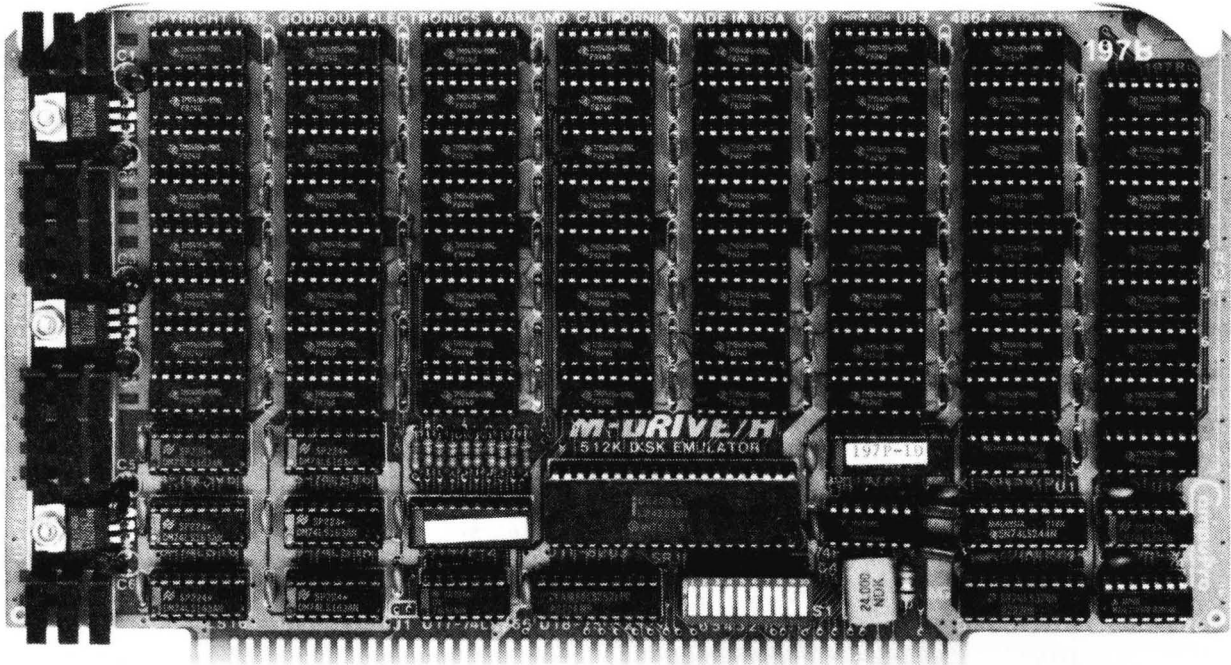


M-DRIVE/H™

TECHNICAL MANUAL



IEEE 696 / S-100

512K SOLID STATE DISK EMULATOR

197B

mac

CompuPro®

A GODBOUT COMPANY

3/83

CONTENTS

How to get you M-Drive/H up and running in under 5 minutes without reading the manual	5
About M-Drive/H	7
Technical overview	7
Configuring M-Drive/H	8
Using M-Drive/H	9
Software	10
Installing M-Drive/H Drivers in other CP/M® systems . . .	10
Sample read/write routines	11-16
Hardware	17
Logic diagrams	17-20
Parts list	21
Component layout	21
Service and limited warranty information	24

M-DRIVE/H TECHNICAL MANUAL
Copyright ©1982
CompuPro a Godbout Company
Oakland Airport, CA 94614

First Printing: December 1982
Second Printing: March 1983

Contents of this document were proofread with the aid of SpellGuard~ a product from SORCIM Inc. Santa Clara, CA.

Document #11978

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

Registered trademarks: CompuPro; CompuPro a Godbout Company: CP/M; Digital Research. Trademarks: M-DRIVE/H; W.J. Godbout: CP/M 80; CP/M-86; MP/M-86 and Digital Research; Digital Research.

All rights reserved. No part of this publication may be reproduced or transmitted in any form, or by any means, without the written permission of CompuPro. Printed and assembled in the United States of America.

How to get your M-DRIVE/H up and running in 5 minutes without reading the manual

This section is for those of you that can't wait long enough to read the rest of this manual to get your M-Drive/H board(s) working. This section assumes that you have a standard CompuPro system, and that the revision levels of your various operating systems are equal to or greater than those shown below:

CP/M 80™ : CP/M 2.2MDB or higher.
CP/M 86™ : CP/M 86 1.1PAA or higher.
MP/M 816™: MP/M 816 2.1D or higher.

If your CompuPro operating system revision level is lower than the above, return your original master diskette to CompuPro and we will upgrade it for you. There is no charge for this service. Make sure your diskette is packaged well and return it to:

CompuPro
ATTN: M-Drive/H Upgrade
Bldg. 727 PO Box 2355
Oakland Airport, CA 94614

If your disk system was not supplied by CompuPro, but uses a standard version of CP/M 2.2, or if you have an older version of the CompuPro operating system and can't wait for the mail, set the switches as shown below and then proceed to the section of this manual entitled "Installing M-Drive Drivers for other CP/M Systems".

SWITCH SETTINGS

There is only one dip-switch on the M-Drive/H board. Positions 1-7 are used to select the two port addresses where the M-Drive/H board resides. Positions 8-10 select the cascade address for each board in the system. Up to eight M-Drive/H boards can exist at the same port addresses, only the cascade address needs to be different for each board.

The standard port addresses for the M-Drive/H board are C6 and C7 hex. To set the board for the standard port address, set positions 1-7 of dip-switch S1 as shown below:

on			x	x	x		
S1*	1	2	3	4	5	6	7
off	x	x				x	x

* Markings on switch, not legend.

To set the cascade address, determine how many M-Drive/H boards there are in the system, and set each of them according to the table following:

1st board	2nd board	3rd board	4th board
on x x x	on x x	on x x	on x
S1* 8 9 10	S1* 8 9 10	S1* 8 9 10	S1* 8 9 10
off	off x	off x	off x x

5th board	6th board	7th board	8th board
on x x	on x	on x	on
S1* 8 9 10	S1* 8 9 10	S1* 8 9 10	S1* 8 9 10
off x	off x x	off x x	off x x x

*Markings on switch, not legend.

These are the only switch settings that are required on the M-Drive/H board.

Plug the M-Drive/H board into the system, making sure that power is off before doing so. Turn the system on and the left-hand drive light should be flashing (as it normally does). Put a system diskette in Drive A and close the door. The system should sign on.

The software should automatically format the M-Drive/H board(s) for use and print a message telling you how much M-Drive/H storage it thinks is available (in K bytes). Check this number to see if it seems correct, depending on the number of boards in the system (each M-Drive/H board contains 512K bytes of storage). If the size seems incorrect, power down the system and check your switch settings. If the size is correct, all is well.

The M-Drive/H appears in your system as drive M:. Once the system is booted, you may treat it just as if it were a disk drive. You can PIP files to it, log onto it, and use STAT to find out how big it is, or how much space is remaining. Remember that all data will be lost in the M-Drive/H when power goes away, so be sure to transfer important work back to permanent media before turning off power. It's a good idea to periodically perform this function anyway, to protect against a power failure.

M-DRIVE/H TECHNICAL MANUAL

ABOUT THE M-DRIVE/H

CompuPro, in a joint effort with G&G Engineering, pioneered the M-Drive concept for microcomputers. The M-Drive concept is to use RAM memory to emulate a disk drive. Thus, the convenience and familiarity of a disk drive was preserved, but the drive now operates at RAM speeds, which is considerably faster than a mechanical disk drive. The original M-Drive was implemented as a software product which used extended address memory beyond the first 64K under CP/M 80. The CompuPro dual processor board (CPU 8085/88) made this possible by actually using the 8088 to manage the M-Drive memory.

This system worked quite well, but had a few limitations. First of all, it was an all right technique for CP/M 80, because all that extra memory was never used. But when running CP/M 86 or MP/M 816, the operating system wants to use that memory for running programs. Secondly, although the extra memory was certainly cost-effective for some applications, it was still quite expensive to get a reasonably sized M-Drive.

Right from the start, we knew that a "hardware" M-Drive was the answer. The result is M-Drive/H (if you haven't figured it out, the /H stands for Hardware). The M-Drive/H board system comprises one to eight boards, providing between 512k bytes and 4 Megabytes of storage.

Now the M-Drive concept is extended easily to all operating systems, without taking up any precious memory space. All of CompuPro's standard operating systems, including CP/M 86 and MP/M 816, contain built in support for M-Drive/H. The 512K of M-Drive/H storage is certainly much less expensive than 512K of regular system memory, but of course the M-Drive/H RAM is dedicated to the M-Drive/H and cannot be used as system memory.

We feel that the M-Drive/H board is the best of its type for IEEE 696/S-100 systems, and we thank you for choosing it.

TECHNICAL OVERVIEW

The M-Drive/H board consists of 512k bytes of dynamic RAM, a dynamic RAM controller circuit and a method for addressing the data in the RAM array.

The M-Drive/H looks like two I/O ports to the system bus, it takes up no memory space. One I/O port is used to load a starting address into the board, and the other is used to read and write data.

The starting address is actually loaded into a series of counters on the board, and bytes are then transferred in sequence without the need to send a new address for each byte. The counters automatically increment themselves to point to the next byte. This speeds up transfers considerably.

The counter has 22 bits, 3 more than are needed for 512K bytes. These extra three bits determine which board of eight should send or receive the data. Thus, the array seems to the software like a contiguous 4 megabytes,

instead of eight separate chunks of 512k bytes each. This makes the programming task quite a bit easier, as well as using a minimum of port addresses.

The dynamic RAM controller circuit consists mainly of an Intel 8203-3 dynamic RAM controller and some external logic to permit operation with 512K of RAM. This circuit handles most of the DRAM access functions, as well as providing the necessary refresh operation and arbitration.

Since the DRAM is isolated from the IEEE 696/S-100 bus, problems normally associated with DRAM and the S-100 bus are avoided.

CONFIGURING THE M-DRIVE/H

There is only one dip-switch on the M-Drive/H board. Positions 1-7 are used to select the two port addresses where the M-Drive/H board resides. Positions 8-10 select the cascade address for each board in the system. Up to eight M-Drive/H boards can exist at the same port addresses, only the cascade address needs to be different for each board.

The standard port addresses for the M-Drive/H board are C6 and C7 hex. To set the board for the standard port address, set positions 1-7 of dip-switch S1 as shown below:

	on			x	x	x		
S1*	1	2	3	4	5	6	7	
	off	x	x			x	x	

* Markings on switch, not legend.

If you desire to use a different set of port addresses, the following table shows how positions 1-7 of S1 relate to the various address bits:

Paddle #	Address Bit	
1	7	
2	6	
3	5	<u>OFF</u> = "1"
4	4	
5	3	<u>ON</u> = "0"
6	2	
7	1	

To set the cascade address, determine how many M-Drive/H boards there are in the system, and set each of them according to the table following:

1st board	2nd board	3rd board	4th board
on x x x	on x x	on x x	on x
S1* 8 9 10	S1* 8 9 10	S1* 8 9 10	S1* 8 9 10
off	off x	off x	off x x

5th board				6th board				7th board				8th board			
on		x	x	on		x		on			x	on			
Sl*	8	9	10	Sl*	8	9	10	Sl*	8	9	10	Sl*	8	9	10
off	x			off	x		x	off	x	x		off	x	x	x

*Markings on switch, not legend.

These are the only switch settings that are required on the M-Drive/H board.

USING M-DRIVE/H

After making sure that power to the system is OFF, plug the M-Drive/H board(s) into the system, of course making sure that their switches are set accordingly. Boot the system in the normal manner. The system should recognize the presence of the M-Drive/H and display an appropriate sign-on message. It should display the size of the M-Drive/H in K bytes (each M-Drive/H board contains 512K bytes). If the size seems incorrect, power down the system and check your switch settings.

At power-up, the system will automatically format the M-Drive/H for use. You may "force" a re-format of the M-Drive by invoking the MFORM command. Type: **MFORM M** followed by a carriage return. The program will ask you if you indeed wish to re-format your M-Drive/H. If you do, it will re-format the M-Drive/H, but it will also wipe out all your files, so be careful.

Once the M-Drive/H is formatted (either automatically or "by hand") it is ready for use. Simply treat it as you would any other disk drive in your system. The M-Drive/H becomes drive M:, and you may PIP files to it, run programs from it, log onto it, in short anything you can do with a normal disk drive. However, operation will be much faster.

Remember that the data you store in the M-Drive/H will be lost when you turn off the power to your computer, so be sure and save any files you have worked on back to permanent storage media, such as a hard or floppy disk. It's a good idea to perform this "back-up" procedure often (even when you're not done for the day) to guard against an accidental power failure.

SOFTWARE

INSTALLING M-DRIVE/H DRIVERS IN OTHER CP/M SYSTEMS

To install the M-DRIVE/H drivers in other CP/M systems, copy the files "MDRIVE.COM" and "MFORM.COM" onto your system disk. You must now make some room for the new drivers above your current CP/M. Currently, the M-DRIVE/H drivers and buffers take about 1K (1024) bytes. This means that you must make a CP/M 1K less than the amount of memory available in your system. If you do not know how to move your CP/M down, see your system manual. The required program is usually called "MOVCPM".

Once you have 1K of room at the top of memory, each time you boot your system, you must call the MDRIVE program. The format of the MDRIVE command is as follows:

```
MDRIVE  -DRIVE DESIGNATOR  ADDRESS
```

Where DRIVE DESIGNATOR is the letter of the logical drive you want the memory drive to respond to (A-P), and ADDRESS is the address in hex of the end of your CP/M, or the TOP of your memory minus 1K. See table 1 to determine what hex address you should use for different memory sizes.

TABLE 1

<u>Amount of memory</u>	<u>Size CP/M to make</u>	<u>Hex address to use</u>
64K	63K	FC00
60K	59K	EC00
56K	55K	DC00
52K	51K	CC00
48K	47K	BC00
44K	43K	AC00
40K	39K	9C00
36K	35K	8C00
32K	31K	7C00
28K	27K	6C00
24K	23K	5C00

The following example will put the M-DRIVE/H drivers at the top of a 63K system with 64k of memory and put the memory drive as drive "M":

```
A>>MDRIVE -M FC00
```

The MDRIVE program will first size the memory drive and report the amount of memory drive it found. It will then patch your BIOS, and install the new drivers at the given address. The memory drive is then automatically formatted. You can now use the memory drive like any other drive in the system, including reformatting it.

We have also included sample M-DRIVE/H drivers that can be assembled into an existing bios. This code was taken out of the standard CompuPro CPM 2.2

BIOS, so some modification and optimization may need to be done for individual CP/M implementations. The disk parameter blocks (DPB) will also need to be set for the number of M-DRIVE/H boards in your system.

SAMPLE READ/WRITE ROUTINES

```

;
;   M - D R I V E / H
;
;   This is a sample, skeletal bios that will drive the CompuPro
;   M-DRIVE/H memory disk.
;   To quickly add this to your existing bios, just replace the "JMP
;   NOTHING"s at the end of each routine with a jump to your BIOS,
;   and assemble it all together. The DPB is set for one M-DRIVE/H
;   board, if you have more boards you will have to adjust it
;   accordingly.
;   Information such as sector, track, and DMA address are saved even
;   if they are not intended for the memory drive. This is intended
;   as an example and as a way for someone to use the M-DRIVE/H board
;   with a minimum amount of trouble and work. The performance of
;   any CP/M BIOS can be greatly enhanced by integrating this code
;   directly into an existing BIOS, as well as save some memory space.
;
;   This code expects the M-DRIVE/H board to be addressed at the
;   following I/O ports:
;
;               OC6H - Data
;               OC7H - Address
;
;   Written L. Ott                               12/17/82
;   ^1982 by CompuPro div. Godbout Electronics
;
;
HBASE: EQU      0C6H
HDATA: EQU      HBASE
HADDR: EQU      HBASE+1

MEMDRV: EQU     'M'           ;SET AS DRIVE "M"
F$RDAT: EQU     6             ;READ DATA
F$WRT: EQU      5             ;WRITE DATA

MRTRY: EQU      10           ;MAX RETRY COUNT
;
MDRIVE: JMP     NOTHING      ;COLD START
        JMP     NOTHING      ;WARM START
        JMP     NOTHING      ;CONSTAT
        JMP     NOTHING      ;CONIN
        JMP     NOTHING      ;CONOUT
        JMP     NOTHING      ;LIST
        JMP     NOTHING      ;PUNCH
        JMP     NOTHING      ;READER
        JMP     HOME         ;HOME
        JMP     SELDSK       ;SELECT DISK
        JMP     SETTRK       ;SET TRACK

```

```

        JMP     SETSEC           ;SET SECTOR
        JMP     SETDMA          ;SET DMA
        JMP     READ            ;READ
        JMP     WRITE           ;WRITE
        JMP     NOTHING         ;LIST STATUS
        JMP     SECTRN          ;SECTOR TRANSLATE
;
;DUMMY ROUTINE FOR NON IMPLEMENTED ROUTINES
;
NOTHING:
        RET
;
HMDSIB: EQU     2048/128        ;SECTORS IN MEMORY DRIVE DATA BLOCK
HMDDSM: EQU     ((512-4)*8)/HMDSIB
;
DPBHMD:                                ;SET FOR 1 BOARD, 8 SPT
        DB     128              ;MDRIVE TYPE FOR MFORM, DO NOT CHANGE
        DW     8                ;SECTORS PER TRACK, THIS AND DSM ...
                                ;...MUST BE INCREASED FOR MORE BOARDS
        DB     4,15,1          ;BSH, BLM, EXM
        DW     HMDDSM-1,128-1  ;DSM , NUMBER OF DIRECTORY ENTRIES
        DB     11000000B,00000000B ;RESERVED DIR BLOCKS, MUST MATCH ...
                                ; ... NUMBER OF DIRECTORY ENTRIES
        DW     0                ;CHECK SUM
        DW     4                ;NUMBER OF RESERVED TRACKS FOR PARITY CHECK
                                ; THIS IS ALWAYS 4
;
HMDDPH:
        DW     0                ;SECTOR TRANSLATE TABLE
        DW     0,0,0           ;SCRATCH
        DW     BUFDIR          ;DIRECTORY BUFFER
        DW     DPBHMD+1        ;DPB BLOCK
        DW     0                ;CHECKSUM VECTOR
        DW     ALV             ;ALLOCATION VECTOR
;
;
;
        S E L E C T   D I S K   D R I V E
;
;
        Select the disk drive for subsequent disk transfers and return
;
        the address of the DPH for the memory drive.
;
;
        ENTRY -- C = disk selection value
;
        DE and l = 0 must determine disk type
;
        = 1 type already determined (only needed for floppies)
;
;
        EXIT -- HL = DPH address of the memory drive.
;
        IF memory drive not selected, control is passed to BIOS
;
;
SELDSK: XRA     A                ;CLEAR A
        STA     SELFLG          ;CLEAR MEMORY DRIVE FLAG
        MOV     A,C              ;
        CPI     MEMDRV-'A'      ;IS MEMORY DRIVE SELECTED
BIOSSEL: JNZ    NOTHING         ;F NOT GO BACK TO BIOS
        MVI     A,OFFH          ;
        STA     SELFLG          ;SET MEMORY DRIVE AS SELECTED

```

```

LXI      H, HMDDPH      ;GET DPH ADDRESS
RET      ;AND DONE
;
;
;   H O M E
;   Set current track to 0
;
HOME:    XRA      A
        STA      CIOPB+2      ;SET TRACK AS 0
        STA      CIOPB+3
BIOSHOM: JMP     NOTHING      ;ON TO BIOS
;
;
;   S E T      T R A C K
;
;   Set track number.  The track number is save for later use during
;   a disk transfer.  The track is save whether or not the memory disk
;   is currently selected.
;
;   ENTRY -- BC = Track number.
;
SETTRK:  MOV     L,C
        MOV     H,B
        SHLD   CIOPB+2
BIOSTRK: JMP     NOTHING
;
;
;   S E T      S E C T O R
;
;   Set sector number.  The sector number is save for later use during
;   a disk transfer.  The sector is saved whether or not the memory
;   disk is currently selected.
;
;   ENTRY -- BC = Sector number.
;
SETSEC:  MOV     L,C
        MOV     H,B
        INX    H              ;+1 TO MAKE LIKE FLOPPIES (1-26)
        SHLD   CIOPB+4
BIOSSEC: JMP     NOTHING
;
;
;   S E T      D M A   A D D R E S S
;
;   Set the address for the next DMA transfer.
;   The DMA address is saved for later use during a disk transfer.
;   The address is saved whether or not the memory disk is
;   currently selected.
;
;   ENTRY -- BC = Disk memory address.
;
SETDMA:  MOV     L,C
        MOV     H,B
        SHLD   DMAADR
BIOSDMA: JMP     NOTHING
;
;
;   S E C T O R   T R A N S L A T E
;
;   Translate the sector number from logical to physical.

```

```

;
; ENTRY DE = 0, no translation required.
; DE = Translation table address.
; BC = Number to translate.
SECTRN:
MOV A,D
ORA E
RZ ;No translation needed.
JMP NOTHING ;Let BIOS translate.
;
; R E A D
;
; Read the currently selected track,sector from the currently
; selected drive.
;
READ: LDA SELFLG
ORA A
BIOSRED:JZ NOTHING
MVI A,F$RDAT ;MUST WANT THE MEMORY DRIVE
STA CIOPB ;TO COMMAND BUFFER
JMP HMDFNL
;
; W R I T E
;
; Write to the currently selected drive at the current track
; and sector.
;
WRITE: LDA SELFLG
ORA A ;IS MEMORY DRIVE SELECTED
BIOSWRT:JZ NOTHING
MVI A,F$WRT ;SET COMMAND AS WRITE
STA CIOPB
; JMP HMDFNL ;AND GO PROCESS
;
; HMDFNL -- Hard memory disk final command processing
; ENTRY -- CIOPB +0 = COMMAND
; CIOPB +2 = TRACK
; CIOPB +4 = SECTOR
HMDFNL:
MVI A,MRTRY ;Set retry count
STA RTRY
HMFNL1: LHL CIOPB+2 ;GET TRACK
DAD H ;*2
DAD H ;*4
DAD H ;*8
DAD H ;*16
DAD H ;*32
DAD H ;*64
DAD H ;*128
LDA CIOPB+4 ;GET SECTOR
DCR A ;STARTS AT 0
CALL SETMDRV ;SET HDRIIVE BOARD AHL
MVI C,128 ;128 BYTE SECTORS
MOV B,C ; also into B for parity check
LHL DMAADR ;DIRECTLY TO CPM BUFFER

```

```

LDA    CIOPB          ;GET COMMAND
CPI    F$RDAT        ;
JNZ    HDINTO        ;IF WRITE COMMAND JUMP
;
;
;    JMP    HDOTOF
;
;    GET A 128 BYTE BLOCK FROM HDRIVE BOARD ADDRESS IS ALREADY SELECTED
;    C IS NUMBER OF BYTES TO MOVE
;    HL IS DMA BUFFER TO PUT DATA
HDOTOF: IN    HDATA          ;GET DATA BYTE
MOV    M,A           ;PUT TO DMA BLOCK
ADD    B             ;ADD UP PARITY BYTE
MOV    B,A          ; AND SAVE
INX    H             ;UP POINTER
DCR    C             ;COUNT THIS MOVE
JNZ    HDOTOF       ;AND DO AGAIN
CALL   PARITY        ;DO PARITY CHECK
IN     HDATA         ;GET PARITY BYTE
SUB    B             ; 0 IF NO ERROR
RZ     ; THEN RETURN
LDA    RTRY          ;GET RETRY COUNT
DCR    A             ;
STA    RTRY          ;REPLACE WITH NEW COUNT
JNZ    HMFNL1
ORI    1
RET                                ;RETURN WHEN DONE
;
;    PUT A BLOCK OF DATA INTO HDRIVE BOARD ADDRESS IS ALREADY SELECTED
;    C IS NUMBER OF BYTES TO MOVE
;    HL IS DMA BUFFER TO GET DATA FROM
HDINTO: MOV    A,M           ;GET BYTE FROM BUFFER
OUT    HDATA         ;TO MDRIVE BOARD
ADD    B             ;ADD UP PARITY BYTE
MOV    B,A          ; AND SAVE
INX    H             ;
DCR    C             ;COUNT THIS MOVE
JNZ    HDINTO       ;IF DONE WITH 128 OR C COUNT
CALL   PARITY        ;SELECT PARITY BYTE LOCATION
MOV    A,B
OUT    HDATA         ;WRITE PARITY BYTE TO MEMORY DISK
XRA    A             ;AND SHOW NO ERROR
RET                                ;RETURN
;
;    FIND LOCATION IF PARITY BYTE FOR TRACK IN CIOPB+2
;    AND THE SECTOR IN CIOPB +4
PARITY: LDA    CIOPB+4    ;Get sector
DCR    A             ;Bias
LHLD   CIOPB+2        ;Get track

;Set M-DRIVE/H board to address in A,HL
SETMDRV:
OUT    HADDR         ;To M-DRIVE/H board
MOV    A,H           ;H is always 0 or 1
OUT    HADDR

```

```

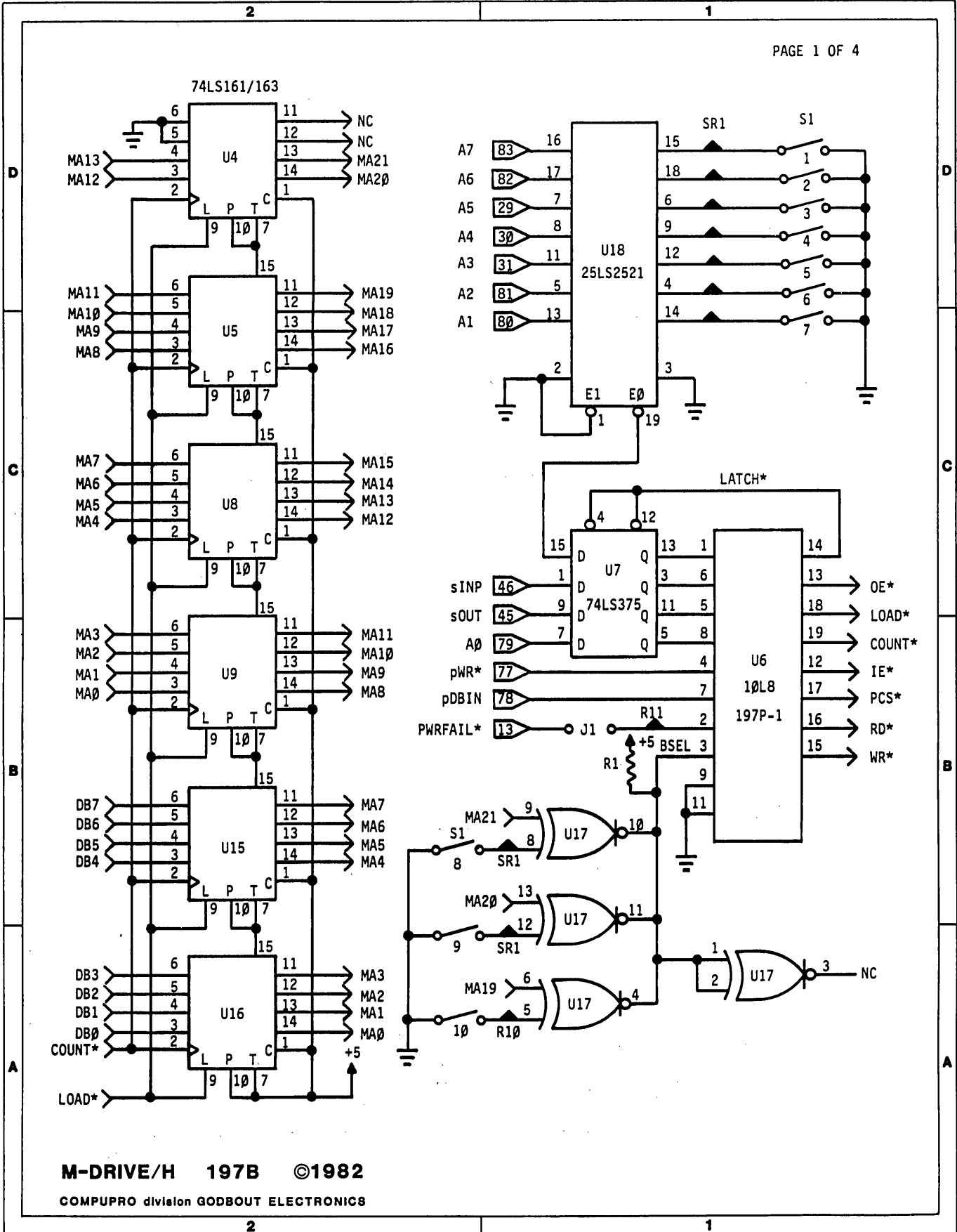
MOV     A,L           ;Upper bit of L picks track 0-3
OUT     HADDR        ; remainder of byte picks sector and byte
RET

;
MDRLNG: EQU    $-MDRIVE      ;LENGTH OF MDRIVE BIOS
SELFLG: DB     1             ;DISK SELECT FLAG
RTRY:   DB     1             ;RETRY COUNTER
CIOPB:  DS     8             ;COMMAND BUFFER
DMAADR:  DW     1             ;ADDRESS OF DMA BUFFER
BUFDIR: DS     128           ;DIRECTORY BUFFER FOR CPM
ALV:    DS     (((HMDDSM*8)+7)/8)+1      ;
;

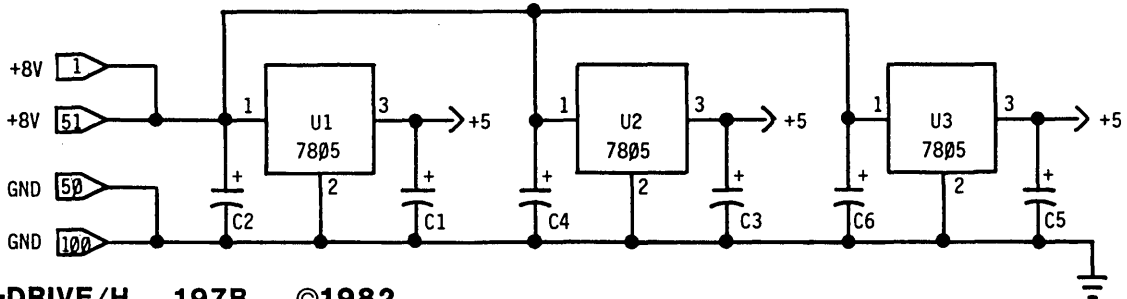
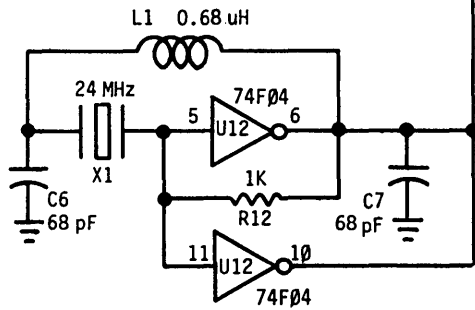
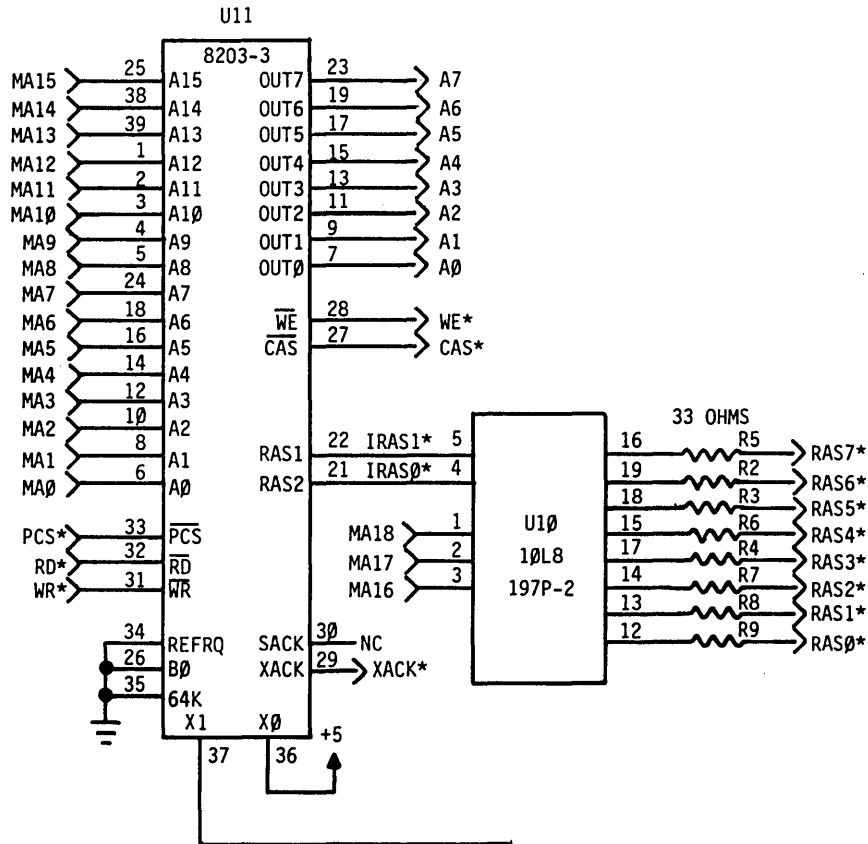
```

HARDWARE

PAGE 1 OF 4

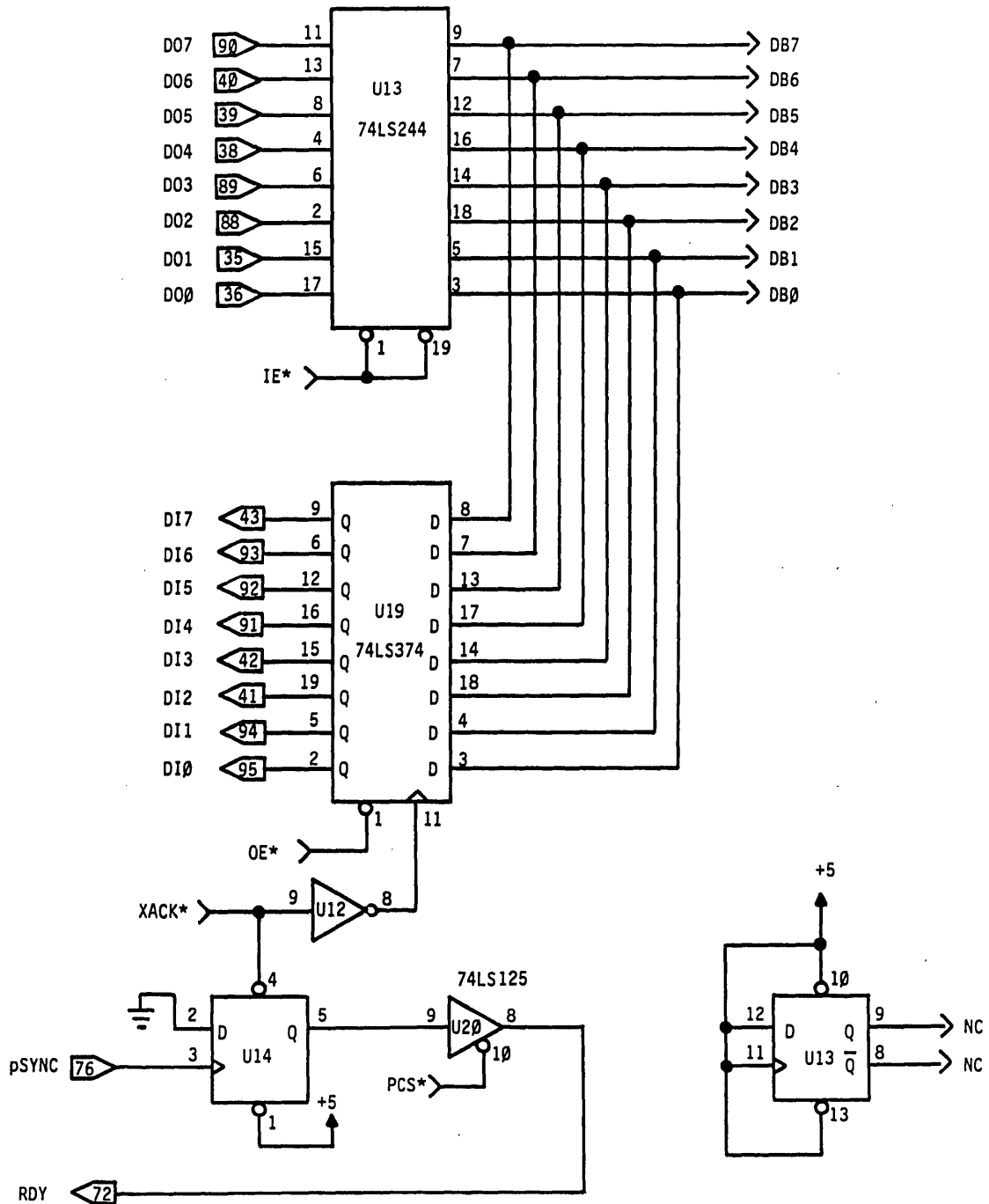


M-DRIVE/H 197B ©1982
 COMPUPRO division GODBOUT ELECTRONICS

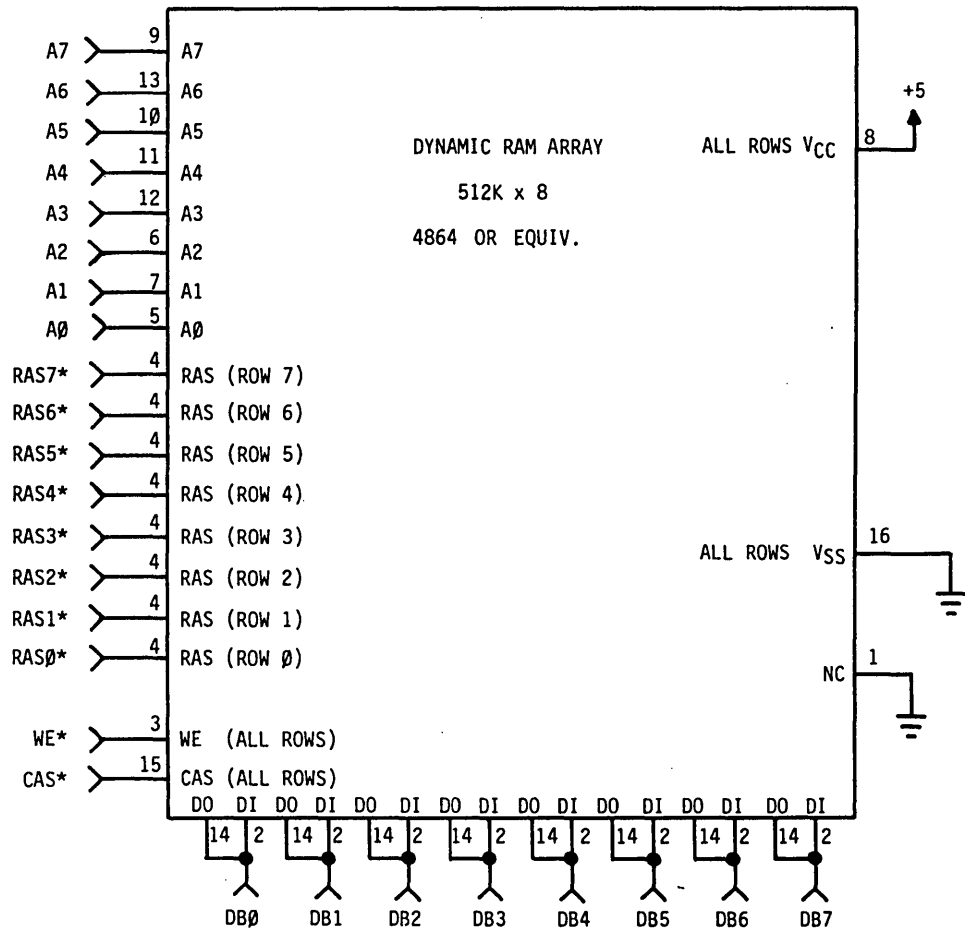


M-DRIVE/H 197B ©1982

COMPUPRO division GODBOUT ELECTRONICS



M-DRIVE/H 197B ©1982
 COMPUPRO division GODBOUT ELECTRONICS



M-DRIVE/H 197B ©1982

COMPUPRO division GODBOUT ELECTRONICS

PARTS LIST

SEMICONDUCTORS

U1	7805	U11	8203-3
U2	7805	U12	74F04
U3	7805	U13	74LS244
U4	74LS161/163	U14	74LS74
U5	74LS161/163	U15	74LS161/163
U6	197P-1	U16	74LS161/163
U7	74LS375	U17	74LS266
U8	74LS161/163	U18	25LS2521
U9	74LS161/163	U19	74LS374
U10	197P-2	U20	74LS125
		U21-84	4864 or equiv.

RESISTORS VALUE

R1	1.5K ohm
R2-9	33 ohm
R10,11	4.7K ohm
R12	1K ohm
SR1	5.1K ohm SIP

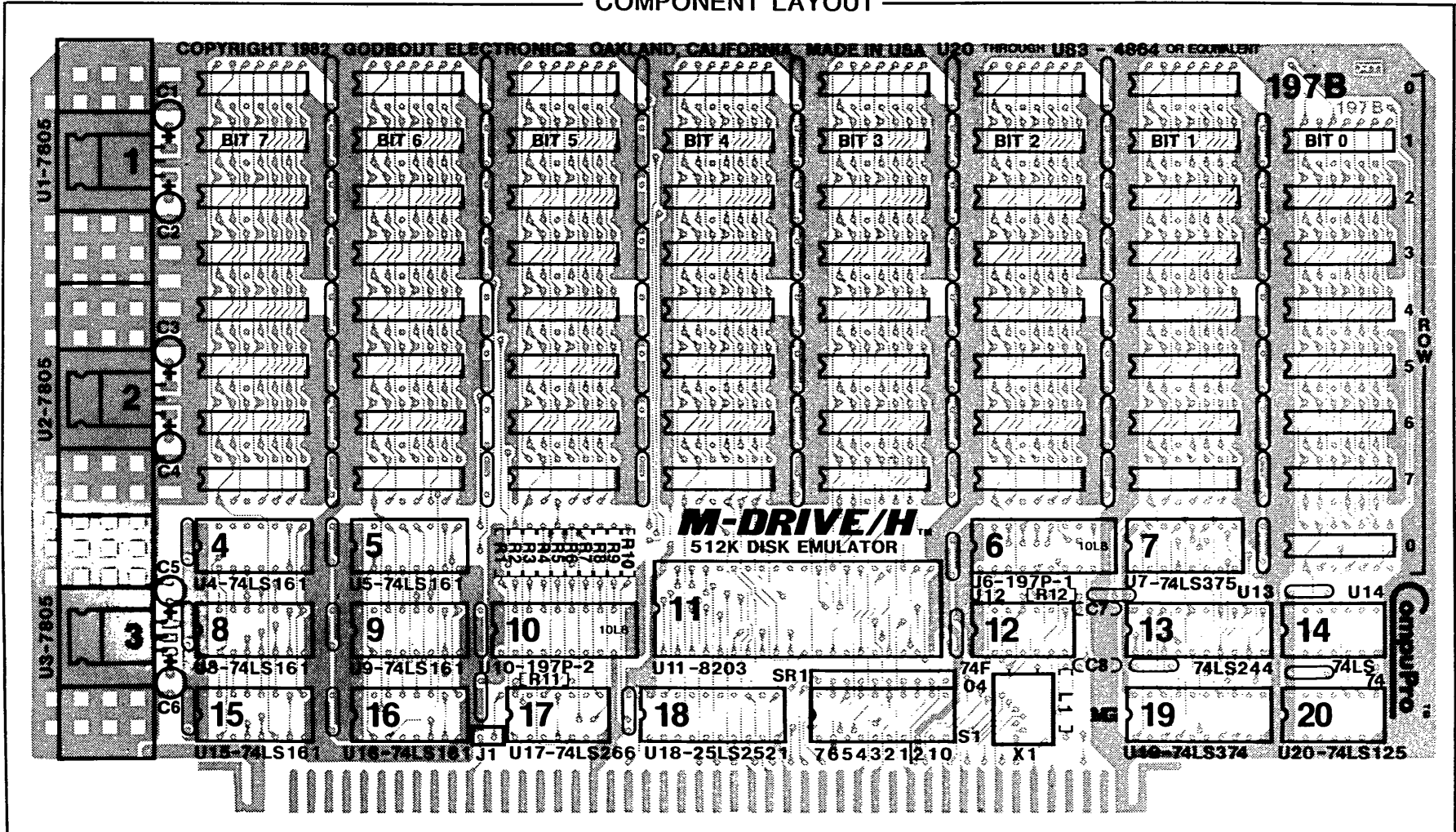
OTHER ELECTRONIC PARTS

L1	.68uH inductor
X1	24 MHz crystal

CAPACITORS VALUE

C1-6	10v or higher
C7,8	68 pF

COMPONENT LAYOUT



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

CUSTOMER SERVICE INFORMATION

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

Box 2355, Oakland Airport, CA 94614

LIMITED WARRANTY INFORMATION

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

If a defective part causes a CompuPro product to operate improperly during the 1 year warranty period, we will service it free (original owner only) if delivered and shipped at owner's expense to and from our Service Center in Building 725, Oakland Airport, CA 94614. If improper operation is due to error or errors on the part of the purchaser, there may be a repair charge. Purchaser will be notified if this charge exceeds \$50.00. If the warranty period has expired, service for CompuPro products is available at a rate of \$50.00 per hour labor plus parts.

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

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

Prices and specifications are subject to change without notice.