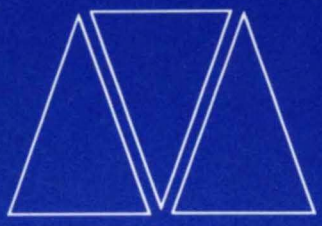


MINI COMPUTER TECHNOLOGY

MODEL

SMC 903 REFERENCE MANUAL

SMC903

REFERENCE MANUAL

Revision A5 January 15, 1982

INTRODUCTION

The SMC903 Storage Module controller provides a low-cost high-reliablity interface for the Interdata line of minicomputers to the CDC Storage Module series of high-performance disk drives and to other drives with compatible interfaces, including drives made by Ampex, Memorex, ISS, and CalComp. Cables are provided for the flat-cable interface only, although operation with non-flat cable drives is possible with a customer-supplied cable. This manual contains the information necessary for a programmer to use the SMC903 properly. In addition, it provides a complete description of the hardware characteristics of the controller.

LIMITED WARRANTY

The SMC903 disk controller is covered by a one-year factory warranty of parts and labor when repairs are performed at MiniComputer Technology, Palo Alto, California. Customer pays freight to MiniComputer Technology; return surface freight is paid by MiniComputer Technology. Cables are warranted for 90 days. This warranty is limited to replacement or repair of defective controllers or cables.

SERVICE

Controllers not under warranty, or damaged or altered, will be repaired or replaced at the option of MiniComputer Technology for a flat charge of $250. Controllers so damaged or altered as to by judged by MiniComputer Technology irreparable will be returned or, at the customer's option, replaced with a new controller board at the current lowest published OEM price.

CONTENTS

BOARD INSTALLATION AND CABLING

The SMC903 controller may be installed in any slot. It must be below its selector channel card if a selector channel is being used; if a selector channel is installed but is not used by the SMC903, the SMC903 may be installed below the selector channel ONLY if the two devices (SMC903 and SELCH) will not be used simultaneously. The SMC903 uses RACK0/TACK0 on the connector 1 side of the backplane only. RACK0/TACK0 on connector 0 (left side as viewed from the back) must be jumpered on the backplane.

The multiplex cable (wide 60-conductor) runs from SMC903 J2 to drive 0. More multiplex cables run from drive 0 to drive 1 if it is present. The last drive in the chain must have a terminator installed.

There is a separate radial cable for each drive, running from SMC903 J0-J1 to drives 0 and 1. The drive should be AC grounded to the CPU chassis as specified in the drive manual. Improper grounding can cause erratic operation, and can be caused by too many ground connections as well as too few.

SMC903 JUMPER OPTIONS

DEVICE ADDRESS (DA): Connect the selection lines 2-200 to either VCC or GND. Lines connected to VCC are added together to form the address, which is always even. The default address is X'70' and the default etch connections must be cut before making new connections. The preferred method of jumpering is to install wirewrap pins and wirewrap the new connections.

EXTENDED DEVICE ADDRESS (XDA): On 32-bit machines with 9- or 10-bit addresses, the connection XDA (normally left open) is made. NOTE: XDA may not be made on 16-bit machines, and when XDA is not made, device address bits 100 and 200 must be connected to GND.

DRIVE JUMPERING: Various jumper options in the drive must be properly selected. The exact options will vary with the particular drive, but in general, the sector size must be jumpered, and the NRZ data option must be selected.

Compute sector jumpering by first determining the total number of bytes per track. This will be 13440 or 20160. In the following description, the values for a track size of 20160 appear in parentheses. Add 60 (80) to the number of data bytes in the sector. The 60 (80) bytes include mechanical tolerances at the beginning and end of sector, VFO lock time, six bytes of ID & ID checksum and two bytes of data checksum. Divide 13440 (20160) by the sum and take the integer portion of the result. This is the number of sectors available per track. Then divide 13440 (20160) by the number of sectors per track and again take the integer portion of the result. Note: on some drives with 20160 bytes per track, the dividend will still be 13440. This applies to the CalComp T82 and T302, for example. The formatter/diagnostic will provide an approximate check on the jumpering during the sector counter test.

NEGATIVE VOLTAGE OPTION: Interdata has recently introduced a series of processors which use MOS memories. This is not a problem in itself, but most of these processors do not have + or -16 volt power supplies. The TDC903 requires a small amout of -16 volts which is internally converted to -5 volts for use with the STORAGE MODULE interfaces. MiniComputer is currently working on a supply which will be resident on the board, however in the mean time, a secondary supply is required. To connect this supply to the TDC903, the following proceedure should be helpful.

1. Make certain that your power supply is set for 16 volts.
   DO NOT ASSUME THAT THE MANUFACTURER HAS DONE THIS FOR YOU!
2. Using a volt-ohm-meter, make certain that the minus output of the supply is not common with the chassis. If it is you must modify the supply or use a different supply.
3. Locate J4 between J0 and J1. Pin 1 is ground, connect the positive supply lead to this point. Pin 2 is no connection. Pin 3 is common with the -16 volt backplane pin.
4. If possible, connect the AC input of the supply to switched 'AC'. This will turn the power supply on and off with the computer.

LOGIC PROBE

The logic probe LED is turned off if either of its pins is pulled low.

OPERATION

The drives should be turned on after the CPU and should be turned off before the CPU.

OS/16 MT2 BOOTSTRAP PROCEDURE

OPERATION:
1. ENTER 50 SEQUENCE
2. INITIALIZE
3. CLOSE SMC903 BOOT CONTACTS
4. RUN 50 SEQUENCE

50 SEQUENCE:

```
0022  0000  S T A R T   H E R E
            (push RUN twice)
0034  8000
0036  0050

0050  D500
0052  0187
0054  4300
0056  008S

0078  710F  CDN+1,15
007A  70--  DN,0
007C  --F0  0,SDN
007E  0###  # OF OS IMAGE (OS16xxxx.###)
```

ERRORS ARE REPORTED IN THE FRONT PANEL DISPLAY AND IN THESE REGISTERS:

```
****    **** **** **** ****    **** **** **** ****
 R2         REGISTER  3            REGISTER F
```

STAGE 1:

1	BAD READ STATUS
2	BAD CYLINDER ID (SHOULD BE 0)
3	BAD HEAD & SECTOR ID
4	BAD ID CHECKSUM
5	BAD DATA CHECKSUM
6	STAGE 2 CODE INVALID -OR- ATTEMPT TO BOOTSTRAP ON A 32-BIT MACHINE USING A PACK FORMATTED ON A 16-BIT MACHINE

STAGE 2:

8	BAD READ STATUS	CURRENT MEMORY ADDRESS
9	BAD CYLINDER ID	CURRENT MEMORY ADDRESS
A	BAD HEAD & SECTOR ID	CURRENT MEMORY ADDRESS
C	BAD ID CHECKSUM	CURRENT MEMORY ADDRESS
D	BAD DATA CHECKSUM	CURRENT MEMORY ADDRESS
F	OS IMAGE NOT FOUND	0

OS/32 MT BOOTSTRAP PROCEDURE

OPERATION:
1. ENTER 50 SEQUENCE
2. SET PSW TO 00000000 (DTA FN SEL)
3. INITIALIZE
4. CLOSE SMC903 BOOT CONTACTS
5. RUN 50 SEQUENCE

50 SEQUENCE:

0050	D500	S T A R T H E R E
0052	0187	
0054	4300	
0056	0088	
0078	710F	
007A	----	
007C	----	
007E	0###	# OF OS IMAGE (OS32xxxx.###)

ERRORS ARE REPORTED IN THE FRONT PANEL DISPLAY AND
IN THESE REGISTERS:

****	**** **** **** ****	**** **** **** ****
R2	REGISTER 3	REGISTER F

STAGE 1:

1	BAD READ STATUS
2	BAD CYLINDER ID (SHOULD BE 0)
3	BAD HEAD & SECTOR ID
4	BAD ID CHECKSUM
5	BAD DATA CHECKSUM
6	STAGE 2 CODE INVALID -OR- ATTEMPT TO BOOTSTRAP ON A 32-BIT MACHINE USING A PACK FORMATTED ON A 16-BIT MACHINE

STAGE 2:

8	BAD READ STATUS	CURRENT MEMORY ADDRESS
9	BAD CYLINDER ID	CURRENT MEMORY ADDRESS
A	BAD HEAD & SECTOR ID	CURRENT MEMORY ADDRESS
C	BAD ID CHECKSUM	CURRENT MEMORY ADDRESS
D	BAD DATA CHECKSUM	CURRENT MEMORY ADDRESS
F	OS IMAGE NOT FOUND	0

See formatter listing for details.

COMMAND SUMMARY

The SMC903 uses two consecutive device addresses (default X'70' and X'71'), referred to in this summary as DEV and DEV1. OC and SS commands may be issued to either device code; RD and WD commands to DEV are for control functions, and to DEV1 are for data transfers.

DATA OPERATIONS

WD	DEV1,A(X2)	Write byte of data to RAM on board SMC903 at
WH	DEV1,A(X2)	current address; increment address.
WB	DEV1,A(X2)	Use only WH
RD	DEV1,A(X2)	Read byte of data from RAM on board SMC903 at
RH	DEV1,A(X2)	current address; increment address.
RB	DEV1,A(X2)	Use only RH

CONTROL OPERATIONS

WF DEV,A(X2) Load control register in SMC903 with one of the following (meaning depends on subsequent OC). Bits 9-15 also set RAM byte address (forced even).

1) cylinder
2) head, offset, and strobe early/late
3) sector
4) drive control bits (reset fault or rezero)
5) byte count for read or write

RH DEV,A(X2) First RH: read checksum register
Successive RH's: read sector counter
(First RH means first after any OC)
Sign bit is set on sector counter read.

SENSE STATUS

SS DEV,A(X2)
SS DEV1,A(X2)

BIT		
8		selected drive fault
9		0
10		selected drive not on cylinder
11		selected drive read-only
12	C	controller busy
13	V	examine status (fault, timeout or seek error)
14	G	sector overflow
15	L	selected drive offline or no drive selected.

OUTPUT COMMANDS

```
OC   DEV,A(X2)    CODE  0  reset controller
OC   DEV1,A(X2)         1  load byte counter
                        2  control tag to drive
                        3  seek
                        4  set head, offset, strobe
                        5  set sector
                        6  -
                        7  -
                        8  read (drive to RAM)
                        9  write (RAM to drive)
                       10  reset RAM address
                       11  reset checksum register
                       12  enable interrupt
                       13  -
                       14  -
                       15  autoload bootstrap code

                  CODES X'80'-X'81' select drives 0-1
```

CONTROL REGISTER MEANING FOR OC COMMANDS

```
CODE  0     control register is reset to 0
      1     byte count for read or write (2-n)
      2     BIT 15  0
                14  0
                13  0
                12  0
                11  fault clear
                10  0
                 9  return to zero
               0-8  0
      3     cylinder
      4     BITS 11-15  head
            BIT   9     offset direction forward
                  8     offset heads
                  7     read strobe early
                  6     read strobe late
      5     sector (0-255)
      6-7   unused
      8-15  control register is reset to 0
```

SMC903 versus TDC803

Most MiniComputer Technology software for the TDC803 may be used with the SMC903. Since the SMC903, like the TDC813, is a halfword device, special attention must be paid to its use with the AL command Normally, however, byte I/O instructions that work with the TDC803 will not work with the SMC903 (or TDC813, for that matter). The OS/16 and OS/32 bootstraps work with the SMC903 because, when run with the SMC903 the AL instruction reads in every other byte. A portion of the bootstrap program is written out in every other byte, and when read in rereads the first bootstrap sector using RH instructions and then operates normally.

In addition, the meanings of the bits sent to the drive with the control tag is different. Software such as the formatter/diagnostic recognizes the SMC903 by the fact that the sign bit is set when the sector counter is read and chooses bits patterns accordingly.

SECTOR MAPPING

MiniComputer Technology disk driver software makes use of a technique we call sector mapping. This procedure is referred to variously as sector interlace, sector staggering, alternate sectors, or sector interleaving. The reason for sector mapping is that typically some CPU processing time is required between accesses to the disk. The time may be necessary for I/O related tasks, such as moving data to and from buffers, or for the caller, who processes the data before asking for the next sector. A program engaged in a search through a file, for example, reads sector 0, scans it, and then reads sector 1, and so on. If mapping is not used, sector 1 will be passed while the program is scanning sector 0, and the program will have to wait nearly an entire revolution of the disk (16.7 milliseconds at 3600 rpm) before it can read sector 1. An alternative is to arrange sectors on the disk so:

PHYSICAL SECTORS: 0 1 2 3 4 5 6 7

LOGICAL SECTORS: 0 4 1 5 2 6 3 7

It may be that the processing takes more than one sector-time. In that case, every third sector may be used:

PHYSICAL SECTORS: 0 1 2 3 4 5 6 7

LOGICAL SECTORS: 0 3 6 1 4 7 2 5

This can be extended indefinitely, but a practical limit is every nth sector, where n is the number of sectors per track. This case, of course, is effectively back to no mapping at all.

Another problem arises when we finish with the last sector on a track and move to the next head:

PHYSICAL SECTORS: 0 1 2 3 4 5 6 7

LOGICAL SECTORS,
HEAD 0: 0 4 1 5 2 6 3 7

LOGICAL SECTORS,
HEAD 1: 0 4 1 5 2 6 3 7

When sector 7 of head 0 is processed, there is no time between it and sector 0 of head 1. A simple solution is to add the head number to the sector number (modulo the number of sectors per track) before multiplying by the mapping factor (2 in this example):

PHYSICAL SECTORS: 0 1 2 3 4 5 6 7

LOGICAL SECTORS,
HEAD 0: 0 4 1 5 2 6 3 7

LOGICAL SECTORS,
HEAD 1: 3 7 0 4 1 5 2 6

The controller always deals in terms of physical sectors, and the ID written by the formatter is in terms of physical sectors, so the driver must translate the logical sector it is given to a physical sector. The mapping subroutine in the SMC903 drivers performs the mapping function, and it may be changed to suit special requirements. Note that in OS/16 MT2 the map factor is written by the formatter as the first word in the ID of each sector, so that each pack has its own mapping factor. The optimum mapping factor depends on the speed of the CPU, whether a selector channel is available, and the

demands of the particular software using the controller.

With OS/16 MT2 and OS/32, the formatter writes the mapping factor in the ID of each sector on a pack, so that each pack carries its own mapping factor.

DIAGNOSTIC FORMATTER

The diagnostic formatter (TDC803-1-E) has two principal functions. First, it provides a thorough test of proper controller operation; second, it provides a means of writing the necessary ID and checksum information in each sector of the pack. In addition, if required, it writes the bootstrap code required for booting an OS/16 MT2 operating system. If assembled for 32-bit operation, a bootstrap record may be written that will load either OS/32 or OS/16 MT2.

The operating procedure for the formatter is as follows: load the program into a known location in memory above the interrupt table. Enter any optional control information into the appropriate data words at the beginning of the program. Start the formatter at the beginning (relative location 0). The formatter will run a series of diagnostic routines, and then format the pack. The formatting sequence consists of writing an entire track, reading it back, and then going on to the next track. After all tracks have been formatted, the entire pack is read a second time.

The following words may optionally be changed to tailor the operation of the formatter to the particular installation. The value in parentheses after each label is the default value. Consult the formatter listing for specific alternative values.

PATN (0001) and PINC (00FF) control the actual data written in each sector. The first halfword will be PATN; the second will be PATN+PINC; the third will be PATN+PINC+PINC; and so on. Worst case patterns (as specified by CalComp) may be used by entering them into PATN with PINC=0000. A recommended double spiral pattern is PATN=0000 and PINC=00FF.

RWFLAG (0003) may be set to read-only, write-only or write/read operation. The options are of use primarily in diagnostic procedures; for example, if a controller has data errors, but another controller works, a portion of a pack should be formatted in write-only mode with the bad controller and then read in read-only mode with the good controller, and vice versa, to determine whether the bad controller is faulty during reads, writes or both.

TYPE (0002) designates the actual drive type being used.

DRIVE (0000) designates the drive number (0-1) on which the pack to be formatted is mounted.

DEVICE (0070) is the device addess of the controller.

ALL (0A11) tells the formatter to format all sectors of the pack. If ALL is set to 0000, only those sectors within the inclusive limits LOSEC-HISEC, LOHEAD-HIHEAD, LOCYL-HICYL will be formatted. If ALL is non-zero, those limit words will automatically be set to cover the entire pack.

DCNT (0100) specifies the number of data bytes per sector. The formatter will compute the number of sectors per track from DCNT and TYPE.

OFF1 (0000) and OFF2 (0000) allow head offsets and read strobe delays to be specified in pass 1 (write and read) and pass 2 (reread) respectively.

OS (0001) specifies the operating system that the pack is to be formatted for. Specifying 0001 or 0002 causes a bootstrap routine to be written. Specifying 0001 also causes an additional halfword to be included in the ID. On a 32-bit system, specifying 0001 will generate a bootstrap record that will load either OS/32 or OS/16 MT2. On a 16-bit system, specifying 0001 will generate a bootrap record for OS/16 only. On either a 16- or 32-bit system, specifying 0002 will generate a bootstrap record for 16-bit DOS, although DOS does not support the SMC903 at

this time.

MAPFAC (0005 or 0007) is relevant only when OS is set to 0001. It specifies the mapping factor to be used by the operating system.

FLAGBAD (0000) may be set non-zero to cause bad tracks to be flagged. The OS/16 MT2 and OS/32 MT drivers recognize the flagged tracks, and the bad tracks are flagged in the system bit map when a READCHECK is performed by DISCINIT, DISCHECK, or the OS/16 INIT command. The number of flags is indicated in the left display at the end of the program (display D). We suggest that PATN be set to 0001 and PINC to 00FF. The offset options OFF1 and OFF2 may also be used to create a worst-case condition. HLTFLG may be set to 0000 to prevent halting on errors.

NDEV (0100) can be set to 0200 or 0400 on 32-bit machines with extended device addressing.

REWRIT (0001) may be changed to 0000 to prevent sectors from being rewritten after an error is detected. This is of use primarily in diagnostic applications.

CHKDTA (00CD) may be set non-zero to cause each data byte to be checked during reads. Otherwise, the checksum is used to check the data.

CONT (0000) may be set non-zero to cause the formatter to run continuously, again chiefly for diagnostic purposes. Passes are counted in PASSES.

HLTFLG (0001) causes the formatter to halt on errors. If set to zero, errors are counted in ERRORS and the most recent 32 errors are saved in the ERRSAV area. When 64K errors are counted, the program will halt unconditionally.

INTRPT (0001) may be set to 0000 to prevent the formatter from using interrupts.

SELCH (0000) may be set to the device address of the selector channel (usually 00F0). If a selector channel is available, this will result in somewhat faster operation.

CYLINC (0001) sets the cylinder step size. For example, if CYLINC=0002, only even cylinders will be formatted. If CYLINC=0000, cylinders are chosen randomly.

OA (0000) is an option used to give a controller an abbreviated checkout without formatting the entire pack.

RNDLIM (0100) specifies the number of passes in the RAM random-data test. If RAM errors are suspected, this value may be increased at the expense of longer running time.

LOSEC-HICYL: see ALL, above.

DIAGNOSTIC FORMATTER ERRORS

The formatter uses the front panel to report errors. An error code appears in the left four bits, and two 16-bit numbers appear in the next 32 bits. For older machines, such as the Model 70, the error code may be found in register F, the left halfword in register E, and the right halfword in register D. For certain error types, the left four bits of the left halfword will indicate an error subtype. The last 32 errors are saved in the ERRSAV area and may be displayed by starting the program at the instruction called DISPLAY SAVED ERRORS. Push RUN twice for each error; the first display will be the normal error display (see below) and the second will be the 0 (cylinder, head & sector) display. When all saved errors have been displayed, the D display is displayed.

After a halt, the formatter may be restarted simply be pressing RUN. However, if anything has been changed (like the program counter) it must be restarted at the B UNHALT instruction called RESTART AFTER ERROR HALT.

NORMAL OPERATION

```
****      **** **** **** ****      **** **** **** ****
 0                  CYLINDER           HEAD     SECTOR
```

This is the normal display of the formatter, except for the initial diagnostic section. The display is of the current cylinder, head and sector being formatted.

WRITE STATUS

```
****      **** **** **** ****      **** **** **** ****
 1                   STATUS           HEAD     SECTOR
```

A status error was encountered after a write.

READ STATUS

```
****      **** **** **** ****      **** **** **** ****
 2                   STATUS           HEAD     SECTOR
```

A status error was encountered after a read.

ID CHECKSUM

```
****      **** **** **** ****      **** **** **** ****
 3          CHECKSUM REGISTER                 CHECKSUM
```

The ID checksum computed by the controller's checksum register did not match the checksum read from the pack.

ID

****	****	****	****	****	****	****	****	****
4	1		CYLINDER ID				EXPECTED	
	2		HEAD ID				EXPECTED	
	3		SECTOR ID				EXPECTED	
	4		MAP FACTOR				EXPECTED	

One of the ID items was in error. The item actually read is on the left and the expected value is on the right. The map factor will have the drive type (TYPE) in its left byte and the map factor in its right byte.

DATA

****	****	****	****	****	****	****	****	****
5		HEAD		SECTOR		SD	SC	RD	RC

A data error was encountered. SD, SC, RD and RC each be 0, A or F and indicate that the corresponding test was passed (A), failed (F) or not made (0). The SD test is a word-for-word check of data read with the SELCH. The SC test is of the checksum after a SELCH read. The RD and RC tests are the same, but for transfers using the RH instruction rather than the SELCH. The general registers also contain details of interest; consult the formatter listing page 4 for their contents. They are not loaded if the error is being displayed from the ERRSAV area.

ILLEGAL FLAG

****	****	****	****	****	****	****	****	****
	6		HEAD		SECTOR		CYLINDER	

An attempt is being made to flag head 0 of either cylinder 0 or 1. These tracks are necessary for the bootstrap and OS volume ID sector and must be good.

PARAMETER ERRORS

****	****	****	****	****	****	****	****	****
7	0					DEVICE NUMBER ODD		
	1					NO RESPONSE FROM DEVICE		
	2					ILLEGAL DRIVE NUMBER		
	3					ILLEGAL DRIVE TYPE		
	4					SECTOR LIMITS ERROR		
	5					HEAD LIMITS ERROR		
	6					CYLINDER LIMITS ERROR		
	7					DATA BYTE COUNT (DCNT) ODD		
	8					DCNT NOT 256 WITH OS		
	9					NO RESPONSE FROM SELCH		
	A0					FLAGBAD SET AND OS NOT = 1		
	A1					FLAGBAD SET, BUT NOT ALL SECTORS SPECIFIED		
	A2					FLAGBAD SET AND RWFLAG NOT 3		
	B					PROGRAM LOADED TOO LOW		

These errors generally indicate that one of the user-selected paramters is incorrect or (in the case of error 1) that the controller is not responding to its address. Error 7-B indicates that the program was loaded so low that it overlapped the interrupt table.

RAM TEST

****	****	****	****	****	****	****	****	****
8	0				APPARENT RAM SIZE			
	1			SIZE	RAM TOO SMALL			
	2			BLOCK	BITS			
	3			ADDRESS	CONTENTS			
	4			BLOCK	BITS			
	5			ADDRESS	CONTENTS			

The 0 display indicates that the RAM test is running. If an error is encountered, one of the other subtypes will be displayed. Subtype 1 indicates that the RAM is too small for the requested sector size. This can also be caused by bad memory chips or faulty addressing. Subtypes 2-4 are various data pattern tests. The word BLOCK indicates which 256-byte blocks are in error. BLOCK=1 indicates that the errors were in the first 256 bytes; BLOCK=2 indicates the second 256 bytes; BLOCK=4 indicates the third 256 bytes; BLOCK=8 indicates the fourth 256 bytes; and so on. Errors in multiple blocks are indicated by the sum of these codes. For example, BLOCK=5 indicates errors in both the first and third 256-byte blocks. When errors are encountered, an error pattern is generated with bits on in the bit position of each erroneous bit. The logical OR of all the error patterns is displayed as OR. By consulting the schematic page C, BLOCK and OR can usually be used to For the SMC903, the 16-bit OR represents the 16-bit width of its RAM. For the TDC803, the left byte of the OR word represents even byte addresses and the right byte represents odd addresses. No particular significance is attached to this distinction; it is done for compatibility between the TDC803 and SMC903. BLOCK, OR and page C of the schematics may be used pin the error down to a single memory chip, if indeed that is the problem. In test 3, the byte address of each halfword in the RAM is written out and checked. In test 5, the random RAM addressing feature of the SMC903 is checked.

CHECKSUM REGISTER TEST

****	****	****	****	****	****	****	****	****
9	0				TEST IN PROGRESS			
		CHECKSUM REGISTER			EXPECTED			

The all-zero display indicates that the checksum register test is running. A large number of data patterns are sent to the controller. The formatter computes the correct checksum (EXPECTED) and then reads the checksum register to determine whether is was properly computed by the controller.

SECTOR COUNTER TEST

****	****	****	****	****	****	****	****	****
A	0				SECTOR COUNTER			
	1			NSEC	LOW COUNT	HIGH COUNT		
	2	SECTOR COUNTER WAS			AND COUNTED TO			

The number of sectors is computed from TYPE and DCNT. If the sector counter does not count from 0 to NSEC-1 or NSEC, error subtype 1 reports the lowest and highest sector numbers read from the sector counter. A bad high count generally is caused by incorrect sector jumpering in the drive. Cable problems are suspect if the counter is not counting at all.

ILLEGAL DEVICE INTERRUPT

****	**** **** **** ****	**** **** **** ****
B	DEVICE	STATUS

An interrupt was received from some device other than the SMC903, or the SMC903 identified itself incorrectly to the CPU when it interrupted. Try pressing INI and restarting the formatter.

INTERRUPT TIMEOUT

****	**** **** **** ****	**** **** **** ****
C	0	SMC903 STATUS

No interrupt was received from the controller when it was expected. This error is most often caused by miswiring the RACK0/TACK0 jumpers on the backplane.

FORMATTER DONE

****	**** **** **** ****	**** **** **** ****
D	FLAG COUNT	ERRORS

This pattern is displayed upon completion. ERRORS is a count of errors encountered. FLAG COUNT is the number of bad sectors (greater than or equal to the number of bad tracks). If both values are FFFF, the bad-sector table has overflowed. This indicates either a serious hardware malfunction or a VERY bad pack. Note that the number of errors encountered on each head is stored in HDTAB. If many errors occur and they are restricted to only a few heads, a drive problem must be suspected.

RAM READER

****	**** **** **** ****	**** **** **** ****
E	RAM ADDRESS	(ADDRESS+0)(ADDRESS+1)

The stand-alone RAM reader may be started at its beginning to read out the contents of the controller's RAM buffer. Two bytes are displayed at a time, together with the address of the first byte. Press RUN repeatedly to step through successive addresses.

DRIVE STATUS

****	**** **** **** ****	**** **** **** ****
F	DRIVE #	STATUS

If this display appears for any lengthy period of time, it indicates a status problem. Most likely causes are: read-only switch on with writing requested; drive deated; non-existent drive selected; cable or terminator not properly plugged in.

OS/32 MT

The SMC903 controller may be used with OS/32 MT R02 by making the entries to the CUP/32 input as specified in the OS/32 driver listing and adding the DCB and driver objects to the driver library. The DCB must be first. Alternatively, CUP/32 may be first directed to the standard driver library. When it encounters the end of the library it will pause with an EOF or end of volume message. Close logical unit 4, assign it to the SMC903 DCB object file, and continue. When the end of the DCB is encountered, CUP will pause again. Close logical unit 4, assign it to the SMC903 driver object and continue.

The type of drive to be used need not be specified at SYSGEN time; it is determined by the driver when the drive is MARKed ON by the operator for the first time. When the system is first loaded, or when a new pack with a different mapping factor is loaded, the driver will reject all I/O requests except a read of sector zero, which the MARK ON command will do. This means that in order to run DISCINIT, the drive must be MARKed ON and then OFF, or DISCINIT will terminate with an I/O error. Driver assembly options include use of the SELCH, ISR or NSU operation, and read-check offset. The read-check offset option allows the drive's heads to be offset during a read-check operation to make error detection more likely. It is set according to the instructions for setting OFF1 and OFF2 in for formatter. Setting NSU allows the driver's interruptable routines to run at higher priority. Setting ISR lets the driver run as an interrupt service routine (ISR) rather than an event service routine (ESR), thus avoiding the system overhead of starting up an event service. ISR should not be used if very fast interrupt response is necessary.

While the DCB is identified with device code 063, the actual device code assembled into the DCB is 054. This is for proper operation with the Interdata disk utilities, which become confused when confronted with an unfamiliar device code. For similar reasons, a fake sectors per track entry appears in the DCB, and the real one is in a new location at the end of the DCB.

Revision F0 supports Cartridge Module and Mini Module devices (CMD/MMD). Since for simplicity all DCBs are assembled to be the same, a new sysgen step is necessary to customize the DCBs for CMD/MMD devices. These devices are characterized by being logically not one, but two devices. CMD devices have both fixed and removable media; MMD devices have special head-per-track areas. Two DCBs exist for each one of this class of devices. One DCB is normal, and the other needs special treatment. The special portions of these drives is accessed by setting special bits in the cylinder and/or head numbers. Also when two DCBs refer to the same device, they must know about each other in order to avoid interference.

To supply this information in the DCBs, MiniComputer Technology has written a program to modify OS32 images (on disk) according to special additional information in the CUP input file. The name of this program is "VMR". VMR takes as input data written on CUP input lines surrounded by angle brackets. CUP treats this data as a comment. Descriptive information is placed on the line containing the device definition.

The first VMR argument (within the angle brackets) is the device type, as is used by the formatter. The second argument is the name of another device. If the second argument is given, the two DCBs are hooked together. Thus for CMD support the CUP input lines might be:

```
4 CMDF:,074,063,D              <1E,CMDR:>
* CMDR:,074,063,D
```

Note that "CMDR:" needs no special treatment since only the fixed media portion of the CMD ("CMDF:") requires special bits for access.

OS/16 MT2 R01

The SMC903 controller may be used with OS/16 MT2 R01 by using the MiniComputer Technology SYSG.CSS command to perform the sysgen. The supplied program POSTCUP edits the output of CUP16 for MCT controllers. In assembling the disk driver, support for drives not used may be eliminated by setting the appropriate TYPn parameters to 0. The driver automatically determines whether a selector channel is present (if specified in the CUP DEVICE statement) and makes data transfers accordingly.

Packs to be used under OS/16 MT2 must be formatted with the formatter's OS flag set to 0001. The map factor is used, and should be chosen after considering the particular application. A map factor of 0007 seems good for a 7/16 with a T50 and no selector channel. A larger map factor would be used with a T80 or T300, and a smaller one with faster CPU's or with a selector channel.

To perform a system generation, the driver object must be in a file named TDCDVR16.OBJ and the CUP input statements must be in a file with the name CUPIosid. Giving the CSS command SYSG osid causes a compete sytem generation to be performed. The output of CUP, which does not know about the MiniComputer Technology controllers, is editted by POSTCUP (supplied by us). The EXEC, FMGR and CMDP modules are assembled, and a system image is generated by TET. CUPOosid (output of POSTCUP) remains, as do EXEC.SCT, FMGR.SCT and CMDP.SCT. The new operating system image is left in OS16osid. For example, with CUP input in CUPIMCT.101, the command SYSG MCT.101 will generate CUPOMCT.101 and OS16MCT.101. The alternative command SYSGX osid should be used in systems with no printer.

DEVICE 54,7n,TDCn:,F0,70 "name"

where the low order two bits of n is the drive number (0-3), F0 is the selector channel address, and 70 is the controller address. The may be changed, of course. If the controller is not to be plugged in under a selector channel, the controller address may be substituted for the selector channel address. The device name is enclosed in double quotes and is separated from the controller address by exactly one blank. We usually use TD1, TD2, etc., but any name of up to four characters may be used, including DSC1, DSC2 etc. if they are not used for other devices.

The DEVICE statement for a floppy disk is:

DEVICE 48,9n,FDCn:,F0,90 "name"
DEVICE 49,9n,FDCn:,F0,90 "name"

Device code 48 is used for single density and 49 is used for double density. Although a selector channel is not normally used with a FDC203, the controller may be plugged in on a SELCH bus, in which case the selector channel must be specified. Otherwise, change the F0 to the controller address. Our standard names are FD1, FD2, etc.

EXAMPLES:

DEV 54,70,TDC0:,70,70 "SM1"
generates an entry for a drive without a SELCH which is drive 0 on a controller with a device address of 70.

DEV 54,71,TDC1:,70,70 "SM2"
generates a second drive on the same controller.

```
DEV 54,70,TDC0:,F0,70 "SM1"
```
generates a drive whose controller shares the busy flag of the SELCH and which may therefore use the SELCH.

The output of POSTCUP must be specially editted if dual-spindle floppy drives such as the Diablo 12 are to be used. Contact MiniComputer for special instructions.

The standard address for a second SMC is 72. A system with two drives on one controller and one on the second and separate selector channels would be specified as follows:

```
DEV 54,70,TDC0:,F0,70 "SM1"
DEV 54,71,TDC0:,F0,70 "SM2"
DEV 54,74,TDC0:,F1,72 "SM3"
```

HARDWARE DESCRIPTION

SECTOR COUNTER & COMPARATOR (schematic page D)

The sector counter is composed of ripple counters E3 & F3. The leading edge of each sector pulse increments the counter, and index pulses reset it to 0. The output of the sector counter (SEC1-SEC128) is made available to the software. The sector counter bits are continuously compared to the requested-sector register E1 & F1, and when they are equal, SEC= is high. When a new drive is selected, or a drive has been off line, E7/5 is low and suppresses SEC= until an index pulse from the drive resyncs the sector counter. SCLR- resets the requested-sector register in anticipation of a bootstrap read.

DRIVE SELECTION (D) A drive-select command is an OC with bit 8 set and the drive address in bit 15. The drive address is clocked into the H10/9 flipflop. The new drive address is compared to the old drive address, and if it is different, NEWDR- is pulsed low to let the sector counter be resynced. The drive select oneshot F8/12 is pulsed to select the drive.

WH DEV & OC (D)

A WH DEV command loads the accumulator contents into the register D1 & D3, which serves as a buffer register from which the sector register, drive bus, and byte counter are loaded. OC commands with bit 8 off are decoded in the selected controller by decoder H16, and low-going pulses appear at its outputs. Note that OC commands 8-15 reset the buffer register to 0.

DRIVE TAGS (D)

The output lines to the drive consist primarily of an 11-bit bus and three tag lines that define the meaning of the bus at any given time. The buffer register contents always appear on the bus, and the tag pulses are initiated by CTAG!- (control tag), HEAD!- (set head tag), and SEEK!- (set cylinder tag). The tags are about 1.5 microseconds long. The control tag line is also activated during a read or write (RWREQ), when the bus lines carry such meanings as head select, read gate, and write gate. The read strobe timing bits and offset bits are included in the head command for software convenience, but must be stored in flipflops, since they are examined by the drive only during a read.

RH DEV1 & SELCH READS (A,B,C)

The RH DEV1 command is used to read a word from the RAM buffer. The current RAM byte is always available on the RAMnn and RAMnnB lines. DR generates RDRAM!, which gates the byte into the CPU. The trailing edge of RDRAM! increments the RAM address counter (page C), which is composed of an 8-bit counter and a 2-bit shift register that selects the bank of 256 words being accessed. A selector channel read looks to the controller the same as an RH DEV1 command from the CPU. RAMBZ- inhibits closely-following DR strobes (and SYNO) until the next byte is ready to be read.

WH DEV1 and SELCH WRITES

The WH DEV1 command writes a word into the RAM buffer and increments the buffer address. DARAM!- strobes the data byte into the tri-state buffer B1-B4 (page B). The trailing edge of DARAM!-, as WRAM!, initiates a RAM-write cycle. The cycle is composed of two overlapping periods: WRAM- and DRAM-. WRAM- is applied to the RAM chips, switching their outputs to the high-impedance state. About 125 ns later, DRAM- enables the data buffer, applying the data byte to the I/O pins of the RAM. After WRAM- ends, DRAM- stays active for about 50 ns, to satisfy the data hold requirements of the RAM. The RAM address counter is incremented at the end of DRAM-. A selector channel write sequence looks the

same as a fast sequence of WH DEV1 commands.

CHECKSUM REGISTER (C)

The checksum register is initially reset to zero by RSTCS!-. Each time a word is read from or written in the RAM, it is also 'added' to the checksum register by being exclusive-ORed with the prior contents rotated three bits and with one bit of the prior contents (CS00) inverted. The checksum register is read with an RH DEV command.

The SMC903 adds two bytes at a time to the checksum register in a manner compatible with two successive bytes in the TDC803.

READ FROM DISK TO RAM

A read is requested after resetting the RAM address, setting a byte count, and issuing a read command, which sets RREQ. When the disk has finished any pending seek (ONCYL) and the requested sector has been found (SEC=), the trailing edge of the sector pulse sets RSF (read sector found, page A). The read command resets the buffer register. RWREQ enables the control tag and the head-select bus line. RSF enables the read gate bus line after a delay to allow for mechanical tolerances. After the delay, a search is started for the read sync pattern (X'A9'). The pattern is detected by the gate E5 (page B). Data bits are shifted into the read shift register D2 & D4 by the disk's clock. When the sync mark is found, the next bit clock turns on RSYNF (read sync found), which in turn enables the bit counter. At the last bit of every byte (halfword), the bit counter H12 generates BIT15, which causes the tri-state buffer C1-C4 to be loaded with the new byte at the next bit clock. BIT15 also generates WRAM!, and the new halfword gets written into the RAM as for WH DEV1, described above. When the byte count is satisfied, WCUP turns off RREQ, which turns off RSF, which turns off RSYNF. BUSY is also turned off. The program can now reset the RAM address counter and read the data from the RAM with RH's or the selector channel.

The byte count specified by the software must be even.

WRITE FROM RAM TO DISK

A write is initiated similarly to a read. A delay (WDLAY-) is enforced between WREQ and WSF to allow the head-select to stabilize before turning on the write gate. WSF is also suppressed if the disk's read-only status line is active. WSF (write sector found) enables the bit counter and the write gate bus line. Every 8 (16) bits, BIT15 causes another byte to be clocked into the write shift register C6 & C8 on page B and the RAM address to be incremented. The disk-supplied bit clock shifts the data serially out to the disk. Only one of the write data drivers is enabled at any one time.

ERRORS

Two errors are directly detected by controller logic. OFLO is caused by the occurrence of a sector mark during a read or write before the byte count is counted down to zero. This is generally due to failure to detect the read sync mark, to improper drive sector jumpering, or to an excessively high byte count. TIMEOUT is generated when a requested operation (read, write, or seek) is not completed within 2-3 seconds. For example, TIMEOUT would be generated if a write were attempted on a read-only drive. This would be considered a software error, since the program should have checked the read-only status bit before requesting a write.

BOOTSTRAP (A)

Closing the bootstrap contacts (J3) after SCLR causes sector 0, head 0, cylinder 0, drive 0 to be read into the RAM. Once the bootstrap is triggered and drive 0 is ONCYL, BSPHD pulses and turns on BSRTZ. These signals are sent to the drive to cause a recalibrate to cylinder 0, and to reset the head address register. When ONCYL- happens as a result of the recalibrate, BSRD- causes RREQ, and a normal read ensues. A byte count of 4096 is implied by the resetting of the byte counter, so a sector overflow (OFLO) usually causes WCUP. The autoload command issues BSP2!- and BSP2!- resets the byte counter, the RAM address counter, and OFLO. The autoload instruction then reads the RAM contents into memory starting at location 0080. Since the autoload is a byte-oriented instruction, special provision must be taken with the SMC903. See the OS/16 and OS/32 bootstrap in the formatter for an example.

CPU and DRIVE

An understanding of the operation of both the CPU and the drive is necessary to thoroughly understanding the SMC903. Read chapter 5 of the Interdata User's Manual and the appropriate drive reference manual. More CPU information can be found in its Maintenance Manual.

MCT SMC900/TDC800 SERIES DISK FEATURES

COLUMN HEADINGS

Tp Cd	Disk specification type code used by formatter program.
Mf	Manufacturer of disk drive.
	CC Calcomp CD CDC MX Memorex AM Ampex
	MD Micro Data OK Okidata FJ Fujitsu KN Kennedy
Model	Disk drive model; name and unformatted size in Megabytes.
	RMV indicates a removable pack in a dual medium drive;
	FIX indicates a fixed pack, or fixed heads.
Hd	Number of usable recording surfaces (Heads).
Cyl	Number of cylinders (Seek positions).
Pm Bt	Number of null bytes and sync code which preceed the sector data.
Bytes Track	Number of bytes per unformatted track.

The followings are reported for 2 standard sector data sizes.

Ovr Hed	Number of non-data (over head) bytes required per sector.
NSC	Number of sectors per track.
Size MBts	Total disk capacity in Mega-Bytes.
SSIZ	Number of bytes actually allocated to each sector.

To compute the parameters for non-standard data sizes proceed as follows:

1) Compute MSIZE the minimum required sector size:
   Add the `Ovr Hed` per sector to the desired data bytes per sector.
2) Compute NSC the number of sectors per track:
   Divide `Bytes Track` by MSIZE, discard any remainder.
3) SSIZ is found by dividing `Bytes Track` by NSC.
   SSIZ, or a simple modification of it, must be set in switches or by strappings in most disk drives.

MCT SMC900/TDC800 SERIES DISK FEATURES

							NOVA 512 Bytes/Sect				INTERDATA 256 Bytes/Sect			
Tp Cd	Mf	Model	Hd	Cyl	Pm Bt	Bytes Track	Ovr Hed	NSC	Size MBts	SSIZ	Ovr Hed	NSC	Size MBts	SSIZ
1	CC	T25	5	407	38	13440	60	23	24.0	584	58	42	21.9	320
2	CC	T50	5	815	38	13440	60	23	48.0	584	58	42	43.8	320
3	CC	T80	5	815	58	20160	86	33	68.9	610	84	59	61.5	341
4	CC	T200	19	815	38	13440	60	23	182.4	584	58	42	166.5	320
4	MX	677/200	19	815	38	13440	60	23	182.4	584	58	42	166.5	320
5	CC	T300	19	815	58	20160	86	33	261.6	610	84	59	233.9	341
6	CD	SM/40	5	411	58	20160	86	33	37.4	610	84	59	31.0	341
7	CD	SM/150	19	411	58	20160	86	33	131.9	610	84	59	117.9	341
8	MX	677/100	19	411	38	13440	60	23	92.0	584	58	42	84.0	320
9	MX	601/25	4	350	22	17920	40	32	22.9	560	38	60	21.5	298
10	MX	601/50	8	350	22	17920	40	32	45.9	560	38	60	43.0	298
11	MX	601/75	12	350	22	17920	40	32	68.8	560	38	60	64.5	298
12	MD	REFLEX/12	2	350	22	17920	40	32	11.5	560	38	60	10.8	298
13	MD	REFLEX/37	6	350	22	17920	40	32	34.4	560	38	60	32.3	298
14	MD	REFLEX/62	10	350	22	17920	40	32	57.3	560	38	60	53.8	298
15	CD	SM/80	5	823	58	20160	86	33	69.5	610	84	59	62.2	341
16	CD	SM/300	19	823	58	20160	86	33	264.2	610	84	59	236.2	341
17	OK	OKIDATA/12	2	339	32	18560	60	32	11.1	580	58	59	10.2	314
18	OK	OKIDATA/24	4	339	32	18560	60	32	22.2	580	58	59	20.5	314
19	OK	OKIDATA/37	6	339	32	18560	60	32	33.3	580	58	59	30.7	314
20	OK	OKIDATA/49	8	339	32	18560	60	32	44.4	580	58	59	41.0	314
21	OK	OKIDATA/61	10	339	32	18560	60	32	55.5	580	58	59	51.2	314
22	OK	OKIDATA/74	12	339	32	18560	60	32	66.6	580	58	59	61.4	314
23	CD	MMD/12	2	320	58	20160	86	33	10.8	610	84	59	9.7	341
24	CD	MMD/25	4	320	58	20160	86	33	21.6	610	84	59	19.3	341
25	CD	MMD/82	10	411	58	20160	86	33	69.4	610	84	59	62.0	341
26	FJ	50MB	3	815	28	20480	46	36	45.1	568	44	68	42.6	301
27	CD	CMD-RMV/16	1	808	58	20160	86	33	13.7	610	84	59	12.2	341
28	CD	CMD-FIX/16	1	808	58	20160	86	33	13.7	610	84	59	12.2	341
29	CD	CMD-FIX/48	3	808	58	20160	86	33	41.0	610	84	59	36.6	341
30	CD	CMD-FIX/80	5	808	58	20160	86	33	68.3	610	84	59	61.0	341
31	AM	160MB	5	1645	58	20160	86	33	139.0	610	84	59	124.2	341
32	CD	MMD-FIX/.9	4	12	58	20160	86	33	0.81	610	84	59	0.72	341
33	CD	MMD-FIX/1	5	10	58	20160	86	33	0.84	610	84	59	0.76	341
34	CD	MMD-FIX/2	5	20	58	20160	86	33	1.67	610	84	59	1.51	341
35	KN	5300/14	1	700	58	20160	86	33	11.8	610	84	59	10.6	341
36	KN	5300/42	3	700	58	20160	86	33	35.5	610	84	59	31.7	341
37	KN	5300/70	5	700	58	20160	86	33	59.1	610	84	59	52.9	341

CDC CMD and MMD SUPPORT

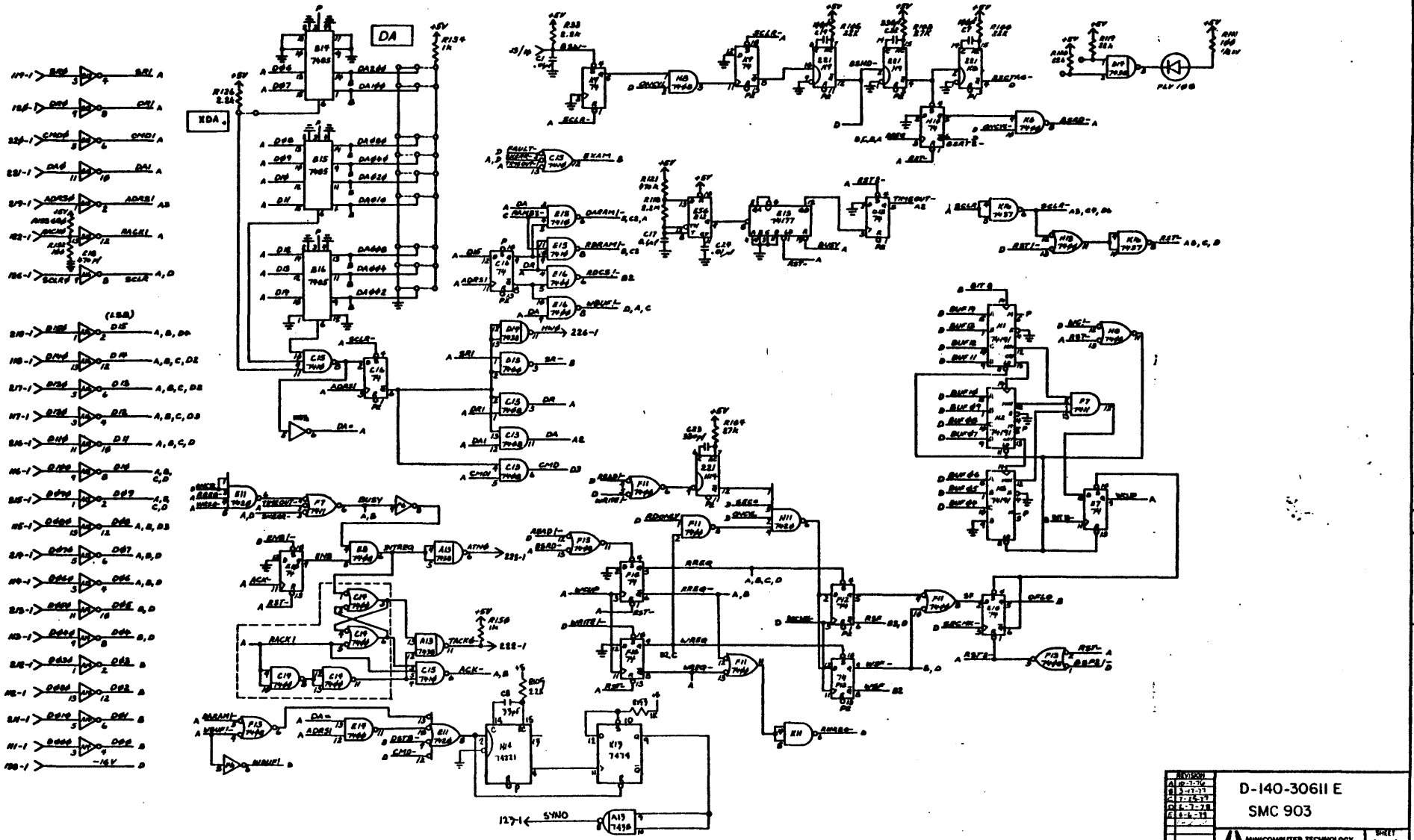
Revision F0 of the disk driver (TDCDVR32) supports Cartridge Module and Mini Module devices (CMD/MMD). Since for simplicity all DCBs are assembled to be the same, a new sysgen step is necessary to customize the DCBs for CMD/MMD devices. These devices are characterized by being logically not one, but two devices. CMD devices have both fixed and removable media; MMD devices have special head-per-track areas. Two DCBs exist for each one of this class of devices. One DCB is normal, and the other needs special treatment. The special portions of these drives is accessed by setting special bits in the cylinder and/or head numbers. Also when two DCBs refer to the same device, they must know about each other in order to avoid interference.

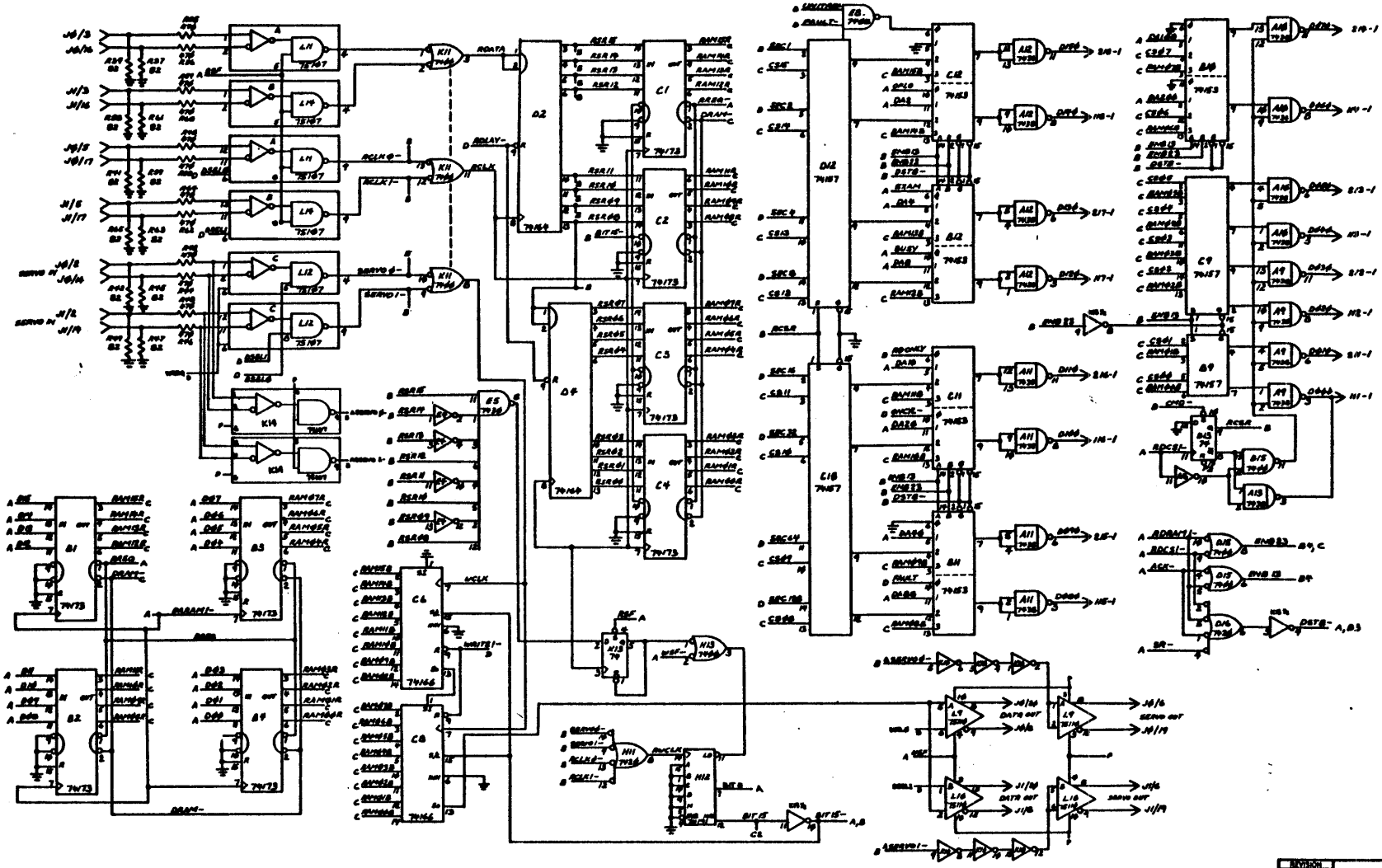
To supply this information in the DCBs, MiniComputer Technology has written a program to modify OS32 images (on disk) according to special additional information in the CUP input file. The name of this program is "VMR". VMR takes as input data written on CUP input lines surrounded by angle brackets. CUP treats this data as a comment. Descriptive information is placed on the line containing the device definition.

The first VMR argument (within the angle brackets) is the device type, as is used by the formatter. The second argument is the name of another device. If the second argument is given, the two DCBs are hooked together. Thus for CMD support the CUP input lines might be:

```
4 CMDF:,074,063,D          <1E,CMDR:>
* CMDR:,074,063,D
```

Note that "CMDR:" needs no special treatment since only the fixed media portion of the CMD ("CMDF:") requires special bits for access.


DA
XDA
+5V
D-140-30611 E
SMC 903
MINICOMPUTER TECHNOLOGY
SHEET 1 OF 4



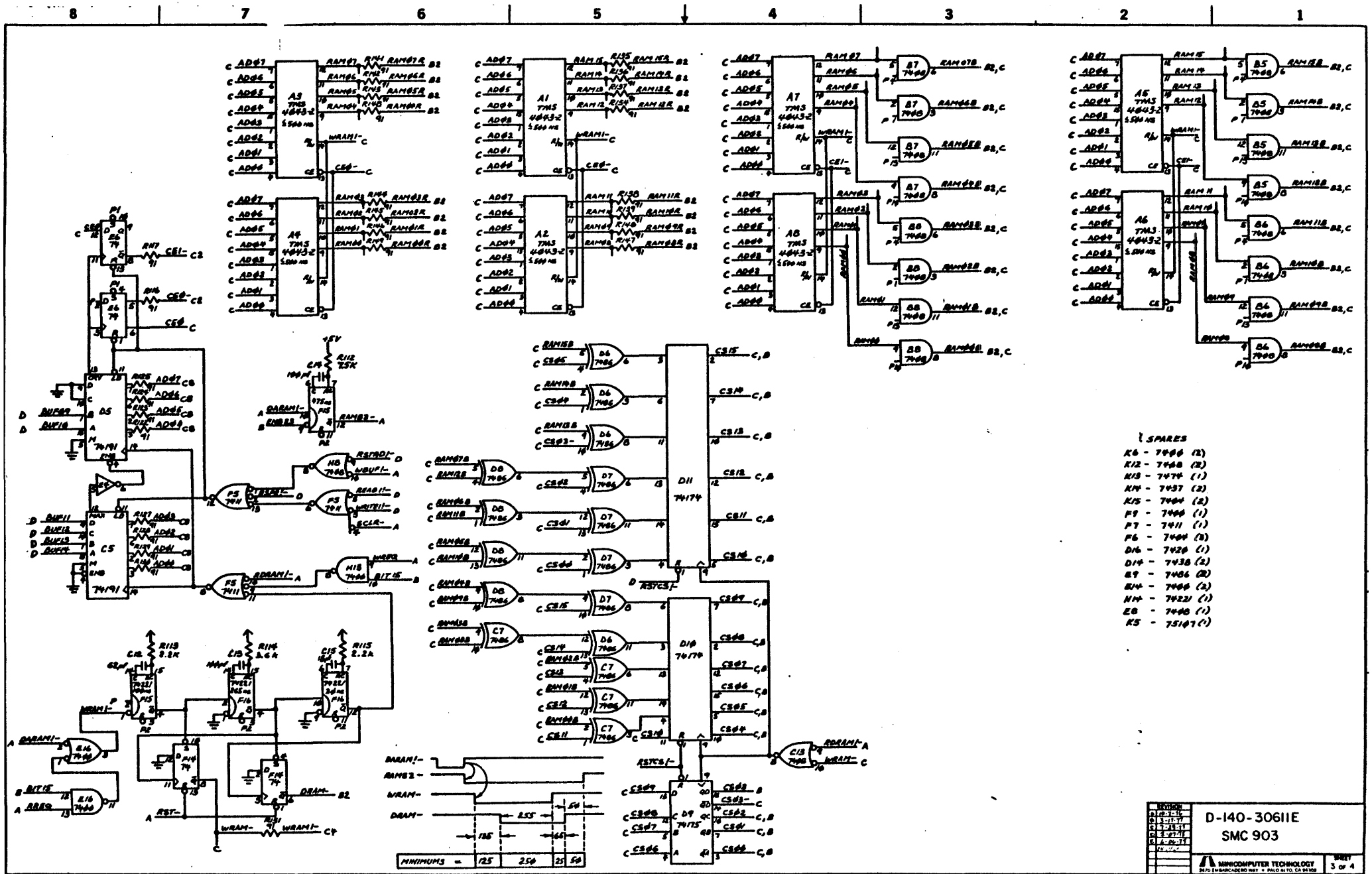
REVISION	
A	2-76
B	3-17-77
C	7-25-77
D	8-02-77
E	10-10-77

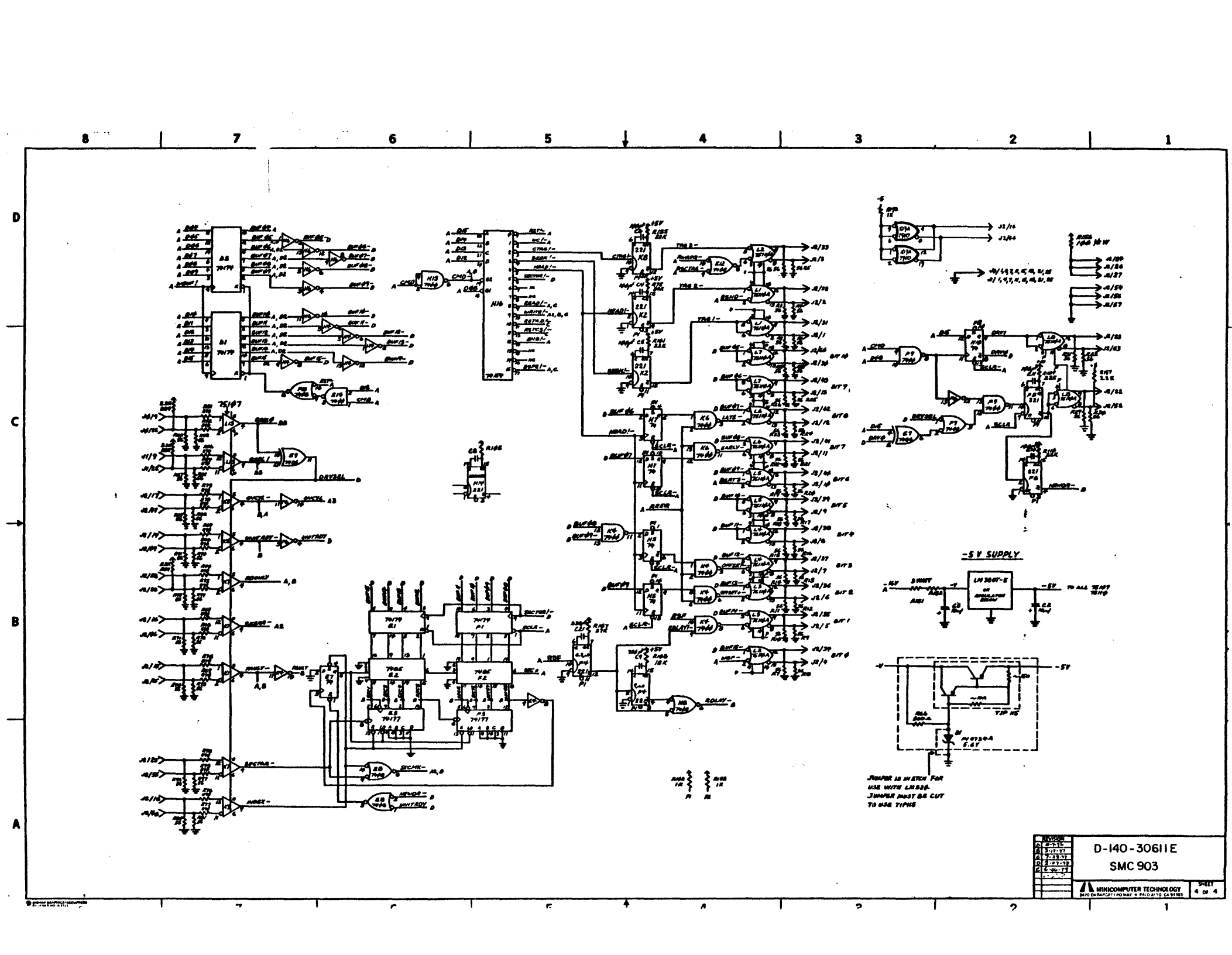
D-140-3061IE

SMC 903

MINICOMPUTER TECHNOLOGY
2470 EL CAMINO REAL • PALO ALTO CA 94306

SHEET 2 OF 4


SPARES
K6 - 7406 (2)
K12 - 7408 (2)
K13 - 7474 (1)
K14 - 7437 (2)
K15 - 7404 (2)
F9 - 7400 (1)
F7 - 7411 (1)
F6 - 7409 (3)
D16 - 7420 (1)
D14 - 7438 (2)
E9 - 7486 (2)
E14 - 7400 (2)
H14 - 74221 (1)
E8 - 7408 (1)
K5 - 75107 (1)
D-140-30611E
SMC 903
MINICOMPUTER TECHNOLOGY
SHEET 3 of 4


-5 V SUPPLY
JUMPER IS IN ETCH FOR USE WITH LM320.
JUMPER MUST BE CUT TO USE TIPNS
D-140-30611E
SMC 903
MINICOMPUTER TECHNOLOGY
SHEET 4 OF 4

MINI COMPUTER TECHNOLOGY

SUBSIDIARY OF E-H INTERNATIONAL, INC.

696 East Trimble Road
San Jose, California 95131
(408) 942-1616 TWX 910-338-2281

125 Highgrove Street, Reading, Berks RG1 5EN
International Tel. No. 44 734 868033
National Tel No. 0734 868033