNEMONIC and CODE:

SETHEAD - Select Head (Only Used on Hard Disk)

0Fh 15-dec

PARAMETERS:

p1, BYTE : Head Number, 0 base.

RESULTS:

None.

DESCRIPTION:

None when received. Hardly any after that either.

As of this writing (JAN 85) the only use of this is to define the floppy disk head to be used in the GET DPB command.

MOVEMEM

MNEMONIC and CODE:

oc$MOVM - DMA memory move

10h 16-dec

PARAMETERS:

p1, BYTE : Low byte address \

p2, BYTE : High Byte address ]-- destination

p3, BYTE : Extended Address /

p4, BYTE : Low byte address \

p5, BYTE : High Byte address ]-- source

p6, BYTE : Extended Address /

p7, BYTE : Block count - in 128 byte segments

DESCRIPTION:

Moves memory via OMNIDISK hardware DMA feature.

No boundary restrictions.

i.e. You can move from anywhere to anywhere.

However the segment to be moved must be a multiple size

of 128 bytes (128 smallest, 32k largest.)

Also note that a block count of 0 means a move block

size of 32k.

If you are operating in a non-extended-addressing

system you need to either disable extended addressing,

or parameterize the DMA move in such a way that will

produce a valid address to move from or to.

Usually a GET GENERAL STATUS command is issued just

after the DMA move command to provide a way of

synchronizing the CPU and DMA operations.

i.e. Wait for DMA completion before doing anything.

Although this is not necessary, it makes keeping track

of \*WHAT\* goes on \*WHEN\* a lot easier.

SAMPLE CODING:

This is a sample implementation of DMA move.

This version can move 1 to 32,768 bytes anywhere to anywhere

w/in a 64k segment. NOTE: Z80 opcodes are used.

; MOVE$SB - Move, w/in memory in context.

; <DE> = source, <HL> = destination, <BC> = length

;

move$sb:

ld a,b ! or a

jr nz,dma$put ; must be zero

ld a,c ! and 128

jr z,move$man ; must be less then 128 bytes,

dma$put: ; therefore move must be non-dma

ld a,c ! and 128-1

push af ; leftover amount saved on stack

ld a,c ! and 128

ld c,a ; fix BC for 128 byte boundary

ld (dma$dst),hl ; Setup omnidisk command string

; destination address

add hl,bc ; Fix HL, to show occurance of move

ex hl,de ;

ld (dma$src) ; Setup omnidisk command string

; source address

add hl,bc ; fix hl, likewise

ex hl,de ; shift BC left one

push hl ! ld hl,0

add hl,bc ! add hl,bc

ld a,h ; number of 128 byte blocks to move

ld (blk$count),a ; store it in the string

pop hl

call ?cmdlist

defb 9 ; Length of command list

defb oc$MOVM ; Omni move command

dma$dst:

defw 0 ! defb 0 ; Extended address = 0

dma$src:

defw 0 ! defb 0 ; Extended address = 0

blk$count:

defb 0 ; how many 128 byte segments?

defb oc$gens ; Get general status..

call ?inchar ; Wait for DMA operation to complete

pop af ; Fix stack, and check for residual

ld b,0 ; amount to move.

ld c,a ; BC = residual length

or a ; HL, DE valid from above

ret z ; transfer to

; this is the "manual" move routine

move$man: ; BC=length, HL=destination, DE=source

ex hl,de ; we are passed source in DE and dest in HL

ldir ; use Z80 block move instruction

ex hl,de ; need next addresses in same regs

ret

GETDPB

MNEMONIC and CODE:

oc$GETDPB - "Read" O/S Disk Parameter Block From Disk Controller

11h 17-dec

PARAMETERS:

None.

RESULTS:

15 byte CP/M-80 DPB on success.

See the section on HOST O/S environment for a list

of defined DPB's.

DESCRIPTION:

Used to determine the density of a floppy disk drive.

Drive is tested for physical format, when found the DPB

defined in the firmware tables is sent back.

For hard disks, the values defined by HTBL command are

returned.

In general track four is used to determine a

floppy DPB. It was selected because

1) The directory track contains data that we least want zapped.

2) System tracks may not be representative of the density of the

entire disk.

3) The system takes 3 tracks on some 5 inch floppy formats.

Example usage:

CALL CMDLIST

DB 5 ;LENGTH OF LIST

DB OC$TRAK ;OMNI COMMAND: SELECT TRACK (SEEK)

DW 4 ;GO TO TRACK 4 FOR GETDPB FUNCTION

DB OC$GDPB ;OMNI COMMAND: GET DPB

DB OC$GENS ;OMNI COMMAND: GET GENERAL STATUS

lxi hl,DPBTBL ; Pointer to data area

jmp ?getdat ; Input O/S block.

NOTES:

As with READ and BOOT, following this command with

GET GENERAL STATUS is a good practice.

If there is an error, this command terminates,

leaving the reason in the GENERAL-STATUS byte.

See BIOS for an example of this command in use.

DFMD

MNEMONIC and CODE:

oc$DFMD - Define Memory Disk

12h 18-dec

PARAMETERS:

p1, BYTE : Upper 8 bits of 24 - bit address

of start of RAMDISK block 0.

p2, BYTE : Upper 8 bits of 24-bit address

of end of RAMDISK block 0.

p3-p8 : Repeat p1,p2 for block 1, block 2,

and block 3.

p9, BYTE : Length of RAMDISK O/S block.

p10-N, : O/S Disk Parameter Block.

p(n+1), BYTE : Init mode.

1 Perform init of RAMDISK on boot.

0 Do not initialize RAMDISK on boot.

RESULTS:

None.

DESCRIPTION:

Describes ram configuration to be used for

OmniDisk RAMDISK. Physically the RAMDISK is accessed

by DMA (TMA), to the host it appears as if it were

another disk. This command saves the 8 bytes of memory

address for use in accessing the memory disk.

If the initialization flag is non-zero, then then

RAMDISK is formatted on cold boot.

The 8 bytes of memory address are configured as 4

groups of 2 bytes. Each group is a start/end pair.

SAMPLE CODING:

call ?cmdlist

db 1+8+1+15+1 ; Length of list

db oc$DFMD ;OMNI COMMAND: DEFINE MEMORY DISK

; Addresses of memory disk areas

;

db start\_block$0, end\_block$0

db start\_block$1, end\_block$1

db start\_block$2, end\_block$2

db start\_block$3, end\_block$3

; Memory disk DPB

;

db 17 ;LENGTH OF DPB

DW 40H ;SPT, 8K BYTES PER "TRACK"

DB 4,0FH ;BSH, BLM 2K ALLOCATION SIZE

DB 0 ;EXM

DW 64 ;DSM, MAXIMUM ALLOCATION UNIT NUMBER

DW 128-1 ;DRM, MAXIMUM EXTENT NUMBER (DIRECTORY)

DW 08000h ;AL0, AL1 BITS FOR ALLOC UNITS IN DIRECTORY

DW 0 ;CKS, NO CHECK VECTOR

DW 0 ;OFF, reserved tracks for system

DW 0 ;psh/phm

db 0 ;RAMDISK INIT FLAG, 0=No init on boot.

SEPSEEK

MNEMONIC and CODE:

oc$SSEK - Force physical seek motion

13h 19-dec

PARAMETERS:

p1, BYTE : Track number 000 to 255.

p2, BYTE : Logical Unit number.

Will not work on physical drives.

RESULTS:

None.

DESCRIPTION:

Force selected floppy disk unit to perform head movement to

selected track.

DEBUG WRITE and DEBUG READ

MNEMONIC and CODE:

oc$tiow - Debug WRITE to OmniDisk

14h 20-dec

Oc$tior - Debug READ from OmniDisk

15h 21-dec

PARAMETERS:

p1, WORD : Memory address, Low byte, High byte

p2, WORD : Block size, Low byte, High byte

NOTE: if p2 is zero, then p1 is execution address.

RESULTS:

Oc$tiow None

Oc$tior 128 bytes

DESCRIPTION:

Debug tools.

GENS

MNEMONIC and CODE:

oc$GENS - Get General Status

16h 22-dec

PARAMETERS:

None.

RESULTS:

Returns a one byte description of current OmniDisk status.

DESCRIPTION:

The interpretation of data returned by the

GENERAL-STATUS command and the EXTENDED-STATUS

command are in section 5 - ERROR SUMMARY.

Note that the only way to clear general status is

by issuing a GET-EXTENDED-STATUS command.

This command is also used to signal the termination

of other commands by forcing the OmniDisk to send a

one byte result code.

See command - MOVM for an example of the latter usage.

EXTS

MNEMONIC and CODE:

oc$EXTS - Get Extended Status

17h 23-dec

PARAMETERS:

None.

RESULTS:

Returns a multi byte description of current OmniDisk status.

DESCRIPTION:

As used with floppy disks this command returns nine

bytes, in order:

general-status, physical sector, 7 byte NEC result code.

This command has no meaning for hard disks except to

clear general status.

An discussion of NEC765A result codes is found in the

ERROR-Interpretation section 5.

SRTY

MNEMONIC and CODE:

oc$SRTY - Set Counters for Floppy Disk Retry Logic

18h 24-dec

PARAMETERS:

p1, BYTE : Number of time to retry floppy disk access

before reporting failure.

RESULTS:

None.

DESCRIPTION:

Defaults to 3 retries by the firmware.

Usually requires no change.

DPBX

MNEMONIC and CODE:

oc$DPBX - Get Extended DPB ( CPM Plus )

19h 25-dec

PARAMETERS:

None.

RESULTS:

Returns 17 byte O/S block for CPM Plus.

DESCRIPTION:

Performs a determination of the physical track format

under the currently selected head, and returns an O/S

table entry defined in the firmware corresponding to

the physical format.

See section 4.1 for a description of CP/M Plus DPB.

GBPB

MNEMONIC and CODE:

oc$GBPB - Get MS-DOS Bios Parameter Block

1Ah 26-dec

PARAMETERS:

None.

RESULTS:

Returns 13 byte MS-DOS Bios parameter block (BPB).

DESCRIPTION:

Determines the BPB from internal tables assigned to

physical formats and the media descriptor byte found

on the media in question. For a short definition of

the BPB, see the section on

HOST O/S ENVIRONMENT : MS-DOS, section 4.6.

GMCS

MNEMONIC and CODE:

oc$GMCS - Get Media Change Status

1Bh 27-dec

PARAMETERS:

None.

RESULTS:

After execution the OMNIDISK data register contains

one of the following:

0 : media has definitely changed

-1 : media status is unknown

1 : media has not been changed

DESCRIPTION:

This opcode was intended to determine media change

status for MS-DOS use, however, it can be used equally

well for other O/S environments.

For example, the CP/M Plus BIOS makes good use of it.

SFDP

MNEMONIC and CODE:

oc$SDFP - Set Floppy Disk Parameters

1Ch 28-dec

PARAMETERS:

p1, BYTE : Write hold delay in 250ms ticks, 1-255ms.

p2, BYTE : Max length of time between last disk access

and head unload in 250ms ticks.

p3, BYTE : 8" floppy step rate in ms, 1-16 ms.

p4, BYTE : 5" floppy track-to-track step time in ms

range: 2-32 ms, rounded up to even

internally.

p5, BYTE : 8" Head load time in ms, 0-255 ms.

p6, BYTE : 5" floppy head load time, 0-255 ms.

p7, BYTE : 8" Head settle time after seek in ms, 0-255 ms.

p8, BYTE : 5" Head settle time after seek in ms.

DESCRIPTION:

Sets the indicated Floppy Disk Drive parameters.

Usually done once per power up, but may be used as

needed. Once set, all further use reflects the

new parameters.

PFREAD/PFWRITE

MNEMONIC and CODE:

oc$PREA - Physical sector read

1Dh 29-dec

oc$PWRT - Physical sector write

1Eh 30-dec

PARAMETERS:

Read : None passed. Track, sector and unit must

have been previously set.

Write : Requires N – BYTE’s as required to fill

one physical sector length of the current

media's format.

DESCRIPTION:

Performs "physical" sector I/O.

On read returns the entire physical sector.

On write expects to receive the entire physical sector,

either by DMA or through data port. These commands are

not normally used.

Error/status is left in GENERAL-STATUS and/or

EXTENDED-STATUS

HDPARM

MNEMONIC and CODE:

oc$HDPR - Set Common Hard Disk parameters

1Fh 31-dec

PARAMETERS:

p1, BYTE : Number of Heads:

1 to 8

p2, BYTE : Precompensation:

Usually around 1/2 of available tracks.

Not a critical parameter.

p3, BYTE : Step time:

0 - 7.5 ms, in 0.5 ms increments.

0 - has a special definition of RAMPED

or COMPILED type seeking.

i.e. p3 of 6 would set step rate to 3 ms.

p4, BYTE : Hard disk controller type:

Set to 20h if using a ECC type controller

or set to 0A0h for a CRC type controller.

wd1000-xx --> crc --> 020h

wd1001-xx --> ecc --> 0a0h

wd1002-xx --> ecc --> 0a0h

DESCRIPTION:

First of all, please note that the OmniDisk controller

controls a generic hard disk. In particular,

the controller does not assume some given number of

heads or a particular operating system file structure.

The OC$HDPR command is followed by 4 bytes giving

1) the number of heads

2) the precomp cylinder

3) the step rate for the disk

4) the (WD) controller type (ECC or CRC)

[Before this command was implemented, these parameters were hardcoded into the OmniDisk firmware, meaning that each type of hard disk needed its own special firmware. This was a distribution headache until the generic hard disk controller was made.]

Examples:

The following bytes, when sent to the OmniDisk controller,

define all hard disks as 3 headed with compiled seek and cylinder 100 as the first pre-comp cylinder:

DB OC$HDPR ;(1FH) COMMAND BYTE

DB 3 ;NUMBER OF HEADS

DB 100 ;(PHYSICAL) PRE-COMP CYLINDER

DB 0 ;STEP RATE (COMPILED SEEK)

DB 20H ;NOT AN ECC CONTROLLER

When the firmware for the OmniDisk controller was first written, it was assumed that there would never be more than one type of hard disk on any given controller. The 5 Meg removable media Syquest drives are ideal for backing up a much larger hard disk. The Syquest drives have only two heads while most larger capacity drives have four or six heads.

The OC$HDPR command is not sufficient because it defines the parameters for ALL hard disks at once.

At long last it is now possible to have several differing hard disks a given OmniDisk controller. One has but to issue the

OC$SIHD command with its required parameters during system startup.

The parameters are:

logical unit [which must have been previously defined],

number of heads,

the precomp (physical) cylinder,

and the step rate.

In typical use, whichever hard type you have the most of,

(or most internal device codes for - see next caution paragraph),

will be setup with OC$HDPR. Any remaining device(s) will be defined with OC$SIHD.

OmniDisk hard disk logical/physical unit concept is confusing

at best. For historical reasons, we intended to provide multiple physical devices on one logical device, not the current multiple logical devices on one physical device. The OC$HDTB command assigns a physical device select (and some other good stuff) to what the rest of the OmniDisk controller believes to be a unique device code. Do not confuse the EXTERNAL physical device select with the INTERNAL device code.

HDTBLS

MNEMONIC and CODE:

oc$HDTB - SET Hard Disk TABLES (DEV, OFFSET, OS BLOCK)

20h 32-dec

PARAMETERS:

p1, BYTE : Logical unit

p2, BYTE : Physical select \* 8

Valid selects are 0,1,2,3

i.e. physical unit 2 would translate to 10h

p3, WORD : Track offset to logical cylinder zero

Low byte first.

p4, BYTE : Length of OS block

0 ---> 256 bytes, 1 ---> 1 bytes, etc.

p5..N, BYTE : O/S block

DESCRIPTION:

The OC$HDTB command is followed by EXTERNAL device

select, a two-byte logical cylinder offset from the

beginning of the device, and a variable length

operating system (OS) block.

In CP/M this is the (15-byte) DPB. The OS block is

limited to 25 bytes. Within the OC$HDTB command,

the OS block is preceded by a length byte.

Example:

The following sets up a slow step rate disk as logical unit P:

DB OC$DLDS ;COMMAND BYTE: DEFINE LOGICAL UNIT

DB 'P'-'A' ;LOGICAL UNIT P:

DB 18H+1 ;HARD DISK UNIT 1

DB OC$SIHD ;COMMAND BYTE: DEFINE INDIVIDUAL HARD DISK

DB 'P'-'A' ;LOGICAL UNIT

DB 8 ;NUMBER OF HEADS

DB 250 ;PRE-COMP CYLINDER

DB 10 ;TAKES STEP PULSES AT 5MS RATE

DB OC$TBLS ;COMMAND BYTE: DEFINE HARD DISK TABLES

DB 'P'-'A' ;LOGICAL UNIT

DB UU\*8 ;PHYSICAL SELECT IS UU (RANGE OF UU IS 0..3)

DW OFFSET ;TRACK OFFSET TO LOGICAL CYLINDER ZERO

DB LEN ;LENGTH OF FOLLOWING BLOCK

DB .... ;CP/M DPB OR MS-DOS BPB

Things to watch for:

Both OC$SIHD and OC$TBLS require the logical unit byte which must be defined as a hard disk by a previous OC$DLDS command.

The "UU" in OC$TBLS is the ONLY place the "real" select code is present.

Note that if "UU" were split into 2 logical units that both

OC$SIDH and OC$TBLS commands would be needed for BOTH

logical units.

Do not confuse "OFFSET" in OC$TBLS with the CP/M DPB field

of the same name.

It is best to set the CP/M offset to 2 to allow cylinders 0 and 1 to be operating system image; Use the OC$TBLS OFFSET to

partition a large hard disk into several logical units.

This little trickery can be used with MS-DOS to partition

a large hard disk.

ERRIGNORE

oc$FDIE - Ignore Floppy Disk Error & Mark Buffer Valid

21h 33-dec

PARAMETERS:

None.

DESCRIPTION:

This new OmniDisk opcode is designed as an aid in

partial recovery of otherwise unreadable floppy disk

sectors. What this code DOES is simple.

More complex is how one might take advantage of what

it does.

When this one-byte opcode is received, the OmniDisk

controller checks its last attempt to access floppy

disk. If that last attempt ended in an error,

then the internal sector buffer for the "bad" sector

is marked as valid.

For example, assume a single-bit error is found in

the data field of some sector on a floppy disk,

and all attempts to read that sector by normal methods

fail. One might be interested in reading the sector

anyway -- perhaps from a sector patch utility or a

simple directory read command.

All that needs to be done is to issue the OC$FDIE

command and then re-issue the read.

WARNING:

If the sector was unreadable because of an error in the sector ID then the (OmniDisk internal) buffer has whatever was leftover from its previous use, most likely the contents of the same numbered sector, but on some other track. Also keep in mind that for sectors longer than 128 bytes that all "logical sectors" of the physical sector are effected.

Related note:

The standard OmniDisk BIOS special cases the ESC key when waiting for a response to the disk error message. If ESC is hit, then the OC$FDIE command is issued before a retry.

SETIHDP

oc$SIHD - SET INDIVIDUAL HD PARMS

22h 34-dec

PARAMETERS:

p1, BYTE : Logical hard disk unit: 00h - 0fh

p2, BYTE : Number of heads

1-8

p3, BYTE : Precomp cylinder

Usually set at 1/2 of all available

p4, BYTE : Step pulse rate \* 2 ms

i.e. 10h = 8 ms

DESCRIPTION:

Sets individual hard disk parameters providing a

method of having multiple differing hard disks

per OmniDisk controller.

Used only if needed on any EXCEPT first hard disk.

GETIHDP

oc$GIHD - GET INDIVIDUAL Hard Disk PARMS and TABLES

23h 35-dec

PARAMETERS:

p1, BYTE : Logical hard disk unit: 00h - 0fh

DESCRIPTION:

The last new opcode, OC$GHID, gets what was set up by

OC$SIHD and OC$TBLS.

Example:

DB OC$GHID ;COMMAND BYTE: GET INDIVIDUAL HD STUFF

DB one byte hard disk logical unit

And get the three bytes of OC$SIDH that follow logical unit, followed by all the bytes that follow logical unit in OC$TBLS.

Do not forget that the length of the operating system block may be different in different operating systems.

Do you wonder what happens if OC$TBLS has never been sent?

The length byte is zero, and that causes 256 bytes of O/S block to be returned.

RESET

MNEMONIC and CODE:

oc$RESET - Re-initialize OMNIDISK

0FFh 255-dec

PARAMETERS:

None.

DESCRIPTION:

Software reset of OmniDisk.

Re-initializes all parameters to original

or default state.

OMNIDISK Technical Reference Manual Operating Systems

4 - HOST OPERATING SYSTEM ENVIRONMENT INTERFACE

4.0 - Preface

Implementation varies with the host operating system.

Those specifically are the differences covered here.

OMNIDISK Technical Reference Manual CPM-80, CP/M+

4.1 - CP/M-80

First, a definition of the Disk Parameter Block,

the O/S interface block to disk drives used by CP/M BDOS.

Table 4-1: CP/M-80 and CP/M Plus DPB

/\*-------------------------------------------------------------\*\

| WORD - SPT Sectors per track |

+-------------------------------------------------------------+

| BYTE - BSH Data allocation block shift factor |

+-------------------------------------------------------------+

| BYTE - BSM 2^BSH - 1 |

+-------------------------------------------------------------+

| BYTE - EXM Extent mask |

+-------------------------------------------------------------+

| WORD - DSM Maximum block number |

+-------------------------------------------------------------+

| WORD - DRM Number of directory entries - 1 |

+-------------------------------------------------------------+

| WORD - ALV Left fed bits indicate dir reserved blocks |

+-------------------------------------------------------------+

| WORD - CKS Size of the directory checksumming vector |

+-------------------------------------------------------------+

| WORD - OFF Number of reserved tracks at disk beginning |

+-------------------------------------------------------------+

And, for CP/M plus, add this....

/+-------------------------------------------------------------+\

| BYTE - PSH Block shift factor \ |

+----------------------------------- Used for deblocking ----+

| BYTE - BSM Block shift mask / |

\\*-------------------------------------------------------------\*/

OMNIDISK Technical Reference Manual CPM-80, CP/M+

Table 4-2: BSH, BLS, BLM, EXM Relationships

Relations between BSH, BLM and the data allocation

BLock Size (allocation unit size.)

--BLS-- --BSH-- --BLM-- ---------EXM----------

DSM<256 DSM>255

1024 3 7 0 ---

2048 4 15 1 0

4096 5 31 3 1

8192 6 63 7 3

16384 7 127 15 7

Offset defines the number of tracks skipped until the directory.

It can be used for partioning hard disks into several logical drives, however, the OmniDisk Firmware provides a cleaner way of doing this.

OMNIDISK Technical Reference Manual CPM

Table 4-3: Supported Logical Formats for CP/M+ and CP/M-80

5-inch formats

Physical SPT BSH/BLM EXM max max-dir alloc. sum Off- Default

Format blk ent vector vector set Skew

5A-1 16 3/7 0 4B 64 C000 16 2 1

5A-2 32 3/7 0 97 64 C000 16 2 1

5B-1 17 3/7 0 4F 64 C000 16 2 1

5B-2 34 3/7 0 A0 64 C000 16 2 1

5C-1 18 3/7 0 52 32 8000 8 3 5

5C-2 36 3/7 0 AA 64 C000 16 2 5

5D-1 16 3/7 0 4B 64 C000 16 2 1

5D-2 32 3/7 0 97 64 C000 16 2 1

5E-1 20 4/F 1 2D 64 8000 16 3 1

5E-2 40 3/7 0 BD 64 C000 16 2 1

5F-1 16 3/7 0 4B 64 C000 16 2 2

5F-2 32 3/7 0 97 64 C000 16 2 2

5G-1 32 3/7 0 97 64 C000 16 2 1

5G-2 64 4/F 1 9B 256 F000 48 1 1

5H-1 34 3/7 0 9C 64 C000 16 3 1

5H-2 68 4/F 1 9C 128 C000 32 3 1

5I-1 36 3/7 0 AA 64 C000 16 2 1

5I-2 72 4/F 1 AA 64 8000 16 2 1

5J-1 32 3/7 0 9B 64 C000 16 1 1

5J-2 64 4/F 1 9B 64 8000 16 1 1

5K-1 36 3/7 0 AA 64 C000 16 2 1

5K-2 72 4/F 1 AA 128 C000 16 2 1

5L-1 40 3/7 0 BD 64 C000 16 2 1

5L-2 80 4/F 1 BD 128 C000 32 2 1

5M-1 32 3/7 0 97 64 C000 16 2 1

5M-2 64 4/F 1 97 128 C000 32 2 1

5N-1 40 4/F 1 5E 128 C000 32 2 1

5N-2 80 4/F 1 BD 192 E000 48 2 1

Table 4-5: Supported Logical Formats for CP/M+ and CP/M-80

8-inch formats

Physical SPT BSH/BLM EXM max max-dir alloc. sum Off- Default

Format blk ent vector vector set Skew

8A-1 26 3/7 0 F2 64 C000 16 2 6

8A-2 52 4/F 1 F2 128 C000 32 2 6

8B-1 30 4/F 1 8B 128 C000 32 2 1

8B-2 60 4/F 0 181 128 C000 32 2 1

8C-1 32 4/F 1 95 128 C000 32 2 1

8C-2 64 4/F 0 12B 128 C000 32 2 1

8D-1 32 4/F 1 95 128 C000 32 2 1

8D-2 64 4/F 0 12B 128 C000 32 2 1

8D-1 52 4/F 1 F2 128 C000 32 2 3

8D-2 104 4/F 0 1E6 128 C000 32 2 3

8E-1 60 4/F 0 118 128 C000 32 2 1

8E-2 120 4/F 0 231 128 C000 32 2 1

8F-1 64 4/F 0 12B 128 C000 32 2 2

8F-2 128 4/F 0 257 128 C000 32 2 2

4.2 - CP/M Plus ( CP/M 3.1 )

All formats similar to CP/M-80, with the exception the the PSH, PHM are set to 0. The OmniDisk does it's own deblocking, thus not requiring BDOS to set up deblocking buffer areas.

CP/M Plus is available in banked or non-banked version from Fulcrum, configured for the OmniDisk.

Banked version requires a minimum of 3 RAM "banks" set up as follows:

Port 40h is used for Bank selection

Bank 0 Bank 1 Bank 2 Common

00-3F X X X

40-7F X X

80-BF X X

C0-FF X

Select Always

Code 01h 02h 04h Selected

File BCB's (Buffer Control Blocks), Drive Allocation/Checksum vectors, directory hashing, drive buffers and DPB's are all set up automagically by GENCPM. Further, media change status is used rather than directory check-summing for disk change status determination.

At current there is no RAMDISK BIOS implementation for CP/M Plus.

4.3 - MP/M-80

4.4 - CP/M-86

4.5 - MP/M-86

DOS Sector Allocation

4.6 - MS-DOS

Minimum system size to boot MS-DOS is 128k, however, a minimum of 256k is needed to run most utilities.

Table 4-5: Allocation of sectors on MS-DOS disk

Sector Sector

Number Contents

+--------+--------------------------------------------------+

| 0 | Header record |

+--------+--------------------------------------------------+

| 1 | Boot sector for O/S disk. FF's for non boot disk |

+--------+--------------------------------------------------+

| 2-XX | File Allocation Table (FAT) Number 1 |

+--------+--------------------------------------------------+

| XX-XX | File Allocation Table (FAT) Number 2 |

+--------+--------------------------------------------------+

| XX-XX | Disc Directory |

+--------+--------------------------------------------------+

| XX-End | Disc Data Area |

+--------+--------------------------------------------------+

DOS - BPB, definition

Table 4-6: MS-DOS Bios Parameter Block (BPB)

/\*--------------------------------------------\*\

| 3 BYTE near jump to boot code |

|--------------------------------------------|

| 8 BYTES OEM and version number |

|--------------------------------------------|-------

| WORD bytes per sector | BPB \

|--------------------------------------------| \

| BYTE sectors per allocation unit | \

|--------------------------------------------| \

| WORD reserved sectors | \

|--------------------------------------------| \

| BYTE number of FATS | \

|--------------------------------------------| \

| WORD number of root directory entries | \

|--------------------------------------------| \

| WORD number of sectors in logical image | /

|--------------------------------------------| /

| BYTE media descriptor | /

|--------------------------------------------| /

| WORD number of FAT sectors | /

|--------------------------------------------|-------

| WORD sectors per track |

|--------------------------------------------|

| WORD number of heads |

|--------------------------------------------|

| WORD number of hidden sectors |

\\*--------------------------------------------\*/

The three words at the end (sectors per track, number of heads, and number of hidden sectors) are optional.

They are intended to help BIOS understand the media.

Sectors per track may be redundant as it could be calculated

from the total size of the disk.

Number of heads is useful for supporting different multi-head drives which have the same storage capacity, but different number of recording surfaces.

Number of hidden sectors may be used to support drive-partioning schemes.

Media descriptor bytes map directly to FAT ID bytes

DOS - BPB, definition

(which are constrained to the 8 values F8-FFh),

media bytes can, in general be any value in the range 00-ffh.

Table 4-7: Supported Logical Formats for MS-DOS

5-inch formats

WORD BYTE WORD BYTE WORD WORD BYTE WORD WORD

Physical Bytes Sec/ Rsvd #Fats #root #sec in media secs Sec/

Format /Sec Alloc secs dir media byte /FAT Track

5A-1 128 1 1 2 70h 280h 18h 8 16

5A-2 128 2 1 2 70h 500h 19h 8 32

5B-1 128 1 1 2 70h 2A8h 1Ah 8 17

5B-2 128 2 1 2 70h 550h 1Bh 8 34

5C-1 128 1 1 2 70h 2D0h 1Ch 9 18

5C-2 128 2 1 2 70h 5A0h 1Dh 9 36

5D-1 256 1 1 2 70h 140h 1eh 2 8

5D-2 256 2 1 2 70h 280h 1fh 2 16

5E-1 128 1 1 2 70h 2d0h 1ch 9 18

5E-2 128 2 1 2 70h 5A0h 1dh 9 36

5F-1 256 1 1 2 70h A0h 22h 1 4

5F-2 256 2 1 2 70h 140h 23h 1 8

5G-1 128 1 1 2 70h 280h 24h 4 16

5G-2 128 2 1 2 70h 500h 25h 4 32

5H-1 128 1 1 2 70h 2A8h 26h 4 17

5H-2 128 2 1 2 70h 550h 27h 4 34

5I-1 128 1 1 2 70h 2D0h 28h 5 18

5I-2 128 2 1 2 70h 5A0h 29h 5 36

5J-1 512 1 1 2 40h 140h feh 1 8

5J-2 512 2 1 2 70h 280h ffh 1 16

5K-1 512 1 1 2 40h 168h fch 2 9

5K-2 512 2 1 2 70h 2d0h fdh 2 18

5L-1 512 1 1 2 40h 190h fah 2 10

5L-2 512 2 1 2 70h 320h fbh 2 20

5M-1 1024 1 1 2 70h A0h 30h 1 4

5M-2 1024 2 1 2 70h 140h 31h 1 8

5N-1 1024 1 1 2 70h C8h 32h 1 5

5N-2 1024 2 1 2 70h 190h 33h 1 10

Table 4-8: Supported Logical Formats for MS-DOS

8-inch formats

WORD BYTE WORD BYTE WORD WORD BYTE WORD

WORD

Physical Bytes Sec/ Rsvd #Fats #root #sec in media secs Sec/

Format /Sec Alloc secs dir media byte /FAT Track

8A-1 128 4 1 2 44h 7d2h 02h 6 26

8A-2 128 4 1 2 44h fa4h 03h 12 52

8B-1 128 2 1 2 70h 483h 04h 4 15

8B-2 128 2 1 2 70h 609h 05h 7 30

8C-1 512 2 1 2 70h 268h 06h 1 8

8C-2 512 2 1 2 70h 4D0h 07h 2 16

8D-1 1024 2 1 2 70h 134h 08h 1 4

8D-2 1024 2 1 2 70h 268h 09h 1 8

8D-1 256 2 1 2 70h 7D2h 0ah 6 26

8D-2 256 2 1 2 70h fa4h 0bh 12 52

8E-1 512 2 1 2 70h 483h 0ch 2 15

8E-2 512 2 1 2 70h 906h 0dh 4 30

8F-1 1024 1 1 2 70h 268h 0eh 1 8

8F-2 1024 1 1 2 c0h 4d0h 0fh 2 16

OMNIDISK Technical Reference Manual Supported - formats

4.7 - CCP/M (ConCurrent PC-DOS)

4.8 - CP/M-68

4.9 - XENIX

OMNIDISK Technical Reference Manual Error Messages

5 - ERROR MESSAGES - Interpretations

5.0. Abstract

This section covers most error messages returned by the OmniDisk from the GET-GENERAL-STATUS and GET-EXTENDED-STATUS commands.

5.1. Floppy Drive

The General status byte is configures as follows:

11xxxxxx - Error detected in physical format

10xxxxxx - Error detected in physical read

01xxxxxx - Error detected in physical write

00010000 - Logical disk not defined

00010001 - Error on FMTTRK parameter list

00010010 - Can't read ID

00010011 - Retry fails to recover wrong cylinder

00010100 - Disk is of unknown density

Extended-Status

Using extended status yields the following information, after sending general-status.

1: General status, defined as above.

2: Physical sector causing error.

3: 7-byte 765 result phase status, defined below.

NEC 765 Result Phase, Status interpretation

When data transfer commands terminate, their result result phase has started generating 7 bytes of status.

This data is sent to the host verbatim using the GET-EXTENDED-STATUS command.

Byte 1: ST 0 - Status Register 0

2: ST 1 - Status Register 1

3: ST 2 - Status Register 2

4: C - Cylinder Number 0-MAX

5: H - Head Address 0 or 1

6: R - Sector number

NOTE: PHYSICAL sector not logical record.

7: N - Number of data bytes written in a sector.

OMNIDISK Technical Reference Manual Floppy Error Messages

Main Status Register:

Bit Description

0 : Drive number 0 is seeking

1 : Drive number 1 is seeking

2 : Drive number 2 is seeking

3 : Drive number 3 is seeking

4 : Command still in progress

5 : N.A.

6 : N.A.

7 : N.A.

Status Register 0:

Bit(s) Description

0 : Unit select 0

1 : Unit select 1

2 : Head state

3 : Drive Not Ready

4 : Track 0 locate failure,

or assertion of fault signal

5 : Seek end = 1

7,6 : Flags:

00 - Normal command termination

01 - Command not successfully completed

10 - Invalid command

11 - State of drive ready changed during command

Status Register 1:

Bit(s) Description

0 : Missing address mark.

1 : Write attempted on hardware

write-protected drive.

2 : No data on drive, or cannot find sector.

3 : Not used, always 0.

4 : Over run error.

5 : CRC error in data or ID fields.

6 : Not used, always 0.

7 : Attempted access to sector beyond end of track

OMNIDISK Technical Reference Manual Floppy Error Messages

Status Register 2:

Bit(s) Description

0 : Missing address mark in data field

1 : Bad cylinder

2 : Cannot find sector on current cylinder

3 : Not used. (scan equal hit)

4 : Wrong cylinder.

5 : CRC error in data field.

6 : Deleted data address mark found.

7 : Not used, always 0.

OMNIDISK Technical Reference Manual Hard Disk Errors

5.2. Winchester Hard Disk

The Hard Disk error status is returned by GENSTAT and is interpreted as follows:

WD1000-05 Error (Status) register Summary

Bit : 7 6 5 4 3 2 1 0

Field name: BBD CRC - ID - AC TZ DM

BBD : Bad Block Detected.

Find of an ID field marking this sector as BAD.

Used for bad sector mapping.

CRC : CRC error in data field.

When in CRC mode this bit is set when a CRC error

occurs in the data field. When retries are enabled,

ten or more attempts are made to read the sector

correctly. If none of these attempts are successful

only then is BBD set marking the sector bad.

ID : Cannot locate desired Cylinder, head and sector.

AC : Aborted Command. I.E. Drive not ready.

TZ : Track 0 not found.

DM : Data Address Mark (DAM) Not Found after reading

current sector ID mark.

5.3 - RAMDISK

No errors codes defined.

OMNIDISK Technical Reference Manual Diagnostics/Maintenance

6 - DIAGNOSTICS AND MAINTENANCE

OMNIDISK Technical Reference Manual Specifications

7 - ENGINEERING SPECIFICATIONS

7.1 PHYSICAL CHARACTERISTICS

The OMNIDISK is a single height card that occupies one card position in an S-100 motherboard. All input power and communications with the CPU or RAM take place thru the IEEE-696 connector.

J1, J2 and J3 carry signals to and from the various disk drives.

7.2 ENVIRONMENTAL CHARACTERISTICS

Non - Operating:

Temperature: -40 C to + 60 C

Relative Humidity 10% to 90%, non-condensing

Elevation Sea Level to 12,000 ft

Operating:

Temperature: 10 C to 40 C

Relative Humidity 10% to 90%, non-condensing Air

Filtered, as needed to maintain

temperature below 40C

7.3 DC Power Requirements

Operating characteristics rated at +8 volts +3/-1

1500 ma maximum

12 w maximum

Power supplies must meet all other IEEE696.2 (S-100) criteria.

7.4 - IEEE 696/ S100 signal specifications

Below are defined all S-100 signals, however no particular timing information is given.

For complete timing specifications refer to the IEEE-696 standard document:

Standard specifications for S100 bus interface devices IEEE task 696.1/D2.

All signals are TTL level except where noted and follow the usual convention of a low voltage being 0 or FALSE and a high voltage being a 1 or TRUE.

Signals that are active low follow the opposite convention and are denoted by the \* suffix.

Pin Signal Description

NO. Mnemonic

1 +8 Volts Instantaneous minimum greater than 7 volts

Instantaneous maximum less than 25 volts

Average maximum less than 11 volts

2 +16 Volts Power line for +16 volts

3 XRDY External Ready 1. Pulling this line low will

cause the CPU to enter a WAIT state.

4 VI0\* Vectored interrupt line 0.

5 VI1\* Vectored interrupt line 1.

6 VI2\* Vectored interrupt line 2.

7 VI3\* Vectored interrupt line 3.

8 VI4\* Vectored interrupt line 4.

9 VI5\* Vectored interrupt line 5.

10 VI6\* Vectored interrupt line 6.

11 VI7\* Vectored interrupt line 7.

12 NMI\* Non-maskable interrupt.

13 PWRFAIL Input to CPU indicating power failure.

15 A18 Extended address bit 18

16 A16 Extended address bit 16

17 A17 Extended address bit 17

18 STSDSBL\* Status Disable. Allows the buffers for

the 8 status lines to be disabled.

19 CCDSBL\* Command Control. Allows the buffers for

the 6 command control line to be disabled.

20 GND

22 ADSB\* The control signal used to disable the

24 - address line drivers on the CPU.

23 DODSB\* The control signal used to the CPU

data output drivers

24 o The master timing signal for the bus.

25 pSTVAL Status valid strobe from the CPU or

current bus master.

26 pHLDA A control signal indicating that the CPU

is yielding the bus to a DMA device

29 A5 Address bit 5

30 A4 Address bit 4

31 A3 Address bit 3

32 A15 Address bit 15

33 A12 Address bit 12

34 A9 Address bit 9

35 DO1/DATA1 Data out bit 1, bidirectional data bit 1

36 DO0/DATA0 Data out bit 0, bidirectional data bit 0

37 A10 Address bit 10

38 DO4/DATA4 Data out bit 4, bidirectional data bit 4

39 DO5/DATA5 Data out bit 5, bidirectional data bit 5

40 DO6/DATA6 Data out bit 6, bidirectional data bit 6

41 DI2/DATA10 Data in bit 2, bidirectional data bit 10

42 DI3/DATA11 Data in bit 3, bidirectional data bit 11

43 DI7/DATA15 Data in bit 7, bidirectional data bit 15

44 sM1 The status signal which indicates that the

current cycle is an op-code fetch.

45 sOUT The status signal identifying the data

transfer bus cycle to an output device.

46 sINP The status signal identifying the data

transfer bus cycle from an input device.

47 sMEMR The status signal identifying bus cycles

which transfer data from memory to a bus

master, which are not interrupt acknowledge

instruction fetch cycle(s)

48 sHLDA The status signal which acknowledges that a

HALT instruction has been executed.

49 Clock 2 Mhz output from the CPU, synchronous to

the CPU main clock.

50 GND Common with pin 100.

51 +8 volts Common with pin 1.

52 -16 volts Power line for -16 volts.

53 GND Common ground return. ( same as 50 )

54 SLAVE CLR\* A reset signal to reset bus slaves.

Must be active with POC\* and may also be

generated by external means.

58 sXTRQ The status which requests 16-bit slaves to

assert SIXTN\*

59 A19 Extended Address bit 19

60 SIXTN\* The signal generated by 16-bit slaves in

response to the 16-bit request signal

sXTRQ\*.

61 A20 Extended Address bit 20

62 A21 Extended Address bit 21

63 A22 Extended Address bit 22

64 A23 Extended Address bit 23

67 PHANTOM\* A bus signal which disables slave devices

and enables phantom slaves - primarily

used for bootstrapping systems without

hardware front panels.

68 MWRT pWR\* - sOUT

70 GND Common ground return.

72 pRDY Indicates slave ( or the CPU itself) is

requesting the master to enter into

a wait state in the current cycle.

73 INT\* The primary interrupt request bus signal.

74 pHOLD Input to the CPU requesting it to

relinquish control of the bus to

another master.

75 RESET Master reset for the CPU.

76 pSYNC Control signal from the CPU indicating it

it is beginning a new bus cycle.

77 pWR The control signal signifying the presence

of valid data on DO or DATA bus

78 pDBIN The control signal that requests data on

the DI bus or DATA bus from the currently

addressed slave.

79 A0 Address bit 0

80 A1 Address bit 1

81 A2 Address bit 2

82 A6 Address bit 6

83 A7 Address bit 7

84 A8 Address bit 8

85 A13 Address bit 13

86 A14 Address bit 14

87 A15 Address bit 15

88 DO2/DATA2 Data out bit 2, bidirectional data bit 2

89 DO3/DATA3 Data out bit 3, bidirectional data bit 3

90 DO7/DATA7 Data out bit 7, bidirectional data bit 7

91 DI4/DATA4 Data in bit 4, bidirectional data bit 4

92 DI5/DATA5 Data in bit 5, bidirectional data bit 5

93 DI6/DATA6 Data in bit 6, bidirectional data bit 6

94 DI1/DATA1 Data in bit 1, bidirectional data bit 1

95 DI0/DATA0 Data in bit 0, bidirectional data bit 0

96 sINTA The status signal identifying the bus

input cycle(s) that may follow an

accepted interrupt request presented

on INT\*.

97 sWO\* The status signal identifying a bus cycle

which transfers data from a bus master

to a slave.

98 ERROR Indicates the present bus cycle has

generated an error condition.

99 POC\* The power-on clear signal for

all bus devices. 10ms period required.

100 GND System electrical ground.

7.5 WINCHESTER WD100X Interface

Summary of WD1000-05 Winchester Disk Controller Usage

General

The WD1000-05 series is based on a proprietary (to WD) chip set designed specifically for Winchester drive control consisting of WD1010AL-05, and WD1100AL-11. The WD1000-05 is a general purpose Winchester controller card.

The drive signals are based on the Seagate ST506 interface.

Host interface is via a 8-bit, bi-directional bus with appropriate control signals.

All data, status information, and macro commands are transferred via this bus.

An on-board Sector Buffer allows data transfers to the OmniDisk host independent of the actual drive data transfer rate.

Features as used by the OmniDisk:

o Write precompensation

o 5 Mbits/second data rate

o up to 4 Winchester Drives

o up to 8 read/write heads

o 1024 Cylinder addressing range

o 256 sector addressing range

o CRC generation + validation

o Single +5 Power Supply

o Programmable Stepping Rate

o MFM encoded recording

o Soft sectoring

o Step rate: 7.5 ms to 35 us, in 0.5 ms increments

WD1000-05 Host Interface Connector : J5, 40pin

Maximum recommended length : 4 ft.

Gnd Pin Name I/O Description

2 1 DAL0 I/O 8-bit Tri-state Data Bus

4 3 DAL1

6 5 DAL2

8 7 DAL3

10 9 DAL4

12 11 DAL5

14 13 DAL6

16 15 DAL7

18 17 A0 I Address of task register or

20 19 A1 I buffer to be accessed

22 21 A2 I

24 23 CS\* I Select

26 25 WE\* I Write Enable

28 27 RE\* I Read Enable

30 29 Pulled up

32 31 Not Connected

34 33 Not Connected

36 35 INTRQ O Asserted on command completion

38 37 BDRQ O Buffer Data Request, read or write

40 39 MR\* I Master Reset

42 41 Not Connected

44 43 “

46 45 “

48 47 “

50 49 “

WD1000-05 Drive Control Connector Pin Description : J6, 34pin

Maximum recommended length : 10 ft.

Gnd Sig Name I/O Description

1 2 RWC\* O Reduce write current

3 4 HS2\* O Head select 2

5 6 WG\* O Write Gate.

Asserted when valid data is to be written.

7 8 SC\* I Seek Complete.

9 10 TK000\* I Heads over outermost cylinder

11 12 WF\* I Write fault.

Asserted by drive when a write error occurs.

13 14 HS0\* O Head select 0

15 16 Not Connected

17 18 HS1\* O Head Select 1

19 20 INDEX\* I Indicated a start of track.

One per revolution

21 22 DRDY\* I Ready - Drive Selected and up to speed

23 24 STEP\* O Step pulse.

25 26 DSEL1\* O Drive Select 1

27 28 DSEL2\* O Drive Select 2

29 30 DSEL3\* O Drive Select 3

31 32 DSEL4\* O Drive Select 4

33 34 DIRIN\* O Stepping direction, IN : asserted.

WD1000-05 Drive Data Connectors : J1, J2, J3, J4, 20pin

Maximum recommended cable length : 10 ft.

Gnd Signal I/O Name

2 1 \

4 3 \

6 5 \

8 7 / Not Connected

10 9 /

12 11 /

15 13 O + MFM Write Data, RS422 Differential Data

16 14 O - MFM Write Data, RS422 Differential Data

19 17 I + MFM Read Data, RS422 Differential Data

20 18 I - MFM Read Data, RS422 Differential Data

WD 1000-05 Power Connector : J7

Pin Signal Type

1 Not Connected

2 Ground \

3 Ground ] 3.0A max, 2.5A typical

4 +5v, Regulated to +/- 5% /

Physical Characteristics

Dimensions:

Length : 8.00 in. (20.3 cm)

Width : 5.75 in. (14.5 cm)

Height : 0.75 in. ( 1.9 cm)

Environmental:

Operating Temp : 0-50 C

Rel. Humidity : 20-80 %

Air Flow : 100 linear fpm. at 0.5 in.

OMNIDISK Technical Reference Manual Floppy Disk Interface

7.6 5 1/4 INCH FLOPPY DISK INTERFACE (J2)

Pin No. Function

2 LO CURRENT\* Reduce Write Current

4 HLD\* Head Load

6 DS3\* Drive Select 3

8 INDEX\* Index Pulse

10 DS0\* Drive Select 0

12 DS1\* Drive Select 1

14 DS2\* Drive Select 2

16 MOT ON\* Motor On

18 DIR\* Step Direction

20 STEP\* Step Pulse

22 WR DATA\* Write Data

24 WR ENABLE\* Write Enable

26 TR0\* Track 0

28 WR PROT\* Write Protect

30 RD DATA\* Read data

32 SS\* Side Select

34 READY\* FDD is in ready state

ALL ODD GROUND

OMNIDISK Technical Reference Manual Floppy Disk Interface

7.7 8 INCH FLOPPY DISK INTERFACES (J1)

Pin No. Function

2 LO CURRENT\*

4,6,8 NC

10 2 SIDED\*

12 NC

14 SS\*

16 NC

18 HLD\*

20 INDEX\*

22 READY\*

24 MTR ON\*

26 DS0\*

28 DS1\*

30 DS2\*

32 DS3\*

34 DIR\*

36 STEP\*

38 WR DATA\*

40 WR ENABLE\*

42 TR0\*

44 WR PROTECT\*

46 RD DATA\*

48 NC

50 NC

ALL ODD GROUND

7.8 - THEORY OF OPERATION

FUNCTIONAL DESIGN

Introduction

This portion of the manual contains a detailed description of the OMNIDISK hardware.

It is assumed that the reader is familiar with digital logic and with the timing and electrical characteristics of the IEEE-696 bus.

Overview

The OMNIDISK's architecture can be broken down into six functional areas:

o Floppy Disk Interface

o Winchester Controller Interface

o Onboard CPU

o DMA Logic

o IEEE-696 Interface - PORT

o IEEE-696 Interface - TMA

Floppy Disk Interface

At the center of the floppy disk control circuitry is the NEC uPD765 LSI controller chip. This controller chip keeps track of stepping rates, head load/unload timing, formatting, and in general all data transfers between the track buffer/CPU and the actual floppy disk drives. The uPD765 is supported by a number of chips which are described below. In addition it provides a clean interface to the onboard CPU bus, the digital PLL data separator, precompensation shift register, and a multiplexed floppy control bus. The uPD765 requires two main clock signals; a general purpose clock and a write data clock. These are signals are generated thru U8, U5, and U7.

The basic oscillator onboard is operated at 16.000 MHz and consists of 1/2 of U8 (7404) and a 16 mhz rock. This 16.000 MHz signal is fed into a 74LS393 and divided down, generating 8.000, 4.000, 2.000, 1.000, 0.500, 0.250 Mhz synchronous symmetric signals. The 4 Mhz is used to clock the 8085AH CPU and along with the 8 and 2 Mhz signals feeds the 74LS153 MUX inputs. The 74LS153 depending on the signals 8\* and MFM decides what clock the data separator needs. The other half of the 74LS153 performs similar function, \*except\* it uses the 0.250, 0.500, and 1.000 Mhz signals for input and drives the Write Data line of the 765 FDC. The table below summarizes this:

Drive 8\* MFM Write Data Separator

Type Clock Clock

5" 1 0 0.250 Mhz 2.000 Mhz

5" 1 1 0.500 Mhz 4.000 Mhz

8" 0 0 0.500 Mhz 4.000 Mhz

8" 0 1 1.000 Mhz 8.000 Mhz

The 8\* signal is generated by latching CPU data bit 5 in U39 (74LS374) during a CPU output cycle. In general 8\* reflects the current disk size in use. It is low for 8 inch drives and high for 5 1/4 inch drives. The MFM signal is set by the FDC depending on if the commands issued to it use single density mode (FM) of double density mode (MFM). Before the write-clock signal is fed to the MUX it is and`ed by U6 (74LS08) to produce an asymmetrical signal with an on-time of invariably 250ns independent of the actual write-clock frequency.

The 765 FDC (U13) pin 39 controls the demultiplexing of the following signals: 2 SIDED\* and WR PROT\*, TR0\*, STEP\*, LO CURRENT\* and DIR\*.

IC U12 (74LS240) is used to demultiplex these signals. It should be noted that portions of U12 are used in other parts of the circuit.

U4 (74LS145) is used to demultiplex the drive select (US0 and US1) signals.

Additionally, the drive select signals are further divided as a function of the signal 8\*, or drive size selected.

The 765 FDC polls all drives continuously in the selected set (i.e. 5's or 8's) and keeps track of the ready status for each. If status of any drive changes it issues an interrupt to the 8085AH CPU. It in turn sets some flags in RAM modifying \*behavior on the next access to that drive.

A Standard Microsystems Floppy Disk Data Separator (FDDS), FDC-9216B (U28), is used to separate the FM or MFM encoded data provided by the floppy disk drive into separate clock and data inputs used by the 765 FDC. The FDDS consists primarily of a clock divider, a long term timing corrector, a short term timing corrector, and reclocking circuitry. U28 requires a reference clock (REFCLK) frequency dependent upon the disk drive size and density.

(See table above for a breakdown).

Internally the division ratio is selected by inputs CD0, and CD1, but this is hardware set to 1. The FDDS detects the leading edges of the disk data pulses and adjusts the phase of the internal clock to provide the SEPARATED CLOCK output used by the 765 FDC. Separate short and long term correctors are employed to assure accurate clock separation. The internal clock frequency is nominally 16 times the SEPARATED CLOCK frequency. Depending on the internal timing correction, the internal clock may be a minimum of 12 times to a maximum of 22 times the SEPARATED CLOCK frequency.

Write Precompensation

The algorithm used for precompensation is implemented within the 765A/8272A. The actual AMOUNT of precompensation is controlled by external hardware. This is implemented by a decoder and 4-bit shift register.

The shift register U24 (74LS195) is fed with the signal WR\* (765, pin 30) or uncompensated write data. It generates the Early, Normal, and Late signals depending on the status of signals Pre-shift 0 and Pre-shift 1. Normal precompensation condition is derived thru the lack of either PS0 or PS1 signals (U27 and U26). Precompensation is provided only during double density operation by using the MFM signal from U13. The output of the shift register is sent to the drive as the precompensated write. The actual amount of precompensation is controlled by the clock signals to the shift register. An 8 mhz clock results in 125ns of precomp. A typical range for most standard floppy disk drives is 125ns to 250ns. Consult your disk drive manual for the proper value.

Winchester Controller Interface

An interface to the Western Digital 100x series of Winchester Controller boards is provided. The interface is primarily a bidirectional bus driver (U15 - 74LS245) for the eight (8) data lines and an address driver for the various control lines.

U15 is always enabled. The normal flow of data is from the OMNI-DISK to the WD 100x. The data flow is reversed during a read from the WD 100x. The direction is reversed via U49-A.

U16 (74LS244) is a bus driver for the various control signals (A0, A1, A2, WE\*, RE\*, CS\*, and MR\*) to the WD 100x. In addition, U16 buffers the WAIT\* line from the WD 100x.

Onboard CPU

The OMNIDISK utilizes an 8085AH CPU to simplify HOST software requirements in a disk environment. The actual commands implemented in an 8kx8 firmware EPROM (U32) are described in section 3 of this manual.

There are 10K to 16K-bytes of fast RAM provided. This is enough RAM to buffer a full 8K track as well as a physical sector from a Winchester.

DMA Logic

IEEE 696 Processor Signals

The processor control signals are generated thru a state machine implemented by a 4-bit counter (U68), a PROM (U67), and some misc logic. Phase-2 (S100 pin 24) clocks the state machine counter. Ready (S100 pin 72) is or`ed with U67.7 and used to hold the count or state. The A and B output of the counter are fed into a PROM (U67) and are decoded into the resulting outputs for:

pSYNC, pSTVAL, pWR\*, pDBIN, WR\*, RD\*, and ready enable.

The WR\* and RD\* outputs are used for the internal RAM while pWR\* and pDBIN are used for system RAM. The pSYNC output OR`ed with C\* also clocks the internal-DMA counter.

The ready enable signal is OR`ed with the Ready signal in order to disable the counter while Ready is low on the rising edge of bus cycle 2. The PROM outputs will maintain their status until Ready goes high -- no matter how many times the clock cycles, thereby implementing wait states for DMA access.

The processor control signals are enabled simultaneously with disabling the master CPU status and one clock cycle prior to disabling of the master CPU processor signals.

IEEE-696 Status Signals.

The IEEE-696 status signals are latched onto the bus via U66 (74LS374).

The onboard firmware determines the correct status signals and latches the byte. U66 is enabled simultaneously with the disabling of the master CPU status.

IEEE-696 Disabling Signals.

The disabling signals timing is generated via U54 (74LS175). Once the DMA arbitration gives the OK (MINE), U54 provides the timing to disable and to enable the IEEE-696 address, data, status, and processor signals.

The 'C' output disables the address, status, and data signals. In addition the 'C' output enables the processor signals and status signals as described above.

IEEE-696 Address Signals.

The IEEE-696 address signals are generated by cascading a series of 74LS161 counters (U40-U45). The counters are software loaded. The count signal is pSYNC. Three (3) 74LS244 buffers (U57, U59, and U61) are used to drive the bus with the 24-bit address from the counter chain. The extended address driver may be continuously tristated by removing jumper J6.

Internal Address Bus Signals.

The internal address is generated by a combination of

counters and latches. The lower seven (7) address bits are generated by a dual 4-bit counter (U53 - 74LS393). The eighth (8) address bit is latched via an 74LS74 (U69). The upper eight (8) address bits are latched onto the address bus via an 74LS374 (U30).

The DMA circuit is designed to transfer 128 bytes at a time. At the end of an 128-byte transfer, U53 clocks the release of both the internal and external address buses.

DMA Arbitration.

Although the DMA arbitration circuit is not identical to the IEEE-696 standard circuit, it provides the same response.

The DMA priority level is software set. The priority is latched into the lower 4-bits of U39 (74LS374). A PROM (U55 - 74S473) replaces many of the discrete TTL gates of the IEEE-696 circuit. U55 has open collector outputs, and is the bus driver. The remainder of the circuit is consistent with the IEEE-696 specification.

A DMA cycle is started by causing IWANT to go high. This is created by performing an OUT to the GODMA port. Once the priority logic determines that the bus is available, the MINE line goes high. The MINE signal starts a triggering sequence of D-type flip-flops (U54 - 74LS175). This sequence of triggered flip-flops is used to enable and disable the various address, data, status, and control bus drivers.

IEEE-696 Interface

The port address decoding is provided by Switch 2 and 3 in combination with two eight-bit comparators (U58 and U60 - 74LS682). The extended address feature may be disabled thru J8.

Data and commands are latched into the OMNIDISK thru U62 (74LS374). The data is latched via a combination of board select and a pWR\*. A byte available status signal is created at this time via a D-type flip-flop (U22a). The byte available status may be read by both the system CPU and the board CPU. The byte available status signal is reset when the board CPU reads the data.

Data and status is returned to the system by latching the data into a 74LS374 (U65).

The chip select for U65 sets a status line (U22B) which may be read by the system CPU or may drive an interrupt line. This status line is reset when the system CPU reads the data port. A jumper pad (J7) is provided to route the interrupt signal to INT\*, NMI\*, or the Vectored Interrupt lines.

OMNIDISK Technical Reference Manual DRIVE SPECIFICATIONS

8 - SELECTED DRIVE SPECIFICATIONS

8.0 Abstract

This section may be helpful in determining drive parameters that are used in setting up the OMNIDISK for operation. In general ONLY one floppy drive out of all combinations of 5" and 8", connected to the OMNIDISK should be terminated.

For example, if an 8" drive is terminated, do not terminate any of the 5 1/4" drives.

8.1 5.25" Floppy Drive Configuration

TANDON 100-2 (STEP5=6, LOAD5=0, SETL5=15)

The appropriate straps on the dip shunt (1E) should be connected, the others left open. This is usually accomplished by removing the shunt from its socket, bending out one of the pins for each strap which is to remain open, and plugging the shunt back into the dip socket such that the bent-out pins are not inserted into the dip socket.

Drive select is determined by the straps connecting pins 2 to 15 (DS0), 3 to 14 (DS1), 4 to 13 (DS2), and 5 to 12 (DS3). Selecting drive 4 (DS3) may require other modifications to the drive pc (see the Tandon TM100 drive manual). If only one 5 1/4" drive is used, the strap connecting pins 2 to 15 (DS0) is the only strap which must be connected.

If more than one 5 1/4" drive is used, the appropriate drive select strap should be connected on each drive. In addition, when multiple 5 1/4" drives are used, be sure that the strap connecting pins 6 to 11 (MX) is not connected.

The other straps should remain unconnected. The last drive on the cable should have the terminator resistor pack (2F) plugged in only if no 8" drive has been terminated.

OMNIDISK Technical Reference Manual TEAC

TEAC 55 A,B,C,D,F,G

Required jumpering scheme for TEAC55 A,B,C,D,F,G type 5.25" floppy disk drives used under OmniDisk.

HS \* \* UR \*==\* PM \*==\*

DS0 \* \* ML \* \*

DS1 \* \* IU \* \*

HM \*==\* HL \* \*

DS2 \* \* SM \*==\*

DS3 \* \* UO \* \*

MX \* \* RE \* \*

In addition the LP jumper must be installed for TEAC drive types 55G and 55GVF.

8.2 8" Floppy Drive Configuration

SHUGART 800/801 (STEP8=6, LOAD8=35, SETL8=8)

Designator Description OPEN SHORTED

DS1-4 Drive Select ONE ONLY \*

T1,3,4,5,6 Terminators for MPX inputs LAST DRIVE

T2 Terminator for RHL X

RR Radial Ready X

RI Radial Index X

R Ready X

800/801 Hard Sector Option 801\* 800\*

I Index X

S Sector Output X

DC Disk Change X

HL Stepper Power from HL X

WP Write Protect Allowed X

NP Write Protect Not allowed X

D Alt input - in use X

A Radial HD Load X

B Radial HD Load X\*

X Radial HD Load X

C Alt input - HD Load X\*

Z In Use from DS X

Y In Use from HL X

DS Stepper Power from DS X

8,16,32 Hard sector options Not Used

2-18, even Nine Alt-I/O pins All Open

D1,2,4,DDS DS option All Open

Note: \* - indicates a change from a factory jumper.

function Disk Controller page #

OMNIDISK Technical Reference Manual QUME

QUME - 8"

Jumpering scheme for QUME type 8" floppy disk drives used under OmniDisk.

\* \* \* \* \* \* \*

| | | | | |

\* \* \* \* \* \* \* C D DC 2S

A B X R I Z H \*--\* \* \* \* \* \*--\*

L

SIEMENS FDD 100-8

Designator Description OPEN SHORTED

RAD SEL 0-3 Drive Select ONE ONLY\*

BINARY SELECT All Open

RAD STEP Radial Step 1 2

RR Radial Ready X

G Stepper Power X\*

H Stepper Power X\*

F Stepper Power X

J Radial HD Load X

K Radial HD Load X

L Radial HD Load

HS Hard Sectored

SS Soft Sectored

8/16 Hard Sector size

U Activate LED

S Activate LED

R Activate LED

H Activate LED

SE Data Separator

TE Data Separator

V Write Protect Not Allowed

E Write Protect Allowed

C,A,B,M

D

Note: \* - indicates a change from a factory jumper.

OMNIDISK Technical Reference Manual SIEMENS FDD 200-8

SIEMENS FDD 200-8

Designator Description OPEN SHORTED

RAD SEL 0-3 Drive Select ONE ONLY\*

BINARY SELECT All Open

RAD STEP Radial Step 1 2

RR Radial Ready X

G Stepper Power X\*

H Stepper Power X\*

F Stepper Power X

J Radial HD Load X

K Radial HD Load X

L Radial HD Load

HS Hard Sectored

SS Soft Sectored

8/16 Hard Sector size

U Activate LED

S Activate LED

R Activate LED

H Activate LED

SE Data Separator

TE Data Separator

V Write Protect Not Allowed

E Write Protect Allowed

C,A,B,M

D

11,12,3,8 X

4,7,13 X

Note: \* - indicates a change from a factory jumper.

OMNIDISK Technical Reference Manual Mitsubishi 2894

Mitsubishi 2894 8-inch Floppy Disk Drive

Designator Description OPEN SHORTED

PJ 1 X

PJ 2 Ready X

PJ 3 Door lock option X

PJ 4 Head load from DS X

PJ 5 Alt input - Head Load X\*

PJ 6 Head load latch X

PJ 7 Stepper power from HDLD or DS X

PJ 8 X

DS1-4 Drive select ONE ONLY

1B-4 Side Select by DS X

RR Radial Ready X

RI Radial Index X

R Ready Output X

I Index Output X

2S Two-Sided status output X

DC Disk Change option X

WP Write protect allowed X

NP Write protect not allowed X

A DS with HL and DS X

E No function X

Z In use LED from DS X

Y In use LED from HL X

S1 Side Select using DIR SEL X

S2 Standard side select X

S3 Side Select using DS X

STM1 Stepper power from INTF X

STM2 Stepper power normal X

SF Internal switch filter X

NSF without switch filter X

F In use LED/in use option X

HUN HD unload w/o delay X

HUD HD unload with delay X

DL1 Door lock by in use X

DLH Door lock by head load X

DLM1 Door lock with HDLD and DS X

DLM2 Door lock with HDLD X

Terminator dip should only be installed in last drive.

Note: \* - indicates a change from a factory jumper.

OMNIDISK Technical Reference Manual Shugart

SHUGART 850/851 (STEP8=3, LOAD8=35, SETL8=15)

Designator Description OPEN SHORTED

DS1,2,3,4 Drive Select ONE ONLY \*

1B,2B,3B,4B Side Select using DS ALL

RR Radial Ready X

RI Radial Index X

R (SHUNT 4F) Ready X

2S Two - Sided X\*

850/851 Hard Sector Option 851\* 850\*

I (SHUNT 4F) Index X

S (SHUNT 4F) Sector Output X

DC Disk Change X

HL (SHUNT 4F) Stepper Power from HL X

WP Write Protect Allowed X

NP Write Protect Not allowed X

D Alt input - in use X

M Multi-Media Option X

DL Door Lock Option X

A (SHUNT 4F) Radial HD Load X

B (SHUNT 4F) Radial HD Load X\*

X (SHUNT 4F) Radial HD Load X

C Alt input - HD Load X\*

Z (SHUNT 4F) In Use from DS X

Y In Use from HL X

S1 Side Select Option X

S2 Standard side select X

S3 Side Select Option X

TS,FS Data Separator Option TS FS

IW Write Current Switch X

RS Ready Standard X

RM Ready Modified X

HLL Head Load Latch X

IT In use terminator X

HI HL or in use to in use ckt X\*

F M2FM X

AF FM or MFM X\*

NF M2FM X\*

Note: \* - indicates a change from a factory jumper.

SHUGART 850/851

\*

|

\*

M

\*

HLL

\*

\*

DL

\*

\*

Y

\*

RS RM\*=\*

\*

\*

DS AF NF

\* \* \*=\*

\* \*D \*

HI \*=\*C |IT

\*

\*S1

\*=\*S2

\*S3

\* \*

| |FS

IW \* \* \*

\*=\*Z

851 \*=\*HL

\* \*=\*A

\*=\* \* \*B

850 \*=\*X

\*=\*I

\*=\*R

\*=\*S

DS1:DS2:DS3:DS4

\* \* \* \* \* \* 2S | DC

|

\* \* \* \* \* \* \* \* \* \*

\* \* \* 1B 2B 3B

4B

OMNIDISK Technical Reference Manual HARD DISK Configuration

8.3 Winchester Drive Configuration

Inserted here is a list of parameters for a few hard drives, Due to hardware diversity switch options where not attempted to be described.

Buffered or ramped seek is a method of positioning by stepper motors that is used on hard disks. This mode allows the controller to send out all the step pulses very quickly (35us). The drive then determines the distance to travel to the target cylinder and ramps the heads up maximum seek speed by varying the step pulse width. Near the end of the seek, the heads are slowed down in time to come to rest on the final target track. After a pause for head settling, the drive READY signal goes true.

Drives that use buffered seek may be used in straight step mode, however, all will be slower. For example a CMI drive averaged 93ms per seek in 3ms/step vs. 20ms per sec in buffered mode. On some hard disk units a switch/shunt option may need to be set to use ramped seek.

OMNIDISK Technical Reference Manual HARD DISK Types

heads cyls precomp seek type/speed

----- ------ ------- ---- ----------

CMI

CMI5616 6 256 150 Buffered seek

CMI5619 6 306 150 Buffered seek

Control Data Corp

CDCWREN5 697 Buffered seek

Micro Science

HH612 4 306 150 Buffered seek, 1/2 Height

IMI

IMI5006 2 306 150 Buffered seek

IMI5012 4 306 150 Buffered seek

IMI5018 6 306 150 Buffered seek

Shugart

SA712 4 306 150 Buffered seek

SA1004 4 256 128 Buffered seek

Syquest

SQ306RD 2 306 150 Buffered seek

SQ312RD 2 612 180 Buffered seek REMOVABLE, 1/2 height Seagate

ST206 2 306 Buffered seek

ST506 4 153 3ms step RATE

ST406 2 306 150 Buffered seek

ST412 4 306 150 Buffered seek

ST419 6 306 Buffered seek

Tandon

TM501 2 306 150 Buffered seek

TM502 4 306 150 Buffered seek

TM602E 4 153 75 3ms step RATE

TM603S 6 153 75 3ms step RATE

TM603SE 6 230 75 3ms step RATE

Micropolis

1302 3 830 400 Buffered seek

1303 5 830 400 Buffered seek

1304 6 830 400 Buffered seek

BASF (Agfa Gevert)

6188R 4 460 230 3ms step RATE

Miniscribe

3212 2 306 150 3ms step RATE

3425 4 306 150 3ms step RATE

6032 3 1024 150 Buffered seek

6053 5 1024 150 Buffered seek

6085 8 1024 150 Buffered seek

8425 4 612 150 Buffered seek

Priam

V170 7 1024 320 Buffered seek

9 - Utility Programs

9.0 Summary

FMT3 - CP/M FORMAT utility

SETSKEW - A utility to set sector skewing factor

on any device.

BOOT - A utility for simulating COLD boot

condition from CP/M.

TMOV - A test utility for determination of TMA

reliability.

BPROM - Source for boot PROM. Residence is on

OmniDisk if power-on-boot option is used.

LU - A utility to change mapping of physical

devices to logical devices.

DPB - Examine the DPB associated with a drive.

9.0 FMT3 - usage

CP/M format utility.

CP/M FORMAT UTILITY, writes formatting information on disk tracks. Used for floppy, hard and RAM drives. Options are selected by menu.

9.1 LU - usage

This program changes the logical to physical assignments used by the OmniDisk controller.

Use format:

LU 8$$$$ 5$$$$ W$$$$ X$

Where 8, 5, W and X are key letters for 8-inch floppy disk, 5-inch floppy disk, winchester (hard disk) and RamDisk. The $$$$ parameter area following the key letter is used to reference the physical unit address: DS1...DS4 for floppies, and units 0..3 for Winchester disks. For example:

LU 8AB 5CD WE XF

Defines A: & B: as 8 inch floppy, C: & D: as 5 inch floppy, E: as hard disk, and F: as ram disk.

For a system that boots on 5" and then needs to continue running with A: as the hard disk, use:

5AB WC

Configuration as booted:

WA 5CB To Swap A: with C:

or

WA 5BC Old A: is now B:, old B: now C:, old C: now A:

Note that you can not reference physical drives that are not all ready configured in your BIOS, i.e. If you add an 8-inch drive to a 5-inch system you will have to change and then reassemble your BIOS.

References to logical drives other than what you have previously defined in your BIOS are not allowed. A logical unit may not be used more than once. You must have an A:, further this A: must have a same-size system on it as you are running (i.e. 62k/56k swaps will not work). Defining A: as ram disk without a valid system on it will cause a condition cured only by power down.

9.2 SETSKEW - usage

9.3 BOOT - usage

9.4 TMOV - usage

TMOV will test your system's ability to successfully do DMA. If a '\*' appears, it indicates an error and a ' ' (blank space) indicates an end to a pass.

(Note - TMOV.ASM must be patched to direct its output to your console port. Change the console port address to that used for your system and reassemble TMOV.)

TMOV.ASM is included in your distribution disk. If you are using revision 3.4 or better of the MPUZ monitor program then you can simply enter "TMOV" at the monitor prompt.

Refer to the reference document on NEWZ-3 for more information.

Either way, you must pass the TMOV test before enabling the DMA feature in your BIOS or device drivers.

9.5 BPROM - usage

9.6 COPY - usage

9.7 MCCONV - usage

9.9 SYSGEN5 - usage

SYSGEN - SYSTEM GENERATION PROGRAM

Universal.. Works with RAMDISK, HARDDISK, or FLOPPY. Prompts for source drive, destination drive. May be called with parameter indicating file from which to get system.

9.10 SHOWDPB - usage

DPB is a utility used to display the current DPB for a drive. DPB [drivename:] or default to current drive.

9.11 FORMAT, MS-DOS version

Writes formatting information on disk tracks.

Used for floppy, hard and RAM drives running under MS-DOS.

format [D:][/O1][/O3...]

The following options are available for the MS-DOS utility format.

D: - Specify drive name, omit if default drive is to be

used.

V - Do not query for volume name.

D - Use both sides of disk.

9 - Use 9 sector IBM format for 5.25" floppy.

Default format for 5.25" floppy is 8 - sector type.

S - Transfer system image.

1 - Use one side of disk.

O - Set skew rate for hard disk

For example to format and transfer system to 8"

Double-sided disk for MS-DOS in B: the command line would be:

format B: \s \d

To format a single-sided 8" in C: without an MS-DOS system image:

format C: \1

Component List

Designation Type Description

U1,U29,U49 74LS32

U2 74LS04

U3,U6,U27 74LS08

U4 74LS145 Floppy Select DEMUX

U5 74LS393 Clock divisor

U7 74LS153 Clock MUX

U8 7404 Oscillator

U9,U56 7416

U10 74LS02

U11 74LS04

U12 74LS240 Latch/Mux for FD signals

U13 upC765A or 8272A FDC

U14 8085AH CPU

U15 74LS245 HD port buffer, data

U16 74LS244 HD port buffer, address

U17-U21 unused

U22,U38 74LS74

U23 74LS244 S2 Port, status Port

U24 74LS195 Precompensation Shift-reg.

U25 74LS74

U26 74LS02

U28 FDC 9216 B Multimode Data Separator

U30,U39,U52 74LS374

U31 74LS373 8085A address latch

U32 27C64 Firmware EPROM

U33 HM 6116-2 2k x 8 Buffer

U34 HM 6264-12 8k x 8 Buffer

U36 74LS27

U37 74LS10

U40-U45 74LS161,74LS163 24-bit DMA address counter

U46 74S472 512x8 PROM Boot firmware

U47 74LS00

U48 74LS155

U50 74LS139

U51 74LS138

U53 74LS393 Internal DMA counter

U54 74LS175 DMA timing latch

U55 74LS473 IEEE 696 DMA priority

arbitration

U58,U60 74LS682 Port Address comparator

U57,U59,U61 74LS244 24-bit DMA address buffer

U62 74LS374

U63,U64 74LS244 DMA data buffers

U65 74LS374

U66 74LS374

U67 TBP 18S030 32x8 PROM, DMA state machine

U68 74LS161 counter, DMA state machine

U69 74LS74

U70 74LS04

U71 74LS08

Designation Type Description

S1-3 8 switch dip

C1,C4,C5,C9-11 0.1 uF Bypass

C2 22 pF

C3 0.001 uF

C6,C12,C13 1.0 uF Tantalum bypass

R1,R3 1.0k

R2 470 ohm SIP 6-pin

R4 4.7k SIP 10- pin

R9 4.7k SIP 5 - pin

R10 4.7k SIP 8 - pin

R5,R7 1.2k

R6,R8 680 ohm

VR1,VR2 7805,LM340T-5 or equiv. 5V regulator

X1 16.000 mhz Xtal

OMNIDISK Technical Reference Manual State Prom

APPENDIX D: State PROM table for DMA operation

;

; USED TO GENERATE PSYNC, PSTVAL

;

; PDBIN, WR AND RD SIGNALS

;

DB 0F6H,0F6H,0F6H,0F6H ;00-03

DB 0F6H,0F6H,0F6H,0F6H ;04-07

DB 0F6H,0F6H,0F6H,0F6H ;08-0B

DB 0F6H,0F6H,0F6H,0F6H ;0C-0F

DB 076H,07EH,075H,077H ;10-13

DB 07EH,076H,06EH,02EH ;14-17

DB 076H,056H,075H,077H ;18-1B

DB 056H,076H,052H,012H ;1C-1F

OMNIDISK Technical Reference Manual APPENDIX F

8085/Z80 Drivers

;

; ?ONELIST

; send one byte to omnidisk listed after call to

; ?onelist, continue execution after byte

;

?onelist:

xthl ! mov a,m ! inx h ! xthl

;

; ?SENDCMD

; send one byte to omnidisk

; passed in <a> register

;

?sendcmd:

push psw ; Save byte to send later

sendcmd1:

in omnistat

ral ! jc sendcmd1 ; Wait for ready

pop psw

out omnidata ; Output it

ret

;

; ?CMDLIST

; Issue a list of commands just after call instruction

; continue execution after list OMNIDISK Technical Reference

; Manual APPENDIX F

;

?cmdlist:

xthl

mov c,m ; C=list length

inx h

cmdlist1:

mov a,m ; get one byte

inx h

call ?sendcmd ; Send it to controller.

dcr c

jnz cmdlist1 ; More commands to send?

xthl ; Set return address just past list

ret

;

; INCHAR

; accept one data from OMNIDISK to <A>

;

?inchar:

in omnistat

rar

jnc ?inchar ; Wait until ready

in omnidata ! ret ; Grab data

OMNIDISK Technical Reference Manual APPENDIX I

;

; SOURCE IS BPROM.ASM

; LAST CHANGE 17 MAY 84

;

; THIS PROGRAM RESIDES ON A 512x8 PROM ON THE OMNIDISK CONTROLLER

; ITS PURPOSE IS TO GET A SYSTEM UP FAST.

;

; DURING THE FIRST PART OF ITS EXECUTION THE PROGRAM APPEARS

; ALL OVER MEMORY DURING READ CYCLES BECAUSE THE OMNIDISK

; ONLY USES THE LOW 9 BITS OF THE 24-BIT MEMORY ADDRESS.

; "PHANTOM" IS ASSURTED TO DISABLE OTHER MEMORY THAT MAY BE

; PRESENT. "PHANTOM" IS NOT ASSURTED DURING WRITE

; CYCLES -- THIS ALLOWS THE PROGRAM TO COPY ITSELF TO REAL

; MEMORY AND THEN TO CAUSE THE "PHANTOM OF THE BOOT" TO RESET.

;

; UNTIL OUR "PHANTOM" IS RESET WE CAN ONLY WRITE TO REAL MEMORY.

;

SWAP EQU 0FDH ;SWAP PORT FOR COMPUPRO (tm) 8085/8088

OC$BOOT EQU 0 ;BOOT SYSTEM

OC$SWRT EQU 1 ;WRITE SYSTEM (INVERSE OF BOOT)

OC$MODE EQU 2 ;DMA/IO MODE

OC$GLDS EQU 5 ;GET LOGICAL DEVICE SET

OC$RBPH EQU 7 ;RESET BOOT PHANTOM

OC$UNIT EQU 9 ;SELECT (LOGICAL) UNIT

OC$TRAK EQU 10 ;SELECT TRACK

OC$RECD EQU 11 ;SELECT (LOGICAL) RECORD

OC$DADR EQU 12 ;SET DMA ADDRESS

OC$READ EQU 13 ;READ

OC$WRIT EQU 14 ;WRITE

OC$HEAD EQU 15 ;SET HEAD (NOT USED, ANYWHERE)

OC$GDPB EQU 17 ;GET DPB (GET CP/M DPB FROM OMNI)

OC$GENS EQU 22 ;GET GENERAL STATUS

OC$EXTS EQU 23 ;GET EXTENDED STATUS

OMNIDATA EQU 0A0H ;<== DATA PORT FOR OMNI CONTROLLER

OMNISTAT EQU OMNIDATA+1 ;STATUS PORT FOR OMNI

INREADY EQU 01H ;\*\* OMNI HAS DATA TO SEND

OUTREADY EQU 80H ;\*\* OMNI BUSY

;

; THE FOLLOWING CODE HAS A LITTLE TRICK IN IT:

;

; ON RESET THE 8085 STARTS AT 0000. THE 200H BYTE BOOT PROM

; IS MAPPED INTO ALL ADDRESS, REPEATING AT 200H INTERVALS.

;

; THE CODE FROM THE 16 NOP'S TO THE FIRST TIME THE "JNZ L1" IS

; EXECUTED, RUNS AT 000x. THE JMP TAKES US TO "L1" IN THE

; LOW 200 ADDRESS RANGE AND WE CONTINUE RUNNING AT 2xx UNTIL

; WE JMP TO THE BOOT SECTOR WHICH WE LOAD AT 0000.

;

;MUST BE ON 100H BOUNDRY FOR LOOP AT "L1"

ORG 200H

;16 NOP'S - SOME CPU BOARDS NEED THIS DELAY

;(THE LOOP AT "L2" NEEDS THE FIRST NOP)

DW 0,0,0,0,0,0,0,0

IN SWAP ;WAKE UP 8088 IF USING COMPUPRO (tm) 8085/8088

LXI H,2FFH ;GET READY TO MOVE 3FF..200

L1: MOV A,M ;MOVE PROG TO REAL

MOV M,A ;MEMORY "UNDER" PROM

INR H

MOV A,M ;INIT...

MOV M,A ; ... 3xxH AREA

DCR H

INR L

JNZ L1 ;IF NOT DONE MOVING TO REAL MEMORY

SPHL ;200H ==> SP

IN OMNIDATA ;READ&TRASH TO RESET ANY TRASH IN REG

MVI A,OC$RBPH ;RESET BOOT PHANTOM...

OUT OMNIDATA ;...TO OMNI

;

; NOW WAIT FOR OMNI TO HONOR REQUEST TO

; RESET THE PHANTOM OF THE BOOT

;

; THE METHOD WE USE IS QUITE SIMPLE. WE STORE A

; NON-ZERO (OC$RPHP) INTO THE FIRST BYTE OF THIS CODE

; AND THEN COMPARE THAT FIRST BYTE AGAINST THE VALUE

; WE JUST STORED. THE STORE WILL ALLWAYS WORK -- BUT

; UNTIL THE BOOT PHANTOM RESETS, THE MEMORY FETCH

; PART OF THE "CMP M" WILL FETCH THE ORIGINAL BYTE.

; WE CONTINUE UNTIL WE FETCH THE BYTE WE STORED.

;

;NOTE THAT <HL> = 200H

L2: MOV M,A ;THIS STORE ALLWAYS WORKS BUT

CMP M ;IF BS PHANTOM NOT RESET, THIS FETCHES PROM

JNZ L2 ;IF BOOT PROM IS NOT RESET

L3: CALL ?CMDLIST

DB 5 ;(LENGTH OF LIST)

DB OC$UNIT,0 ;SELECT WHATEVER IS THE A: UNIT

DB OC$BOOT,1 ;READ ONLY FIRST SECTOR OF BOOT

DB OC$GENS

LXI H,0000 ;READ BOOT SECTOR INTO LOCATION 0000

CALL ?GETDAT

ORA A ;GENERAL STATUS

JNZ BOOTER ;IF ERROR GO DELAY AND RETRY UNTIL IT WORKS

;

; BOOT SECTOR WAS READ OK -- BUT IS THIS AN

; 8080 BOOT OR AN 8086 BOOT?

;

; CHECK FIRST INSTRUCTION. IF IT IS

; AN 8086 JMPS THEN THIS IS AN 8086 BOOT, ELSE

; WE ASSUME IT IS AN 8080 BOOT.

;

MOV L,H ;<HL> = 0000

MVI A,0EBH ;8080 XCHG OR 8088 JMPS INSTRUCTION

CMP M

JZ BOOT86 ;IF 8086/8088 BOOT IN PROGRESS

PCHL ;ELSE...8080 BOOT -- GO FIRE IT UP

;

; FAILURE READING BOOT SECTOR

;

; WAIT SEVERAL SECONDS BEFORE TRYING AGAIN

;

BOOTER: LXI H,0

MVI A,30 ;FOR DELAY COUNTER (=5 SEC ON 6MHZ 8085)

L4: DCR L ;\*\*

JNZ L4 ;\*\* ON 8085 THESE LOOPS TAKE 1,118,464

DCR H ;\*\* T-STATES OR ABOUT 1/CLOCK SEC

JNZ L4 ;\*\*

DCR A

JNZ L4

JMP L3 ;TRY AGAIN AFTER DELAY

;

; ?CMDLIST

; Issue a list of commands just after call instruction

; continue execution after list

;

?cmdlist:

xthl

mov c,m ; C=list length

inx h

cmdlist1:

mov a,m ; get one byte

inx h

call ?sendcmd; Send it to controller.

dcr c

jnz cmdlist1 ; More commands to send?

xthl ; Set return address just past list

ret

;

; GETDAT

; Get data block from OMNIDISK into memory <HL>

;

?getdat:

in omnistat

rrc

jnc ?getdat ; Wait until data available

ani 40h ; What do we have here: DATA –or- GEN$STAT

in omnidata ; Grab data

rz ; We have: GEN$STAT

mov m,a ; We have: DATA

inx h

jmp ?getdat

;

; ?SENDCMD

; send one byte to omnidisk

; passed in <A> register

;

?sendcmd:

push psw ; Save byte to send later

sendcmd1:

in omnistat

ral

jc sendcmd1 ; Wait for ready

pop psw

out omnidata ; Output it

ret

;

DB 'Copyright (c) 1984 W/W Componts Inc.'

DB ' 17 MAY 1984'

;

; THE FOLLOWING 8088 CODE IS USED IF WE HAVE BOOTED TO

; A COMPUPRO (tm) 8085/8088 CPU.

;

; THE 8085 "IN SWAP" AT THE FIRST OF THIS PROM CAUSES THE 8088

; TO BEGIN AT ITS RESET ADRESS, 0FFFF0H. THAT TAKES US TO 3F0H

; IN THIS PROM BECAUSE THE HIGH-ORDER BITS OF THE ADDRESS ARE

; IGNORED. THE "JMP FAR" INSTRUCTION FOUND THERE SETS <CS> TO

; ZERO AND THE PROGRAM COUNTER TO 2F0H. NOW AT 2F0H THE 8088

; EXECUTES "IN AL,SWAP", CAUSING THE 8085

; TO START RUNNING AGAIN.

;

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; \*\*\* BEFORE THE 8088 SWITCHES, IT HAS TIME \*\*\*

; \*\*\* TO PRE-FETCH THE JMPS 300H INSTRUCTION \*\*\*

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;

; WHEN THE 8088 IS STARTED AGAIN IT EXECUTES THE PRE-FETCHED

; "JMPS 300H" AND APPEARS TO START EXECUTION AT 300H.

;

ORG 2F0H

DB 0E4H,SWAP ;8088 IN AL,SWAP

DB 0EBH,00EH ;8088 JMPS 300H

ORG 2FEH ;\*\* 8080 CODE USED IF 8086 BOOT ON DUAL

BOOT86:

IN SWAP ;\*\* PROCESSOR BOARD. BY DOING IT HERE,

;\*\* THE 8080 WILL RESTART AT 300H IF EVER

;\*\* THE 8086 WANTS TO SWAP BACK TO IT.

;

; 8086 RESTART VECTOR

;

ORG 3F0H

DB 0EAH ;8088 JMP FAR ...

DW 2F0H,0 ;8088 JMP FAR DEST & CODE SEG

; IF WE HAVE BOOTED TO A 8086, THE "IN AL,SWAP" WILL DO NOTHING

; AND WE CAN EXECUTE "REAL" 8086 CODE WHICH WILL BE BURNED INTO

; THE PROM STARTING AT 300H.

END