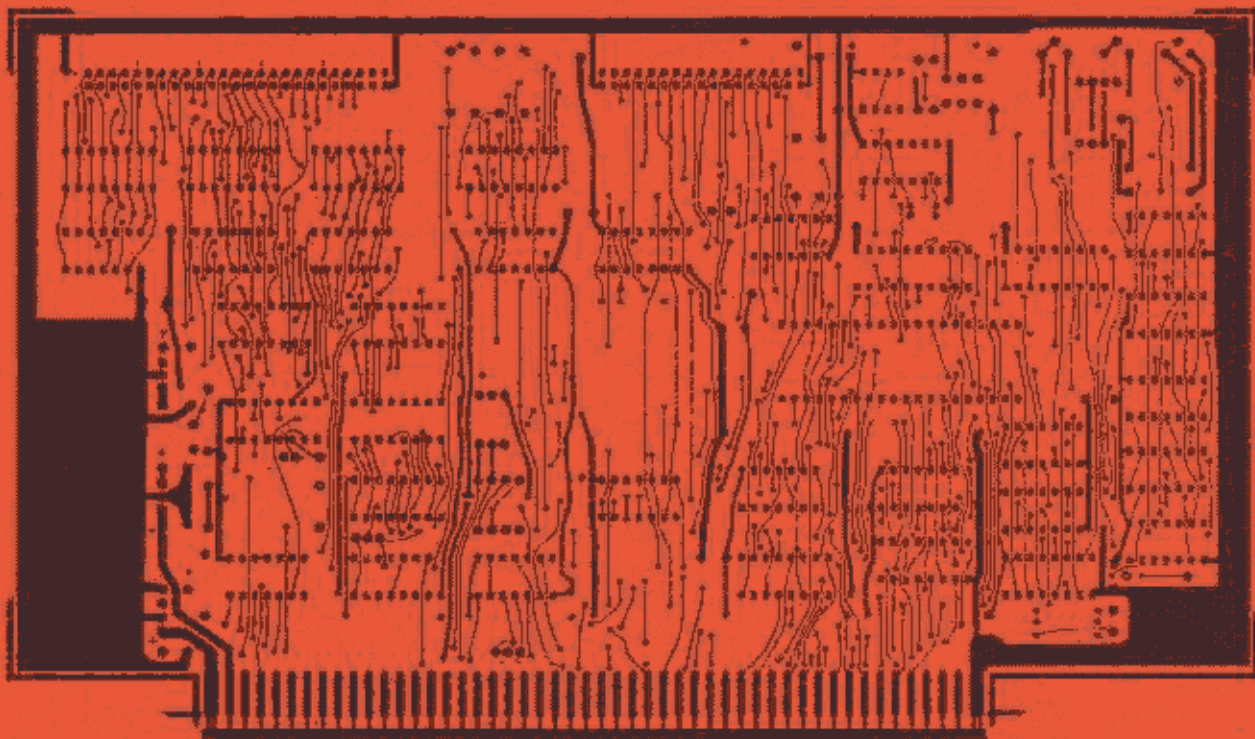


XOR-DSK

OPERATION MANUAL



- 2 OR 4 MHz 8080 or Z80 OPERATION
- PROGRAMMABLE WRITE PRECOMP.
- SINGLE OR DOUBLE SIDED
- TRANSPARENT DENSITY SELECTION
- TWO PHASE LOCK LOOP CIRCUITS



DEALER:
U.S. MICRO SALES
16392 Assembly Lane
Huntington Beach, CA 92649
(714) 898-1492
CUSTOMER SERVICE:
(714) 898-5525

TABLE OF CONTENTS

INTRODUCTION	1
GENERAL DESCRIPTION	1
CAPACITY	1
COMPATIBILITY	1
DRIVE SELECT OPTIONS	1
MINI OPERATIONS	1
DRIVE NOTES	2
STANDARD 8" DRIVES	2
MINI DRIVES	2
REV D DISK CONTROLLER JUMPERING FOR VARIOUS SIZES DRIVES	3
RESISTOR CHANGE ON DISK CONTROLLER BOARD FOR 5 1/4 " DRIVES	3
TROUBLESHOOTING	4
DISK CONTROLLER CHECKS AND ERROR MESSAGES	4
DISK UTILITIES	5
DSKTYP	5
DSYSGEN	5
D	5
DDT	5
ED	6
ASM	6
LOAD	6
STAT	6
PIP	6
SUBMIT	7
XSUB	7
MACRO.LIB & Z80.LIB	7
BLKSYS61.SUB	7
IO.ASM	7
FLOP.ASM	7
BLK.ASM	7
DATA.ASM	7
CBOOT.ASM	7
BLKEQU61.ASM	8
MDISK	10
MDISK - MODIFICATION OF 64K BOARD FOR MDISK USE	10
DSYSGEN EASY METHOD OF TRANSFERRING OPERATING SYSTEM TO DISKS	11
MODIFYING AND GENERATING CP/M SYSTEMS	12
DETAILED DSYSGEN OPERATING INSTRUCTIONS	15
DDUMP	18
OPERATION	18
FILL	18
EDIT	19
SERVICE POLICY	20
DSK REV D PARTS LIST	21
DSK REV D PARTS LIST (CON'T)	22
PART LOCATING DIAGRAM	23

SCHEMATIC

INTRODUCTION

The world of 'Double Density' disc controllers is one that has been long in coming to the general Microcomputer marketplace. It is not without good reason. There are difficulties in data recording and retrieval encountered with this method that challenge the state of the art.

Testing a controller to respond within certain specifications is a rather strait-forward task. Being assured that similar conditions will be supplied to the controller in the field is no mean task. To make things worse the instabilities that can occur as a result of poor media (tape surface), head design and alignment, and installation environment are usually beyond the ability of most end users to detect and correct. To assist in the resolution of these possible difficulties, we have gone one step further than most of our competitors and provided additional hardware testing devices and software aids that can help point to the nature of problems. This allows efficient corrective service to be implemented.

We are confident that the XOR advanced design controller is the finest available today. It is the result of many years of constant engineering development, and refinement taking advantage of the many advancements in the state of the art. We believe it will prove to be the most reliable S-100 controller on the market.

The design of the XOR disk controller is based on the use of the Western Digital WD-1795 floppy disk formatter LSI, in association with the WD-2143-03 Four phase clock generator and the WD-1691 Floppy support logic chips. The controller features:

1. Soft sector IBM format capability.
2. Automatic track seek with verification.
3. Single and double density formats.
4. Single/multiple sector read.
5. Selectable 128, 256, 512, 1024 byte sector lengths.
6. Single/Multiple sector read.
7. Double buffering of data 8 bit bi-directional bus for Data, Control, and status.
8. Programmed data transfer.
9. Inverted data bus.
10. Write pre-comp.
11. Side selection output.
12. 8 in and 5 1/4 in mini floppy operation.

GENERAL DESCRIPTION

The XOR disk controller is designed to replace Single Density disk controllers in most S-100 applications where increased data storage capacity is required.

CAPACITY

The XOR disk controller can provide up to 2.4 meg storage using two double density/double sided drives. Because of the increased data density, double density data transferred from diskette to user memory is accomplished at twice the rate of single density operations.

COMPATIBILITY

Data transfer is accomplished via programmed I.O. methods which allow it to operate with all Static and most Dynamic memory. The controller can switch, under software or hardware control, to single density mode so that IBM 3740 compatible data can be read and written.

DRIVE SELECT OPTIONS

As shipped the XOR disk controller is configured as follows:

Drive A, and B, are set up for 8 in operation.
Drive C, and D, are set up for 5 in operation.

MINI OPERATIONS

Both 8 in and 5 in drives can be connected to the controller at the same time. Read, Write, and Copy operations can be accomplished between any of the 4 drives, in any sequence.

DRIVE NOTES

Using the XOR DFOCO program the following optional drive configurations are available:

STANDARD 8" DRIVES

FORMAT B:

Standard 3740 IBM single density/single side 128 bytes 26 sectors per track.
Capacity is 241K.

DFORMAT B:

128 bytes per sector 51 sectors per track, double density, single sided.
Capacity is 474K.

DFORMAT B: SIZE 512

512 bytes per sector 16 sectors per track, double density, single sided.
Capacity is 596K.

DFORMAT B: SIZE 512 DBL

512 bytes per sector 16 sectors per track, double density, double sided.
Capacity is 1212K

MINI DRIVES

DFORMAT C: MINI 48

128 bytes per sector, 30 sectors per track, double density, single sided.
Capacity is 169K.

DFORMAT C: MINI 48 SIZE 512

512 bytes per sector, 10 sectors per track, double density, single sided.
Capacity is 181K.

DFORMAT C: MINI 48 SIZE 512 DBL

512 bytes per sector, 10 sectors per track, double density, double sided.
Capacity is 362K.

DFORMAT C: MINI 96 SIZE 512

512 bytes per sector, 10 sectors per track, double density, single sided.
Capacity is 366K.

DFORMAT C: MINI 96 SIZE 512 DBL

512 bytes per sector, 10 sectors per track, double density, double sided.
Capacity is 750K.

REV D DISK CONTROLLER JUMPERING FOR VARIOUS SIZES DRIVES

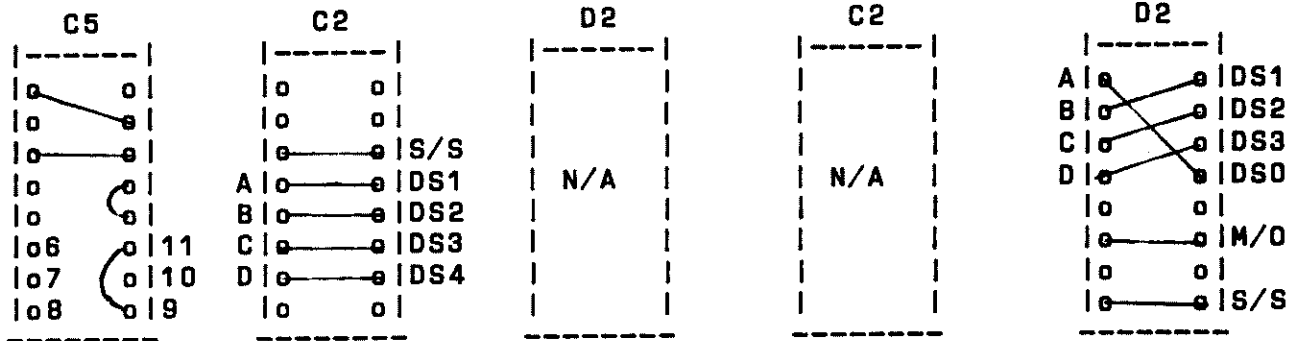
DRIVE
SELECT
BLOCK
C5

8" DRIVE
HEADER
C2

5 1/4" DRIVE
HEADER
D2

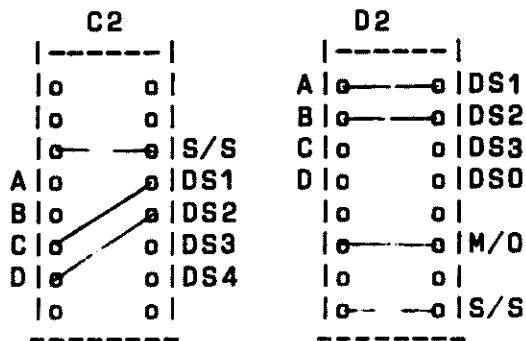
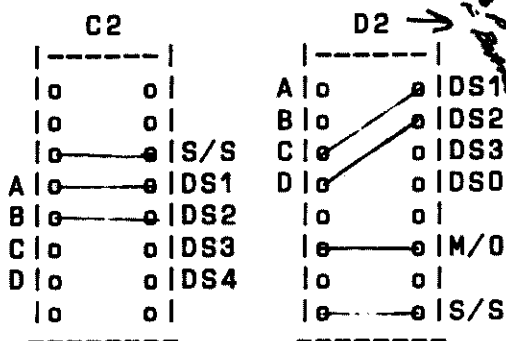
ALL 8 INCH

ALL 5 INCH



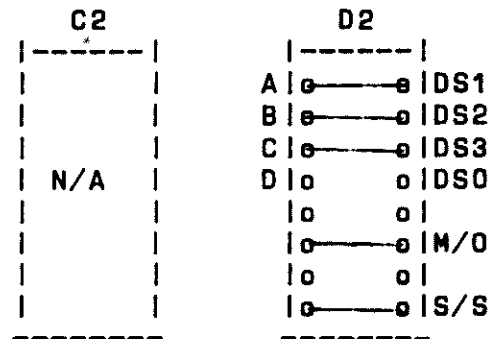
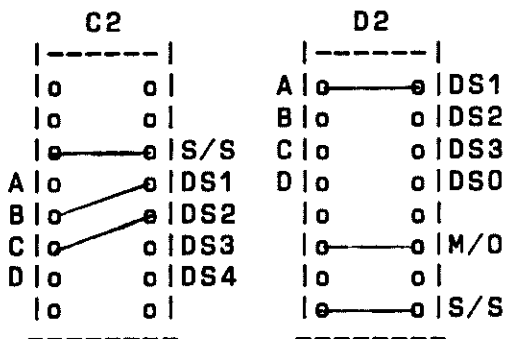
A & B = 8" - C & D = 5 1/4"

A & B = 5 1/4" - C & D = 8"



A = 5 1/4" - B & C = 8"

A, B, C = 5 1/4"



NOTE: S/S = SIDE SELECT
M/O = MOTOR ON

RESISTOR CHANGE ON DISK CONTROLLER BOARD FOR 5 1/4" DRIVES

It is necessary to change resistor R20 next to the 555 IC from a 39K resistor to a 120K resistor when using any 5 1/4" drives.

TROUBLESHOOTING

DISK CONTROLLER CHECKS AND ERROR MESSAGES

The XOR ver 1.85 prom has internal fault messages which will be displayed on the screen if the following occurs:

For no system "NO SYSTEM" will be displayed, for a Read ID or CRC error, DISK ERROR will be displayed. Try another disk with a known good system.

If you attempt to boot up with the door open the message "DRIVE NOT READY" will be displayed.

The error code 10 means that READ ID NOT FOUND, either the controller cannot read a valid sector on the disk or it is not there in the first place.

The error message 08 means that sufficient time has not been given for the head to load and to read a valid track, generally you can reboot successfully.

CRC error means that the proper Cyclical Redundancy Checks have not been made, this can be caused by bad diskette, malfunction in the controller, or incorrect speed of the drive.

In XOR systems with the Tandon thin line drives, the speed can be checked by looking at the large belt spool on the side of the drive under a florescent light. The strobe lines should be stationary. Refer to picture for the location of the motor speed adjustment.

An error code of DRIVE NOT READY can be caused by not enough time for the motor to get up to speed. This is sometimes caused by belt slippage. Belt dressing from your auto parts store that is usually used on slipping car fan belts, will cure this problem. Remove the belt and place on a piece of newspaper. Spray the inside of the belt and allow to dry before replacing.

There are no user adjustments to the controls on the board that would normally have to be adjusted. In the event that qualified technical support personnel are available to work on the controller with adequate test equipment, we have provided the following comments. We can not be responsible for warranty repair on any boards that have been tampered with.

1. Check all on board regulated voltages.
2. Check for 4 MHZ clock at pin 7 of G2 74S124, adjust R-27 for the correct frequency.
3. Check for 1.4 Volts DC at pin 2 of G2 74S124.
4. Adjustment for 200 ns Precompensation is as follows:
 - A. Run DFORMAT, and check pin 1,3,5,7 of WD 2143-03 for 200 ns. Adjust R25 as necessary.
 - B. Check pin 6 of WD 1691V for double pulse width, when the head steps above track 43.
5. Note: Pin 1 of both 8 in and 5 in connectors is on the left when facing the board.
6. When using all 4 drives 5 in, remove 8 in header C2.

DISK UTILITIES

DSKTYP

DSKTYP is a disk identification program. The program asks the user which drive to check, once answered the program will print to the console what type of diskette is in the specified drive. The program asks for the drive to be tested and then replies with the type of diskette in the drive. This is used to see if your disks are compatible with each other for functions such as DCOPY in which both disk formats must be the same.

A>DSKTYP

TO EXIT TO CP/M TYPE A "Q" or "C"
SELECT THE DRIVE TO BE TESTED (MUST BE CAP)
THE DISK IN DRIVE B IS 8 IN 512 BYTE SECTOR WITH SKEW

DSYSGEN

DSYSGEN is used to copy a developed system onto the system tracks of a disk (hard disks as well as floppys).

(THE INSTRUCTIONS FOR THE EASY METHOD OF COPYING A SYSTEM FROM DISK TO DISK IS ON PAGE 11 OF THIS MANUAL.)

D

D is a directory utility which will print to the console all files on disk in alphabetical order and their file size. At the end of the directory the total number of files and their total size is also printed to the console. Other drives can be referenced for example:

"D"<CR>, shows a directory of currently logged drive.

"D B:"<CR>, shows directory of files on drive B.

"D B:*.ASM"<CR> shows all the .ASM files on drive B.

DDT

DDT is a utility which has many features one of which allows the user to read .COM and .HEX files into RAM for execution. Under DDT the user may edit, examine, or link .COM or .HEX files or make small routines for test or other purposes. If under DDT you wish to save an edited program, use the CP/M SAVE NN FILENAME,TYPE command. (NN being the number of records to be saved)

(More information concerning DDT can be found in DIGITAL RESEARCH'S "CP/M MANUAL in the DYNAMIC DEBUGGING TOOL (DDT)" Chapter.)

ED

ED is a source editor for all source files.

(More information concerning ED can be found in DIGITAL RESEARCH'S CP/M MANUAL in the chapter "ED: A CONTEXT EDITOR FOR CP/M DISK SYSTEM".)

ASM

ASM is a 8080 source code assembler. This utility makes a

(More information concerning ASM can be found in DIGITAL RESEARCH'S "CP/M MANUAL in the chapter entitled ASM.)

LOAD

LOAD takes the .HEX files developed from a source file and makes a .COM for execution of the file.

STAT

STAT is a utility which will print to the console the size of any file or the amount of storage remaining on the disk.
Example:

```
"STAT"<CR> "STAT B:"<CR>
"STAT *.*"<CR> "STAT B: STAT.COM"<CR>
```

(More information concerning STAT can be found in DIGITAL RESEARCH'S CP/M MANUAL in the chapter USER'S GUIDE FOR CP/M OWNERS)

PIP

PIP is a copy utility which will copy disk to disk from one to all files on the source disk.

Example:

```
"PIP"<CR>
* (This means you are in the PIP program and is a
response from the program.)
```

```
"PIP B:=A:PIP.COM"<CR> Copies the file PIP.COM from the
disk in drive A to the disk in drive B.
```

```
"PIP B:=A:*.*"<CR> Copies all the files from drive A to
drive B.
```

```
"PIP A:=B:*.*"<CR> Copies all the files from drive B to
drive A.
```

(More information concerning PIP can be found in DIGITAL RESEARCH'S CP/M MANUAL in the chapter entitled USER'S GUIDE FOR CP/M OWNERS.)

SUBMIT

SUBMIT is a special utility which allows console input simulation so that long tedious steps which have to be repeated quite often can be done by typing a one line command. Example submit file.

```
DDT
IEXAMPLE.HEX
R
GO
SAVE 1 EXAMPLE.COM
```

XSUB

XSUB extends the power of SUBMIT to include line input to programs as well as the console commands, the XSUB when included must be in the first line of the sub file.

(More information concerning XSUB can be found in DIGITAL RESEARCH'S "CP/M MANUAL in the chapter entitled USER'S GUIDE FOR CP/M OWNERS".)

MACRO.LIB & Z80.LIB

MACRO.LIB & Z80.LIB are the library files for most standard Z80 Macro assemblers.

BLKSYS61.SUB

BLKSYS61 is the file used to develop 8" & 5" disk systems with BLOCKING/DEBLOCKING MULTI SECTORS.

IO.ASM

IO is the basic INPUT/OUTPUT routine used within the BIOS.

FLOP.ASM

FLOP is the driver used for all the floppy drives.

BLK.ASM

BLK is the BLOCKING/DEBLOCKING MULI SECTOR routine.

DATA.ASM

DATA is the common data file which contains all of the necessary tables for the operation of the BIOS.

CBOOT.ASM

CBOOT is the cold boot loader for all systems.

BLKEQU61.ASM is the equate table used for the system. Following is an example of an equate table:

```

;EQU
    TITLE 'XOR STANDARD BIOS VS 2.22 02/20/83'
;
;***** SYSTEM SIZE *****
;
;   MSIZE IS THE EFFECTIVE SIZE OF YOUR SYSTEM AFTER ACCOUNTING
;   FOR CBIOS AND THE HOST BUFFERS. THE VALUE COMPUTED IS USED
;   AS THE NUMERIC VALUE 'NN' IN YOUR MOVCPM SEQUENCE:
;
;       E.G.   MOVCPM  NN *
;
;   MSIZE   EQU   80
;
;*****
;*****
;   SYSTEM GENERATION EQUATES
;
TRUE   EQU   OFFFHH           ;DEFINE VALUE OF TRUE.
FALSE  EQU   NOT TRUE        ;DEFINE VALUE OF FALSE.
;
;***** FLOPPY RELATIVE EQUATES *****
;
ONEFLP EQU   TRUE            ;IF ONE FLOPPY
TWOFLP EQU   FALSE          ;IF TWO FLOPPY
TREFLP EQU   FALSE          ;IF THREE FLOPPY
FORFLP EQU   FALSE          ;IF FOUR FLOPPY
;
MAXI8  EQU   TRUE            ;TRUE IF STANDARD 8" FLOPPY
;
MINI48 EQU   FALSE          ;TRUE IF MINI DISK DRIVE 48TPI
MINI96 EQU   FALSE          ;TRUE IF MINI DISK DRIVE 96TPI
;
SEKRATE EQU   00            ;00 = 3MS, 101 = 6MS, 102 = 10MS, |
;                               QUME, 850 1801 103 = 15MS |
;                               MINIS  |           |
SLIMLINE EQU   FALSE        ;TRUE IF HALF HI DRIVES
;
;***** HARD DISK RELATIVE EQUATES *****
;
ONEHD  EQU   FALSE          ;TRUE IF 5,10 MEG
TWOHD  EQU   TRUE           ;TRUE IF 16,20 MEG
TREHD  EQU   FALSE          ;TRUE IF 24,30 MEG
FORHD  EQU   FALSE          ;TRUE IF 40+ MEG
;
HD      EQU   TRUE           ;TRUE IF WESTERN DIGITAL
;
;***** SYSTEM CONFIGURATION EQUATES *****
;
MMPM   EQU   FALSE          ;IF MP/M SET TRUE
;
MLDR   EQU   FALSE          ;IF M/PM LOADER SET TRUE
XMLDR  EQU   FALSE          ;IF M/PM LOADER W/O BOOT
;

```

```
MDISK EQU FALSE
;
DUAL EQU FALSE
;
DBLOCK EQU TRUE
F256 EQU FALSE
F512 EQU TRUE
MAXHST EQU 512
;
SIDE2BIT EQU FALSE
;
```

```
;TRUE IF MDISK
;TRUE IF DUAL SPINDLE DRIVE (PERSCI)
;TRUE FOR MULTISECTOR OPERATION
;TRUE FOR 256 SECTOR FLOP
;TRUE FOR 512 SECTOR FLOP
;SIZE OF BLOCKING BUFFER
;SIDE TWO COMPARE BIT (NORMALLY OFF)
```

MDISK

MDISK is now with us. - WHAT IS MDISK you ask? It is a kind of cache memory system that is CP/M oriented. It takes the place of the "D" drive and is as large as your pocket book. For all intents and purposes it is a disk drive. Only it stores it's data in RAM. It can be expanded in 48K increments to about 500K. If you do a lot of assembler or compiler work and can think of several hundred better things to do than sit waiting for the processor to complete its task, this may be useful to you.

Typical performance increases over our floppy times will be in excess of 100 per cent. (That means you will only have to sit half as long.) Those of you who are familiar with our system will realize that as we are already the fastest around, this is some real improvement. Comparing it to an Apple or an Orange is a joke.

MDISK will assemble a 20K assembly source program producing a 30K .PRN file and associated .HEX and .SYM files, also reading in two 20K MACRO LIB files in 21 seconds ! Our Apple took 3 1/2 minutes and our Plastic Shack 2 min and 14 secs to do similar tasks. Unfortunately the Orange was in peels by the time we got to it so no testing could be done.

Loading a large .COM file like MBASIC (28k) takes .68 sec. Actually hard to measure. It's fun impressing your friends with it. Call your nearest U. S. Micro Sales office for more information on this option.

MDISK - MODIFICATION OF 64K BOARD FOR MDISK USE

A modification to the memory of XOR systems for high speed operation.

It is necessary first to modify the original 64K board for MP/M use. Next the equate for Mdisk must be turned on in Othe BIOS. The operating system must then be reassembled and sysgened.

As many 48K boards as required may then be added addressing each one appropriately.

When Mdisk signs on it scans available memory and reports Mdisk XXX K available.

The programs running in M disk are assigned in CP/M as the 4th logical storage device. For instance in a dual floppy system, Mdisk will be "D".

In a hard disk system: A = HARD DISK
 F & G = FLOPPIES
 H = MDISK

Remember after using Mdisk be sure to save all programs on Mdisk to either floppy or hard disk to save them.

DSYSGEN EASY METHOD OF TRANSFERRING OPERATING SYSTEM TO DISKS

One of the more common functions you will be performing is the transfer of the operating system from one disk to another. This can be easily accomplished with the Dsysgen program. All XOR systems are organized in the following manner: Tracks 0-1 are reserved for the operating system. Track 2 is the directory. Tracks 3-77 are for data.

The formatting process must be performed first. This process completely blanks a disk and then places sector marks on the disk where the data is later to be written. The operating system is written next using DSYSGEN. REMEMBER, first format a disk with DFOCO, then type DSYSGEN, Answer (N) and (N) to the first two questions that come on the screen. A table will then appear with a sub-line asking "CODE BYTES FOR SOURCE DISK?". Usually you will type 1A, (the system disks we distribute with our systems are 1A) which signifies a 512, single sided, double density source disk. The other common code byte is 9A for a 512 double sided, double density, source disk.

The next question will be enter source drive, which will normally be A, hit a <cr>. A message will appear at bottom of the screen, at this point hit a "return".

The same table will re-appear and you will enter the code byte for the destination disk, usually either 1A, or 9A, and then enter the destination drive, usually B, hit <cr>, and wait for a short period, a complete message should appear, type a control C and you will be back in CP/M.

The preceding procedure has transferred the operating system from A to B. You should now be able to remove the disk from drive B, do a reset, insert it in drive A and again boot up.

If you should ever see the error message, "type of source disk and destination disk do not match, do you wish to continue?" type "N", (this is a fatal error, you will destroy your operating system if you continue). You are probably trying to copy a single sided system to a double sided formatted disk. The program will restart, re-check the formats and try again.

Another item to be noted, all XOR systems are compatible with standard 128 single density, however there is not enough room on the system tracks of a single density disk to place the system. You must always boot up on a Double density disk in "A" drive. Single density files may then be transferred from "B" to "A" with the PIP program.

MODIFYING AND GENERATING CP/M SYSTEMS

The CP/M operating system is composed of several parts: the BDOS (Basic Disk Operating System), the BIOS (Basic Input Output System), the CCP (Command Control Processor), and the BOOT (Cold Boot Loader).

In the U S Micro Sales system, the programs BOOT and BIOS are broken into several parts that are concatenated at the time that the system is built into one large module. The reason this is done is that fewer base modules need to be maintained to support all of the varied disk systems. Also, most larger multi-user modules would not fit in the system under the editor.

This BIOS system is intended to be used with SUBMIT modules that bear the name of the object BIOS produced. For example, BLKSYS61.SUB contains the following:

```
PIP
BLKSYS61.ASM=BLKEQU61.ASM, IO.ASM, CXLST.ASM, FLOP.ASM,
BLK.ASM, DATA.ASM, XBOOT.ASM
```

To concatenate this module, type: SUBMIT BLKSYS61 <CR>.

The PIP program picks out the necessary modules from the disk you have reserved for systems and creates the correct BIOS. The MACRO assembler will then assemble it for you.

Maintaining operating systems is a problem in that often minor "fixes" need to be added to them and having a dozen or so "versions" around means that not all of them will get the latest updates.

One of the greatest advantages of this system is that you only maintain one set of basic modules. In fact, we recommend that you keep only one backup copy of the disk containing these modules. In that way, you can be sure that the software you are building is the latest revision.

Modifications you wish to make should be added by putting conditional assembly flags around the code that is optional and establishing a flag in the main xxxEQU61.ASM header file. This keeps confusion and multiple copies to a minimum. Also, if you need to see if a certain group of files are present you can type: "DIR *.SUB" and just the system building files will appear. This is a real time saver when dealing with a hard disk that can display 4 or 5 screens of directory!

Programs you will need on the system disk to assemble a system:

1. ALL NECESSARYASM
2. CUSTOM BUILT ...SYS.SUB
3. SUBMIT.COM
4. MACRO.COM
5. MACRO.LIB
6. Z80.LIB
7. MOVCPM.COM
8. ED OR POLYVIEW [Polyview is HIGHLY recommended over ED]
9. DSYSGEN.COM

A - MODIFICATION *** (STEP BY STEP INSTRUCTIONS)

The first thing to do is to edit the appropriate xxxEQU61.ASM and set all the equates to suit the configuration of the system you desire.

B - EXAMINATION ***

Next, type the ...SYS61.SUB file and look at the subparts needed to concatenate the final BIOS. Then Type DIR *.ASM and see if all the correct files are present.

C - CONCATENATION AND ASSEMBLY ***

If they are, type SUBMIT ...SYS61 The final product will be a ...SYS61.HEX file.

D - GENERATING CP/M ***

Type "MOVCPM 61 *"

The program MOVCPM.COM will create the BDOS for you. Usually, this needs to be done only once. Spend some time and make several of the common system sizes (e.g. 63K, 62K, 48K, etc.). When the MOVCPM has completed its job, type: "SAVE 34 CPMXX.COM" where XX is the system size.

E - PUTTING SYSTEM ON SYSTEM TRACKS ***

This is the job of DSYSGEN.COM. When you load DSYSGEN, it will ask you if the system has a file on disk. Answer 'N'. Then it will ask you if you want to build a system. This time, answer 'Y'. DSYSGEN will then ask you for the name of the CP/M file and then the name of the BIOS file. It will fetch the data that you have prepared and put it in the correct places in preparation for installation on the disk of your choice.

The next thing that will appear on the screen is a large table which has at the left of it some code bytes that will be written to the last byte of the first sector of the disk chosen.

At this time, the system you have created can be put on any of the disk formats specified in the table. Of course, the hardware must be compatible with the type of media you have picked and the disk must have been formatted with a format compatible with your choice (i.e. you could not put a 512 SKEW system on a 128 byte, double density disk.)

It should be noted that you need not put the system on a disk at this time. In fact, the computer system on which the operating system was created need not even be capable of writing to the drive type that the program you have built is designed for. You can type a control 'C' in answer to the "DESTINATION DRIVE" question and then type: "SAVE xx xxxCPMxx.COM" (the x's are unique system identifiers). The prepared operating system can be transported to the desired system via the floppy disk and read in using the first option on DSYSGEN (The one which asks if the system is a file on disk).

A third method for transferring operating systems is achieved by answering 'N' to both of the first questions DSYSGEN asks. DSYSGEN will then ask for the codebytes of the source disk you are going to take the operating system from. Then it will ask for the drive. From that point on you can proceed just as if you had constructed a system or read it as a file from disk.

The programmer must make sure that the operating system put on the system tracks will function properly. Very little error checking can be done in this process. It is extremely easy to make a "fatal" error. One of the most common is assembling a system that is too large to fit on the system tracks.

The best we can suggest for checking the load size is to go ahead and try to put the system on the disk. The DSYSGEN program will report to the screen the system size, load address, ending address and BIOS jump table address for the current system. Also it will see if the system will fit on the system tracks of the media you have selected.

Use the following as a guide for how large systems may get:

PROGRAM	SIZE	TRACKS REQUIRED	IMAGE LOAD ADR
COLD BOOT	80H	1	900H - 97FH
CCP	800H	16	980H - 117FH
BDOS	E00H	28	1180H - 1F7FH
BIOS	F80H	31	1F80H - 2F00H
	<u>2600H</u>	<u>76</u>	<u>900H - 2F00H</u>

F - SYSTEM TRACKS AVAILABLE ON VARIOUS MEDIA ***

MEDIA TYPE	TRK-0	TRK-1	TRK-2	TOTAL
5" 128 DD	18	30	30	78
5" 512 DD	18	35	35	88
8" 128 SD	26	26	--	52
8" 128 DD	26	51	--	77
8" 512 DD	26	64	--	90

Note that these values are relative to 128 byte blocks. That is to say, if you multiply the total on the right by 128, the result is what DSYSGEN will record on the system tracks.

If your system exceeds the 76 track guideline used in the example for using DDT to check your system size, you will have to adjust the upper boundary accordingly. Simply add the system size to the image base address and look with DDT at the resulting RAM location.

You also can dump the system tracks into RAM after writing them by re-entering DSYSGEN and letting it get the system by specifying the drive at the prompt "SOURCE DISK?". Exit and type: "SAVE 40 (40 may not be enough) xxCPMxx.COM". Remember, the 40 is PAGES so a SAVE 40 = 40 X 256 = 10,240 or 2800H. If your system is very large, you will have to do some math.

G - DESTINATION DISK PARAMETERS ***

After you specify a destination drive, a large menu will fill the screen. You will be prompted to choose the type of media that you wish to write the system to. The DSYSGEN program is quite universal in that it can service all the system types U S Micro supports. Be sure the computer system and media you are writing to are capable of supporting the operating system you are putting on them. No error checking will be done.

Occasionally, a message will appear that says "CODE BYTE FOR OPERATING SYSTEM AND MEDIA DO NOT AGREE, CONTINUE?" This is caused because the formatting program (DFOCO) has installed what it thinks is the correct code byte for this system in the last byte of the boot sector (7FH). DFOCO is usually correct. If you get this message, go over what you are doing once again, you probably will find something wrong. You may override the warning and write the operating system to the disk.

DETAILED DSYSGEN OPERATING INSTRUCTIONS

DSYSGEN is a program designed to put the CP/M, operating systems on the system tracks of a floppy or hard disk mass storage unit.

The C/PM operating system consists of a cold boot loader, the CCP (Command Control Processor), BDOS (Basic Disk Operating System) and the BIOS, (Basic Input - Output System).

On a floppy disk, the operating system resides on the first two tracks of the disk. The monitor PROM that signs on immediately after a reset contains a short boot routine that reads the first sector into memory at 0000H and jumps to 0000. The cold boot loader, as it is called, then proceeds to read in all of C/PM and put it where it belongs in the system memory. The cold boot loader then jumps to the cold boot entry point in the BIOS and initializes the I/O and completes the sign on.

To prepare for putting the system on the disk, an image of the final operating system must be assembled in RAM. Two separate programs must be prepared.

The first is the C/PM BDOS and CCP. It is prepared by using the MOVCPM utility on your C/PM distribution disk. When you are sure of the size of the system simply type: "MOVCPM <size> *". There will be a short delay and then a message will appear saying "Type SAVE 40 C/PMxx.COM" where xx will be the size.

The other program is a little more difficult in that you must correctly set some "equates" at the beginning to configure it to operate the hardware you are running. In the appendix, you will find some examples of single user, multi user, network and hard disk configurations that should be of help in determining the right way to set up the BIOS. You will notice that the BIOS is now broken into several subparts. The parts are explained as follows:

I/O = Cold boot loader, console input and output, common drive select routines, logical unit map, miscellaneous routines and messages needed by all configurations. I/O must be included in all system configurations.

CXLST = Printer driver, (There are several versions for commonly-used printers such as Centronics, Diablo, etc.)

FLOP = Floppy disk drivers.

BLK = Blocking and De-blocking drivers for use with sector sizes greater than 128.

DATA = Disk parameter tables, skew tables, data storage and disk drive buffers.

XBOOT, CBOOT, or WCBOOT = Cold boot sign on code that gets overwritten by the directory buffer and de-blocking buffer.

The parts of the BIOS are put together using the concatenation feature of PIP. To make the system simply choose the appropriate xxSYS.SUB file that assembles the configuration you need and type "SUBMIT xxSYS". The MACRO assembler is required along with MACRO.LIB and Z80.LIB to construct most modules.

The equates in IO should be modified using a text editor like the one supplied on your distribution disk called ED.COM. ED is difficult to use. We recommend, for assembly language work, that you purchase Polyview by MICRO CONCEPTS in Fullerton, Ca. Word Star by Micro Pro would be an acceptable substitute, but is not as easy to use as Polyview on assembly formats.

The size of the system can be adjusted by changing the MSIZE equate in the BIOS. This number should always be the same as the size you used to create the C/PM BDOS.

Once you have the CP/M image saved on disk and have modified the equates correctly to configure the BIOS to your hardware, run the SUBMIT program on the xxSYS.SUB which links together the SUB modules to create the final BIOS and assemble the whole thing.

At this point, we will assume you have both CPMxx.COM and xxSYS.HEX on the same disk with DSYSGEN.COM. Notice that there is no longer a separate BOOT.HEX file. It is included in the IO section of the BIOS. There is also no need to find the 'BIAS' values to tack onto the BIOS, the DSYSGEN program figures all of that out for you.

When DSYSGEN signs on it will ask if the system you wish to install on the drive exists as a xxxxx.SYS file on the disk already. If you are sure to use the current configuration again, it would be wise to save it after building it with DSYSGEN as a file. Then you need only answer 'Y' to "IS THE SYSTEM ON THE DISK". This saves a few steps. If you are supporting several configurations of mini and 8" drives, various printers, etc., you could keep one diskette with DSYSGEN and all the system files on it. Then putting them on the drives is a snap.

For the moment, we will assume the system has to be built. Answer 'Y' to the query: "DO YOU WANT TO CONSTRUCT A SYSTEM". DSYSGEN will ask for the drive and file name for both C/PM and what is called BOOT + BIOS. DSYSGEN will ask for a destination drive, confirm it, then display a rather large menu.

This menu is a guide for the code bytes stored in the last byte of the first sector of all drives. The appendix explains the meaning of each bit in the code byte, if you are interested. Other details are included for your information. This data was taken from a listing of DSYSGEN, so it is in rather raw form.

Whenever possible, it is advisable to choose the "SKEW" option. Many companies are currently raving about their 'four times faster' C/PM. Optimizing the skew factor is all that is required to multiply disk performance (You should try the double-sided 512 byte 8" configuration if you would like to see a floppy perform like a hard disk).

Using the DFOCO (for Double density FOrmat and COpy) utility, you should have already prepared your disk with the correct format. After answering with the correct code byte, DSYSGEN will write the system onto the disk. If the code byte on the disk and the code byte you have chosen do not agree, the following message will appear: "The new system type and the destination disk format do not match. Do you want to continue (Y or N) ?". This may or may not be ok. DFOCO does not know what you will use for a code byte and has some fixed ideas about what it puts on the disks. Situations such as putting a code byte for an 8" floppy disk on a hard disk drive are incorrect, but choosing a skew code when DFOCO has not put one there (or vice versa) is ok. Presently, skew codes are automatically put on 512 byte sectored disks by DFOCO because we assume you are after the speed and the extra space which that format provides. On other formats such as 128 byte, double density, skew is not assumed because you may have compatability problems with existing disks. In any case, DSYSGEN will overwrite whatever code byte is there if you tell it to.

After writing the system out, DSYSGEN will state that the function is complete and ask if you want to do another disk or hit 'return' and re-boot. Upon being re-booted, it will display a "SAVE" message with a size that has been calculated for the number of bytes in your current system size. If you use this option, you can avoid going through the whole system building process again by simply using the image off disk.

Some attempts are made in the DSYSGEN program to aid you in building a system that will not overwrap FFFF and destroy the boot program that is loading the system. If DSYSGEN detects that your system is too large for the memory size you have chosen, it will tell you so. If the system will not fit on the system tracks of the disk you are using it will notify you of this, as well. In the latter case, DSYSGEN will trim the system to fit (not doing so would hang the disk controller).

DDUMP

This program is an improved DUMP utility for CP/M. Any CP/M file can be DUMPED to the screen in a format similiar to that used by the DDT DUMP command. In addition, any sector or group of sectors may be DUMPED in the same format. DDump should be used for single and double sided disks.

OPERATION

The program may be run either by typing DDUMP or DDUMP followed by the file name or track and sector. Note: The format is quite free. Spaces are usually ignored. They are only required after the words track and sector or T and S, and after the word DDUMP>

If DDUMP is typed the program responds with a heading followed by '*' and waits for more input. Operation in this mode is similiar to other utilities like PIP or DDT.

The operation desired may then be typed in as follows:

```
DDUMP FILE.NAM
DDUMP A:FIL.NAM
DDUMP B:FIL.NAM
```

or DDUMP may be typed separately as:

```
DDUMP
*FILE.NAM
*B:FIL.NAM      (The * is a program prompt)
```

The program may also be used to DUMP disk sectors directly, DUMP any CP/M eight sector group.

FILL

Fill will allow you to fill a certain area of a disk with any character you wish. Within DDUMP *, type FILL. The program will prompt you with Drive to fill?, Track?, Starting sector?, ending sector?, and character to fill.

EDIT

A limited editing feature is included in DDUMP, which allows you to change data on the disk. The edit feature works as follows:

Any single sector on either drive may be edited by requesting display of the sector followed by EDIT.

DUMP 8: TRACK 4 SECTOR 2 EDIT

The requested sector will be displayed followed by an edit prompt

EDIT-

Enter the address of the first BYTE to be changed. The program will respond by typing back the address entered and the present contents of that address. To change the contents of the address enter the new BYTE followed by a carriage return. The program will display the next address and its contents. To stop entering data type a period. The program will redisplay the sector showing the changes made. The edit feature works almost exactly like the S entry in DDT> typing only a carriage return omits entry.

Typing a period merely redisplay the sector from memory; it does not cause it to be written back on the disk. When editing is complete, redisplay the sector by typing a period and type either

WRITE	(writes the sector back on disk)
STOP	(stop editing without writing on disk)

All edit entries must be made in HEX. Entering non HEX characters results in an error message. The permissible address range is 0000 to 007F. Larger addresses give an error message.

The edit feature should be used with caution since it is possible to edit CP/M to "death" by changing a single byte. One occasional valuable use is to restore files that have been accidentally erased. Erasing a file using the ERA command does not erase the data from the disk, but only enters an E5 into the first Byte of the directory. To restore a file, display the directory, such as T 2 S 5. Find the sector containing the name of the file, and change the E5 to 00 and write the sector back onto the disk. This will restore the file provided none of the sectors in the file were changed after the file was erased.

SERVICE POLICY

We would hope that your XOR would never require service, however, we feel that with a thorough reading of the manual most problems will be resolved. Should you feel the need to consult with our Customer Service personnel, they will be happy to answer your questions. Please call the number (714-898-5525) during normal business hours.

If you should require service on your XOR it will be performed at the plant facilities in Huntington Beach, or at an authorized XOR Service Center.

Equipment will be accepted for service only after you have been assigned a "Return Material Authorization Number". RMA. This number may be obtained by calling the Customer Service Hot Line 714-898-5525. The following information will be required:

- The purchase date
- The serial number
- The problem

All shipments received without an RMA will be returned to the sender.

Warranty (6 months parts/labor, from the date of purchase) repairs will be made at no charge for parts and/or labor for XOR hardware only. Merchandise such as drives, terminals, and printers sold by US MICRO SALES as part of their systems will be sent back to the manufacturer for repair. Repair and freight charges for non-XOR merchandise will be billed to you by US MICRO SALES. All warranties are void if any portion of your computer system is altered by anyone other than factory authorized personnel.

Incoming freight MUST BE PREPAID. US MICRO SALES will not accept incoming freight which is marked COLLECT or COD. Shipping charges for the return of repaired equipment to you that is under warranty will be prepaid by US MICRO SALES except in cases where you specify method of shipment other than the one chosen by US MICRO SALES. All return shipping charges for equipment that is no longer under warranty are the responsibility of the party who sent the equipment for repairs.

SUGGESTIONS

If you should have any suggestions regarding the use of this manual, or have any corrections or additions, we would be happy to hear from you. Please address your correspondence to:

U S MICRO SALES
Tech/Man Dept
15392 Assembly Lane
Huntington Beach, CA 92649

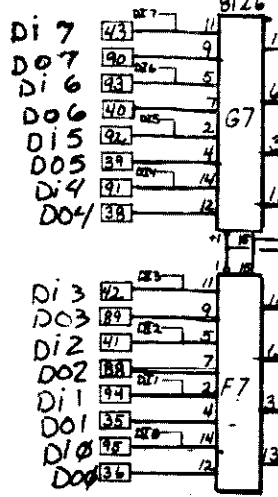
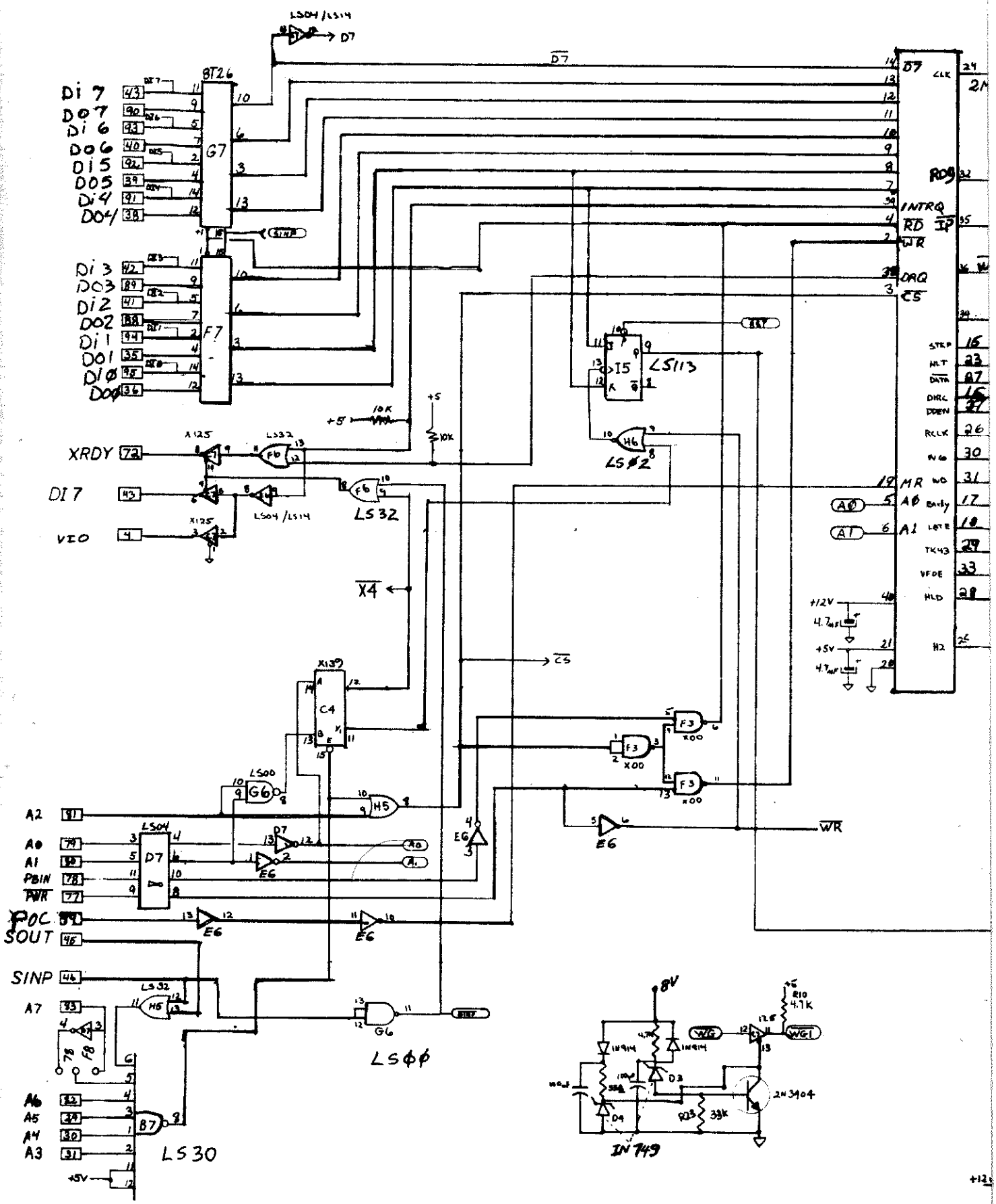
rev 030883

DSK REV D PARTS LIST

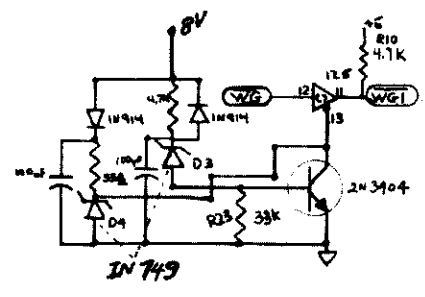
PART NUMBER	DESCRIPTION	QTY.	PART NUMBER	DESCRIPTION	QTY.
I-0000-06	HTSNK-B	3	I-3000-60	100 KRES	1
I-0000-13	NUT 4-40	3	I-3000-67	2.4 KRES	1
I-1000-07	S1 DSK BD	1	I-3000-76	INDUCTOR	1
I-2000-01	8 P SOC	1	I-3000-92	.33 MF 100VR	1
I-2000-05	14 P SOC	18	I-3001-07	10 K POT	2
I-2000-06	16 P SOC	12	I-3001-14	1.5 KRES	1
I-2000-07	18 P SOC	1	I-3001-17	4.7 MF 25VR	3
I-2000-08	20 P SOC	1	I-3001-19	.01 MF 100VR	1
I-2000-11	40 P SOC	1	I-3001-32	22 K CAP	3
I-2000-18	16 PHDR	3	I-3001-45	220/330 14 P	1
I-2000-37	34 P CONN M	1	I-3001-46	100MF 16VRB	2
I-2000-71	50 P CONN M	1	I-7000-02	REG +5	2
I-3000-11	33 RES	1	I-7000-03	REG +12	1
I-3000-15	100 PF 200VR	1	I-7000-08	ZEN +4,3	2
I-3000-27	10 KRES	4	I-7000-12	1N914	2
I-3000-41	220 RES	2	I-7000-27	2N3904	1
I-3000-44	33 KRES	3	I-7000-35	8 MHZXTAL	1
I-3000-45	360 RES	1	I-7400-01	555	1
I-3000-46	39 KRES	1	I-7400-21	7404	1
I-3000-47	4.7 MF 16VR	5	I-7400-25	7416	1
I-3000-50	47 KRES	1	I-7400-41	8T26	2
I-3000-53	560 RES	1	I-7400-71	74LS00	2
I-3000-58	.MF 50VR	10	I-7400-72	74LS02	1
I-3000-59	1 KRES	7	I-7400-74	74LS04	4

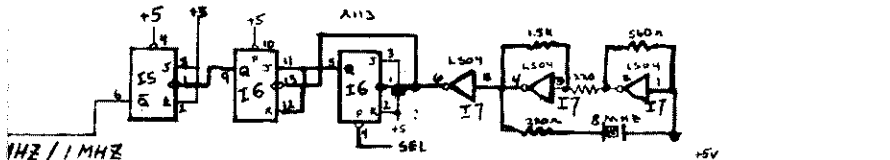
DSK REV D PARTS LIST (CON'T)

PART NUMBER	DESCRIPTION	QTY.	PART NUMBER	DESCRIPTION	QTY.
I-7400-85	74LS32	2	I-7401-26	74LS125	1
I-7400-96	74LS113	3	I-7401-27	74LS124	1
I-7400-97	74LS123	2	I-7401-34	74LS139	1
I-7401-02	74LS157	2	I-7401-35	1691	1
I-7401-07	74LS175	1	I-7401-36	1795	1
I-7401-22	74LS30	1	I-7401-37	2143	1
I-7401-25	74LS393	1			



14	87	CLK	24
13			21
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2			
1			
30	88	RD	32
29			
28			
27			
26			
25			
24			
23			
22			
21			
20			
19			
18			
17			
16			
15			
14			
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2			
1			
31			
30			
29			
28			
27			
26			
25			
24			
23			
22			
21			
20			
19			
18			
17			
16			
15			
14			
13			
12			
11			
10			
9			
8			
7			
6			
5			
4			
3			
2			
1			

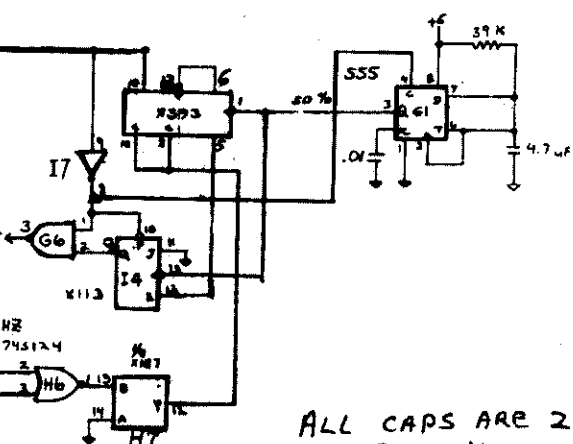
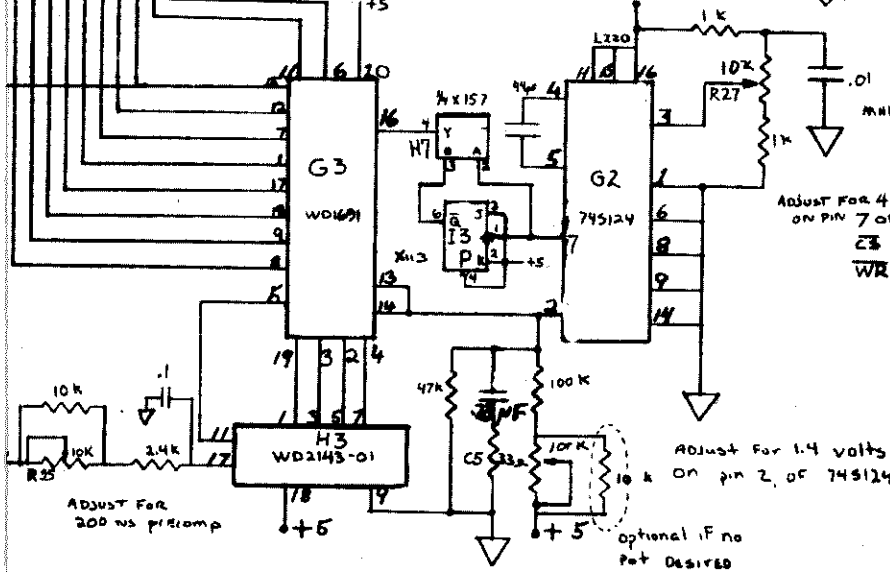
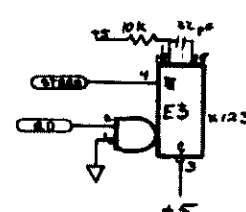
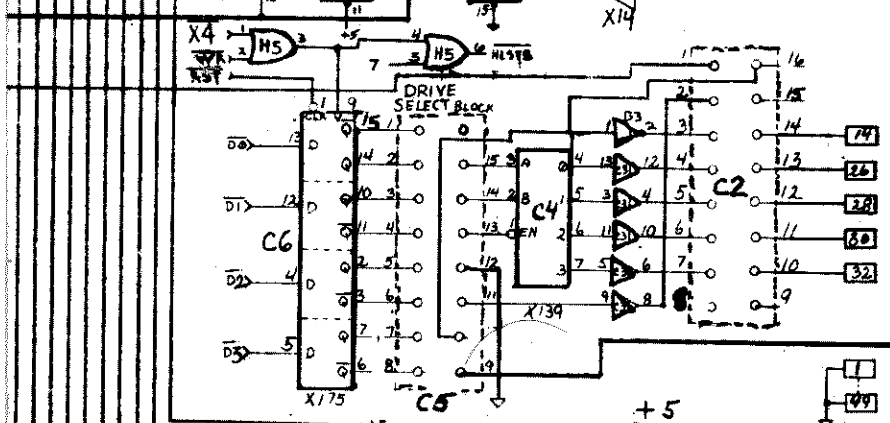
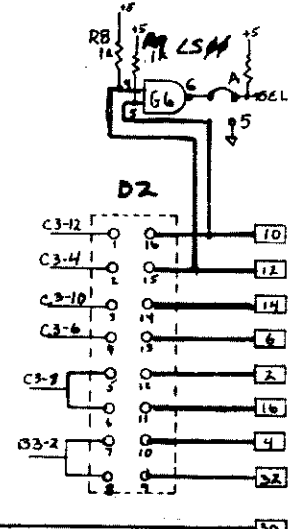
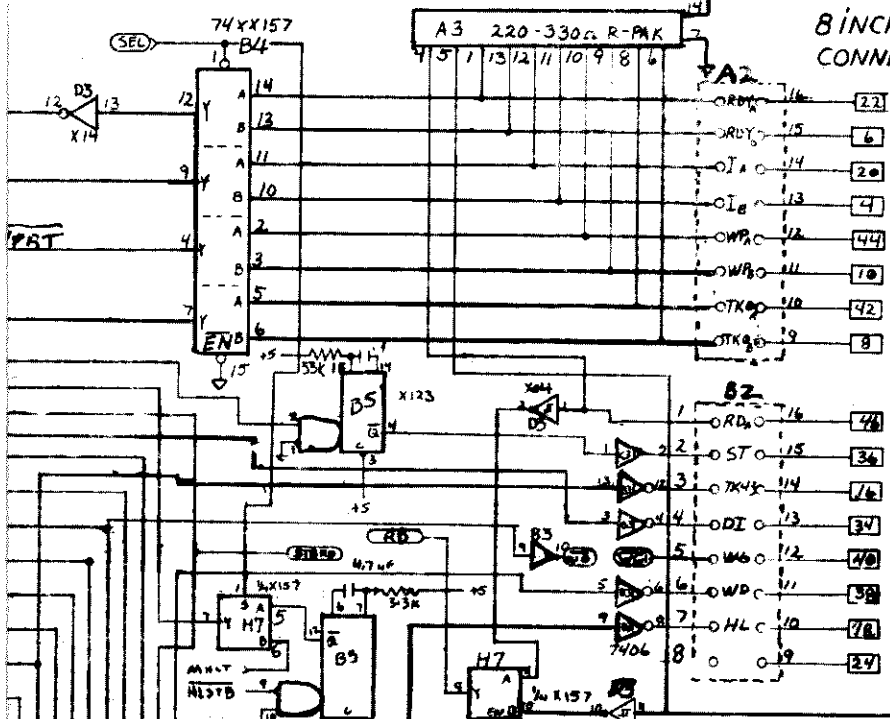




MINI DRIVE CONNECTOR

D2-12	HL	2
D2-10	HZ (SIDE SELECT)	4
D2-13	DS 6	6
B4-10	INDEX	8
C3-10	DS 1	10
C3-6	DS 2	12
D2-14	DS 3	14
D2-11	MOTOR ON	16
B3-4	DIR	18
C3-2	STEP	20
B3-6	WD	22
C7-11	WG	24
A2-8	TK 66	26
A2-6	WP	28
D3-11	RD	30
B3-2	SIDE SELECT	32
		34

8 INCH DRIVE CONNECTOR



ALL CAPS ARE 25V
ALL RES 1/4 W
ALL CAPS AND 25V-UP (1.25 SPACING)
ALL RESISTORS ARE 1/4 WATT (1.25 SPACING)
X = 7443 OR 74

XDS

DATE: 7/16	DESIGNED BY: <i>Ben</i>	DESIGNED BY: T.V.D.
REV D DSK CONTROLLER		
REVISED 11/9/81 REV. D1		

