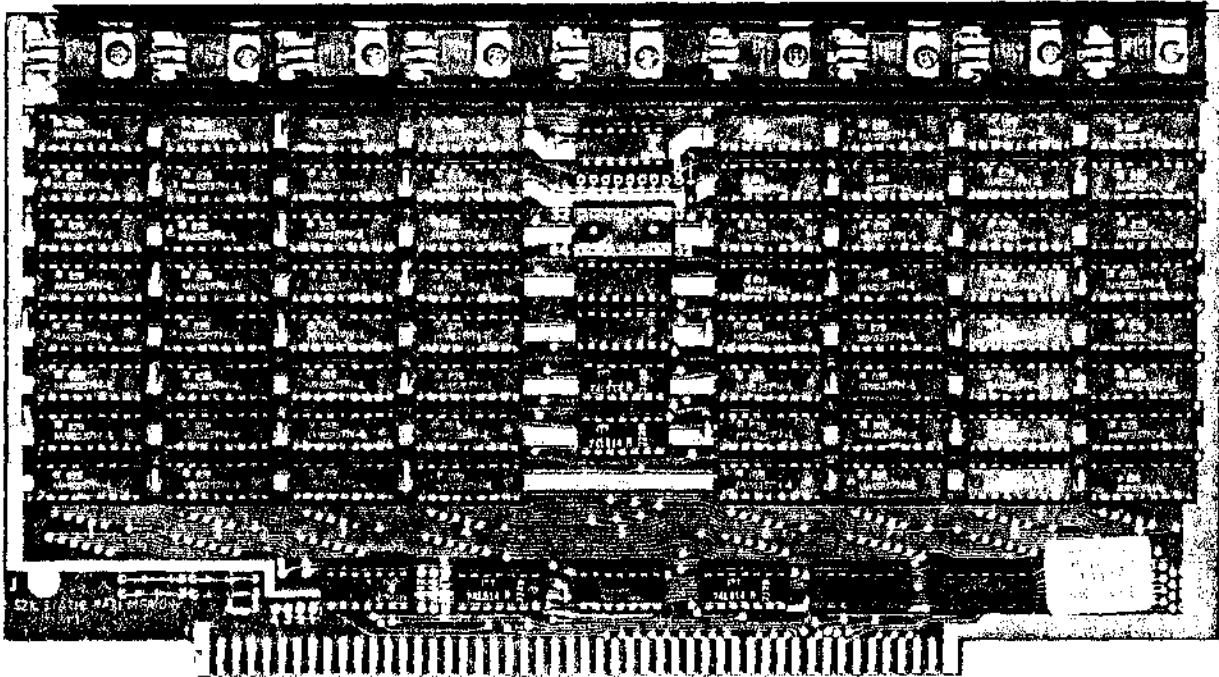


Tarbell 32K RAM Memory



- ★ S-100 BUS ★ 300ns ★ FULLY STATIC MEMORY★
- ★ ALL ADDRESS AND DATA LINES BUFFERED★
- ★ EXTENDED ADDRESSING (Bank Switching)★
- ★ EACH 4K BLOCK INDIVIDUALLY ADDRESSABLE★
- ★ PHANTOM LINE★
- ★ 20 PAGE OPERATING MANUAL★
- ★ FULL 1-YEAR WARRANTY★

Tarbell
Electronics

950 DOVLEN PLACE • SUITE B • CARSON, CALIF. 90746
(213) 538-4251 • (213) 538-2254

CONTENTS

Operation	1
Assembly Instructions	4
Parts List	8
Troubleshooting	9
Schematic Diagram	11
Appendix A (Memory Test Program)	A1

©COPYRIGHT

All rights reserved. No part of this manual schematic diagram, circuit board or descriptions contained herein may be copied or otherwise conveyed without express permission of Tarbell Electronics.

OPERATION

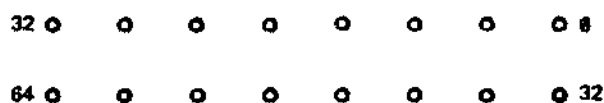
Introduction:

The overall scheme of the 32K Static Memory Board is shown on page 10. Notice data bit 0 is contained by the lowest chips on the board in 8-4K banks. Data bit 07 is at the top. If you were to use this 32K board as a 4K board, you could install only 8 chips in any vertical column. The TMS 4044 (MM5257) is a 4K by 1 memory device. One chip therefore at any location will store one bit, 4K deep.

The right hand vertical column is the one called "CS0" or CHIP SELECT - 0.

In the center top of the board you will notice some white letters saying "CS-76543210".

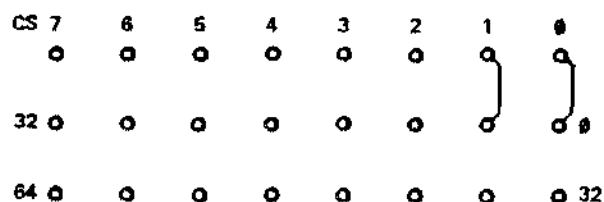
The holes directly above these numbers are connected to the CHIP SELECT lines on all the chips in the columns so marked. The columns are marked sort of backwards. CS0 on the right has a number '9' under it. CS07 on the left has a '1' under it. 'H' row is bit 0 and 'A' row is bit 07. Directly below "CS76543210" you will notice a 16 pin DIP socket. The Pattern of holes and numbers are as follows:



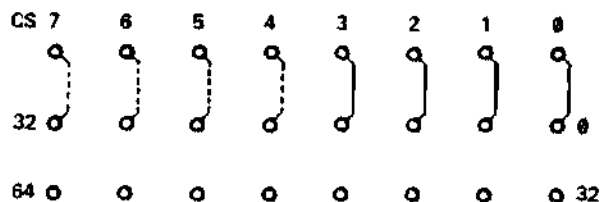
We have decoded all 16-4K blocks. The first 4K block is of course 0. The first block over 32K would be the '32' on the lower right. The last (16th) 4K block would be on the lower left.

Examples:

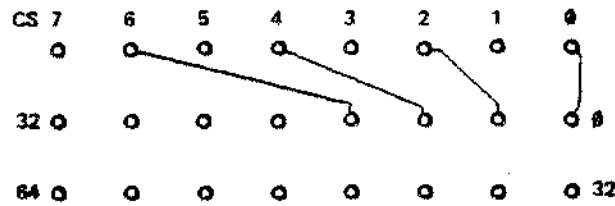
8K, Lowest 2 blocks



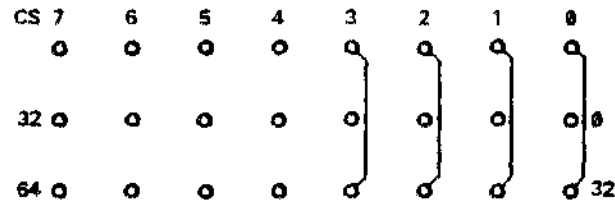
16K, Lowest 4 blocks (32K Lowest 8 blocks - Add Dotted)



16K, Lowest 4 blocks but chips in every other column

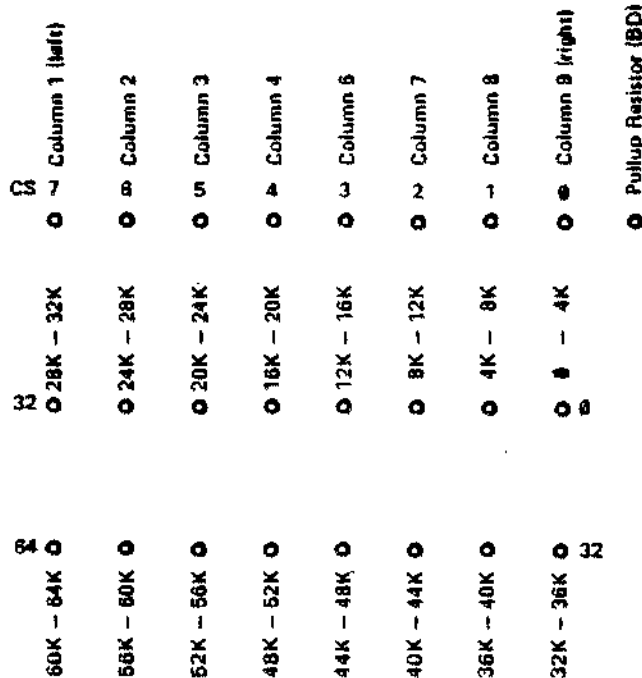


16K, From 32K to 48K (Chips in right side of board)



The placement of the chips on the board and corresponding decoded bank assignment are completely flexible. You could cross all the wires up and have any 4K section anywhere you wish. Try to think of the 32K board as 8 conventional 4K boards, all in vertical columns starting from right to left. Since there are 16-4K banks available for use in a S-100 computer, you must decide where your 8 will go.

More Examples:



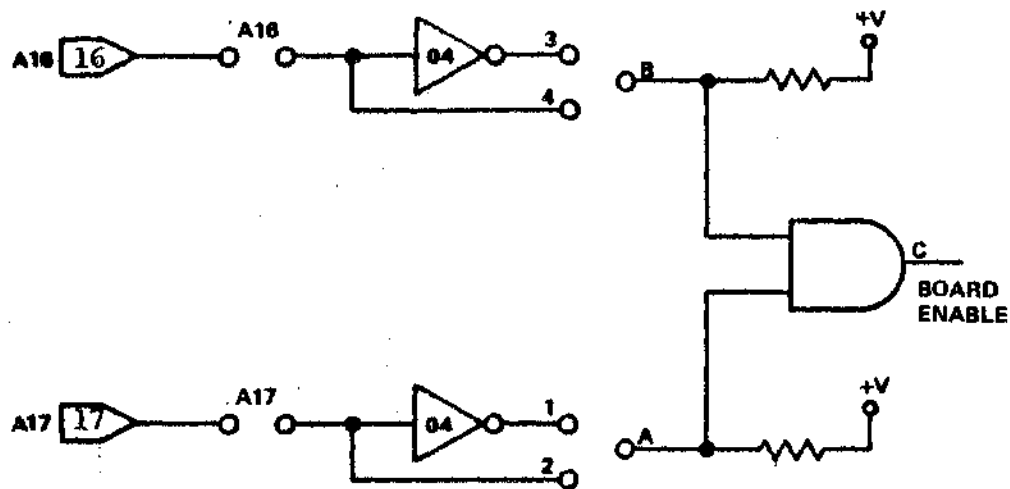
If you are interested in a straight forward static memory board without memory management or more than 64K in your system at one time, do not read further or install *any* additional jumpers. Go on to ASSEMBLY INSTRUCTIONS.

PHANTOM LINE: If jumper marker 'PHANTOM' is installed, whenever S-100 BUSS pin 67 is pulled low, the entire board will be disabled.

MEMORY MANAGEMENT: Two additional address lines are decoded. These are S-100 pins 17 (A-17) and 16 (A-16).

NOTE: IF YOU DO NOT HAVE MEMORY MANAGEMENT SKIP THIS SECTION!

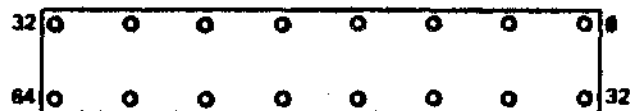
If memory management is not desired DO NOT connect anything to A-16, A-17, 1 2 3 4, or A,B. Otherwise use below to decode 64K, 128K or 256K Board Select.



LOGIC STATE		CONNECT JUMPERS	THIS BOARD IN:
A16	A17		
0	0	A to 1, B to 3	= 0K – 63K Block
1	0	A to 1, B to 4	= 64K – 127K Block
0	1	A to 2, B to 3	= 128K – 191K Block
1	1	A to 2, B to 4	= 192K – 256K Block

ASSEMBLY INSTRUCTIONS

1. () Open bag of .1 mfd capacitors and straighten leads.
2. () Refer to drawing No. 1. Install all .1 capacitors at locations shown.
3. () Turn board and solder all leads.
4. () Locate capacitors C2 in kit. **BE CAREFUL TO OBSERVE POLARITY!** Install each capacitor at locations shown on drawing No. 2 before soldering, go back and check against drawing for incorrectly polarized capacitors. Most boards returned for repair have contained errors in polarity. What happens is that a backward capacitor blows the regulator, which then zaps a whole column of memory chips! (booooo!!)
5. () Install 5 resistors, 1K value at places shown on drawing No. 1. Solder at this time.
6. () Locate all 64 18 pin sockets. Put them into the board one column at a time. Hold them in place with a piece of cardboard and flip board to back side. Solder or bend 2 corner pins of each chip (i.e. pins 8 & 16). Then go on to next column. To complete soldering go down a row after placing board on table long side up. This seems to be the fastest. After soldering entire board, inspect under strong light for solder splashes or webs.
7. () Locate special 16 pin socket that has molded circular receptacles. Install this socket at the decoded address position.

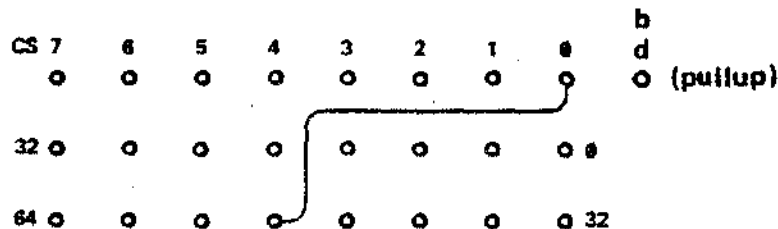


8. () Refer to drawing No. 2. Install remaining 14 and 16 pin sockets. First put a small piece of black tape over feed through holes at top center of board.
9. () Then mount heat sink in place using only outer 2 regulator screw holes. Make sure that the heat sink is positioned perfectly over all capacitors and other screw holes. Spend some time here and do this right. It's not impossible to misalign the sink and short the +5 on a capacitor lead. After setting everything up straight, carefully mount center 7 regulators in place as per board photo. Don't forget to apply a small dab of white heat sink compound to the underside of each regulator before bolting it down. After inside regulators are done, remove outside screws and do those two too. (?)

10. () Time now to start buzzing out the shorts. The most important ones are the +8 to +5 possibilities. Pin 1 of S-100 connector at lower left is +8. Make sure as you go from left to right across board that no pin 18 on any column has a low resistance path to +8. A buzz box can be used if a meter is not available. Address lines can be checked if necessary as well as data. We suggest removing all boards from your computer before plugging in the memory board. Check output from +5 regulators (the lower pin) on all regulators.
11. () Install all 16 and 14 pin chips. *Install row of round chip select pins at 'CS' as per drawing No. (1). The right most pin is a pull up resistor that may be tied to any unconnected bank of chips to insure that they do not intermittently 'select' themselves on noise.

NOTE: ON ALL 16K BOARDS, ALTERNATE UNUSED CS'S MUST BE PULLED UP!

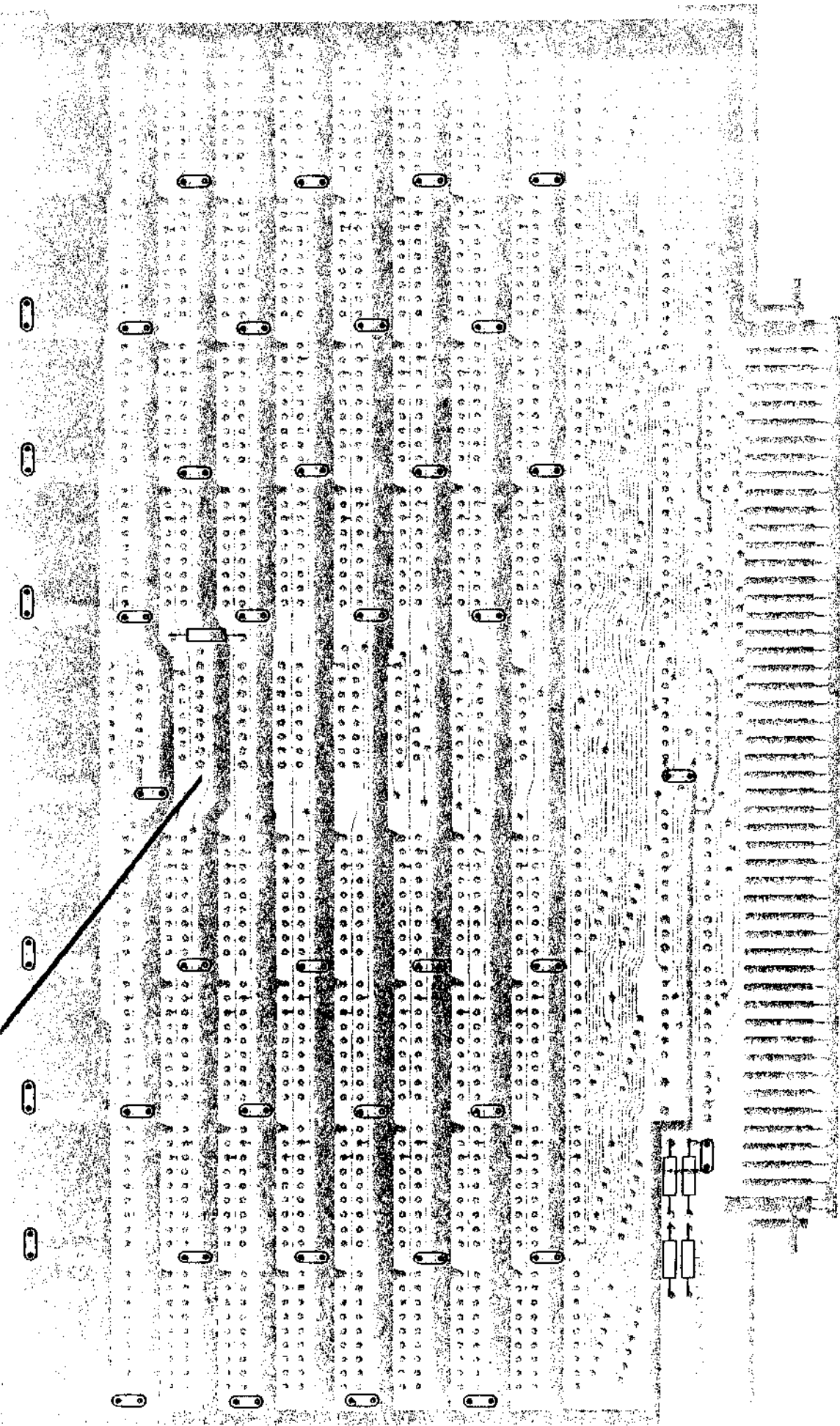
12. () Install a single column of 8 memory chips. Use the correct 'CS' for the column you have filled. They correspond to the physical location on the board of the columns except that the right most 'CS' pin is a pullup and is marked 'BD'. For example if you had filled the right most column with chips, CS line 0 would access that column. If you would like to place that 4K bank at 48K+ for the purpose of test, make a jumper from CS 0 to the fourth pin on the bank decode IC socket.



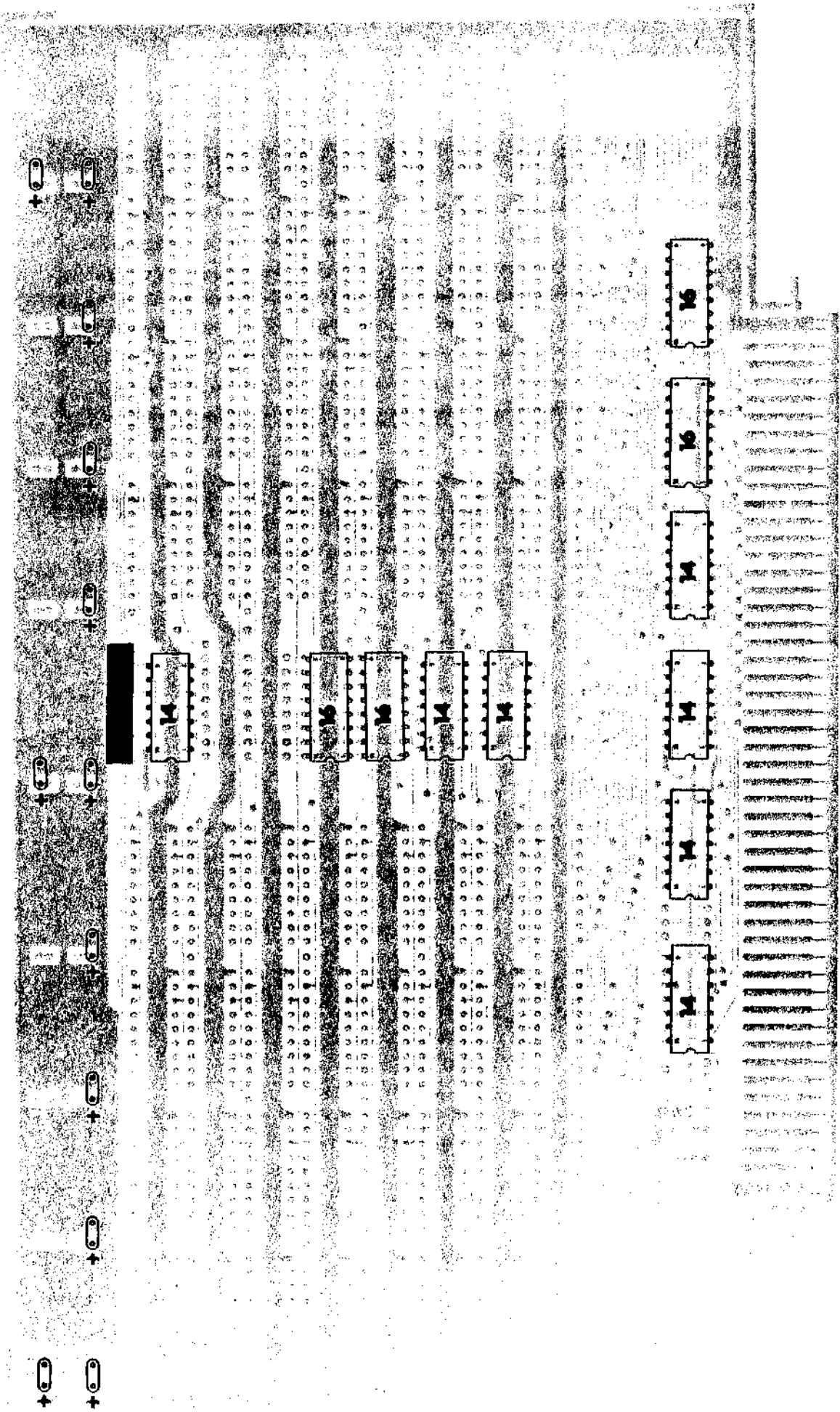
13. () Continue filling columns and testing with whatever memory test you have. We recommend that if you are planning only 16K of memory chips for a while, to use every other column to maximize cooling. A fully stuffed 32K board MUST have some forced cooling (at least 10 cfm) passing over the chip side of the board. Constantly check the board heat during the first 15 minutes of operation. An undercooled board will show data errors within the first hour of operation. These are caused by memory slowing down as it heats up. Usually no permanent damage results. Very hot boards do seem to crash chips more than cool ones though. Any reasonable cooling will allow 2 fully stuffed 32K boards to be run slot to slot. 16K (half stuffed) can be run with convection cooling. A fan never hurts though!

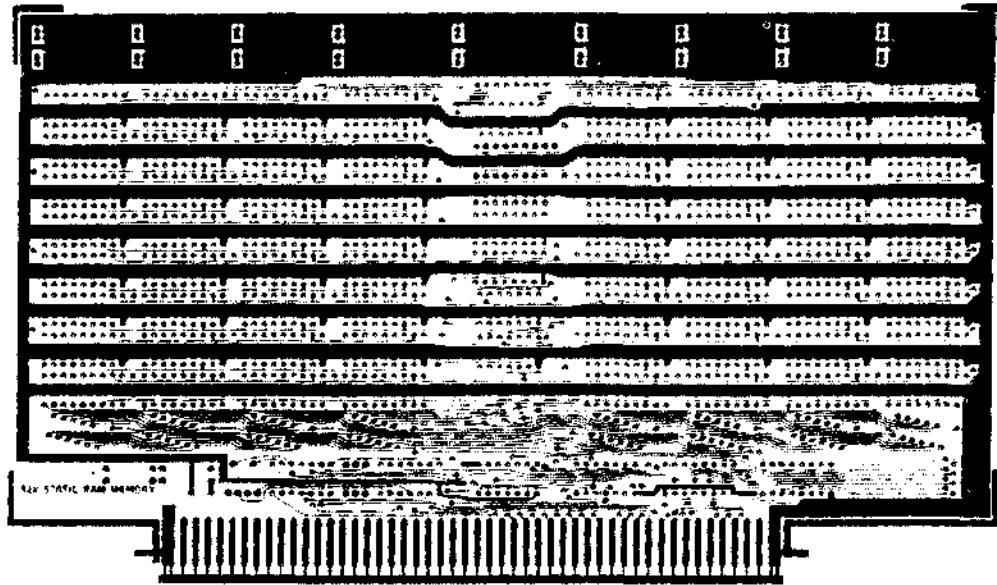
* Refer to finished board photo for IC location.

'CS' PINS



Drawing No. 1





S-100 32K Ram

PART NO.	DESCRIPTION	QTY
32KS100	CIRCUIT BOARD	1
MM5277/TMS4044	MEMORY CHIP	64
7805	REGULATOR	9
16AUGAT	GOLD 16 PIN SOCKET	1
9SOCP	JUMPER SOCKET	9
1KRES	1K RESISTOR 1/4W	5
440SCR	REGULATOR SCREW	9
440NUT	REGULATOR NUT	9
HS9	HEAT SINK	1
18PSOC	18 PIN SOCKET	64
16PSOC	16 PIN SOCKET	4
14PSOC	14 PIN SOCKET	7
475MT	4.7 - 6.8 MFD TANT.	12
.1BYCAP	.1 DIPPED CERAMIC	41
74LS04 OR 74LS14	74LS04 OR 74LS14	5
74368	74368	2
7413	7413	1
7430	7430	1
74138	74138	2

TROUBLESHOOTING

1. Most troubles occurring immediately after construction are shorts. Close visual inspection is much faster than logical circuit tracing for this type of problem.

Open circuits caused by over etching of the p.c. board or bad soldering are by far the most difficult problems to locate. Memory tests run on boards with open address lines give strange results that often do not point to the real problem. A general rule of thumb is that if the problem is random and somewhat unpredictable, an open is probably at fault. If the problem is very predictable and regular, a short or bad chip is at fault.

A 'Buzz Box' is an absolutely indispensable tool in checking memory boards. You do not have to look up from your work each time to verify continuity. The many circuit traces on a memory board are a problem by themselves without having to lose your place each time you have to look up to a meter face.

2. If the memory board has been in service for some time and a problem with it is suspected, a memory test should be run. The Rasmussen test for diseased memory is a factory test designed to display the most common problems first and the most uncommon ones last. The test never finishes by itself. The operator may terminate it by pressing the space bar at any time.

Generally memory chip failures will be found within five seconds. The test takes about three minutes to run all phases in a 32K board. The test falls into a random numbers test at the end of the first phase and will stay there until a space bar is entered.

When initializing the test for a 32K board addressed at 0000, the correct answers to the address prompt entry would be 0000 and 7FFF. The test will do the following:

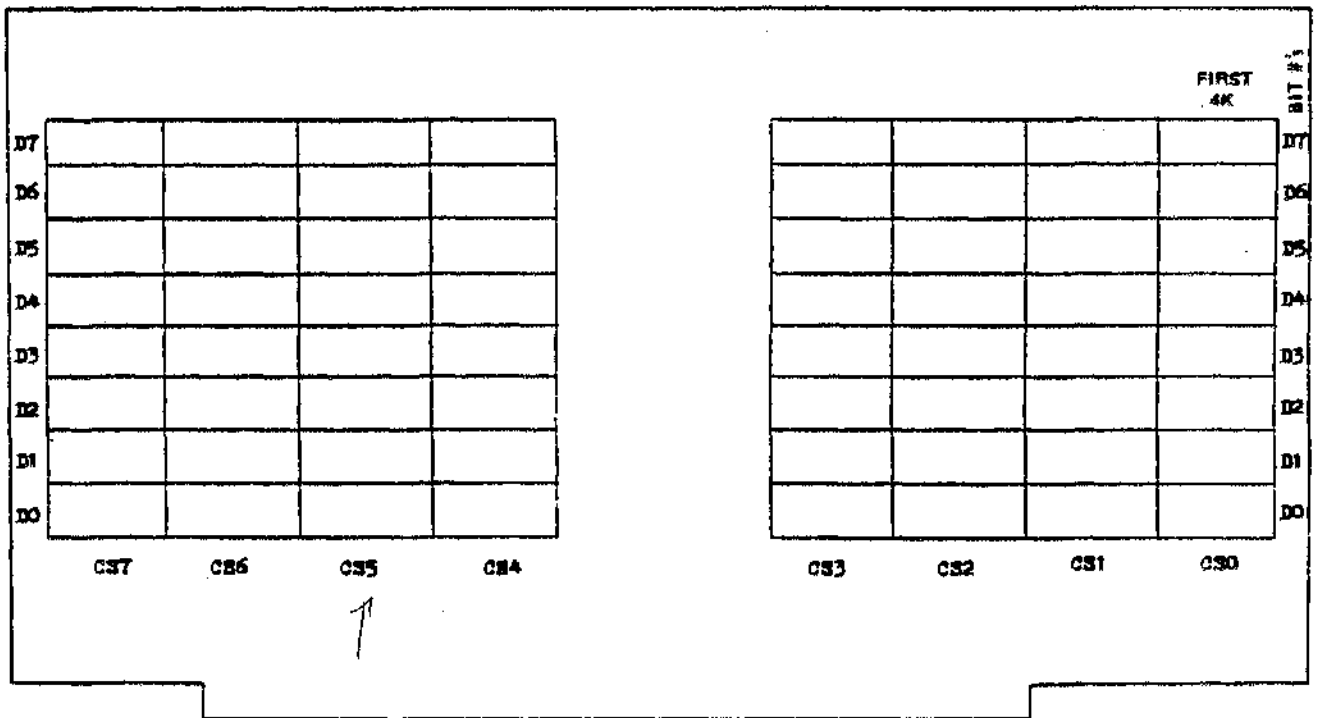
STUCK BIT:	Fills test area with FF's and checks for FF's Fills test area with 00's and checks for 00's
BIT SHORTED:	Rotates a bit from LSB to MSB filling tested memory each time, checking one bit at a time
ADDRESS SHORTED:	Fills all memory with 55 Hex then writes an AA Hex at 0000 (or the lowest address tested). It then tests the rest of memory for 55's. Then it clears location of the AA and writes it into 0001. Then 0002, 0004, 0008, etc. setting a new address bit high each time and testing all of memory. If any address bit is shorted to another, the test will find an AA in another location than the place it wrote one. This test takes the most time and is run last.

RANDOM NUMBERS: A random number routine generates an eight bit number pattern and writes it through all test memory. It then re-inserts the same seed to the routine and test reads the memory. A new seed is generated and the exercise is repeated with a new pattern. This goes on and on reporting each loop through until aborted with the space bar.

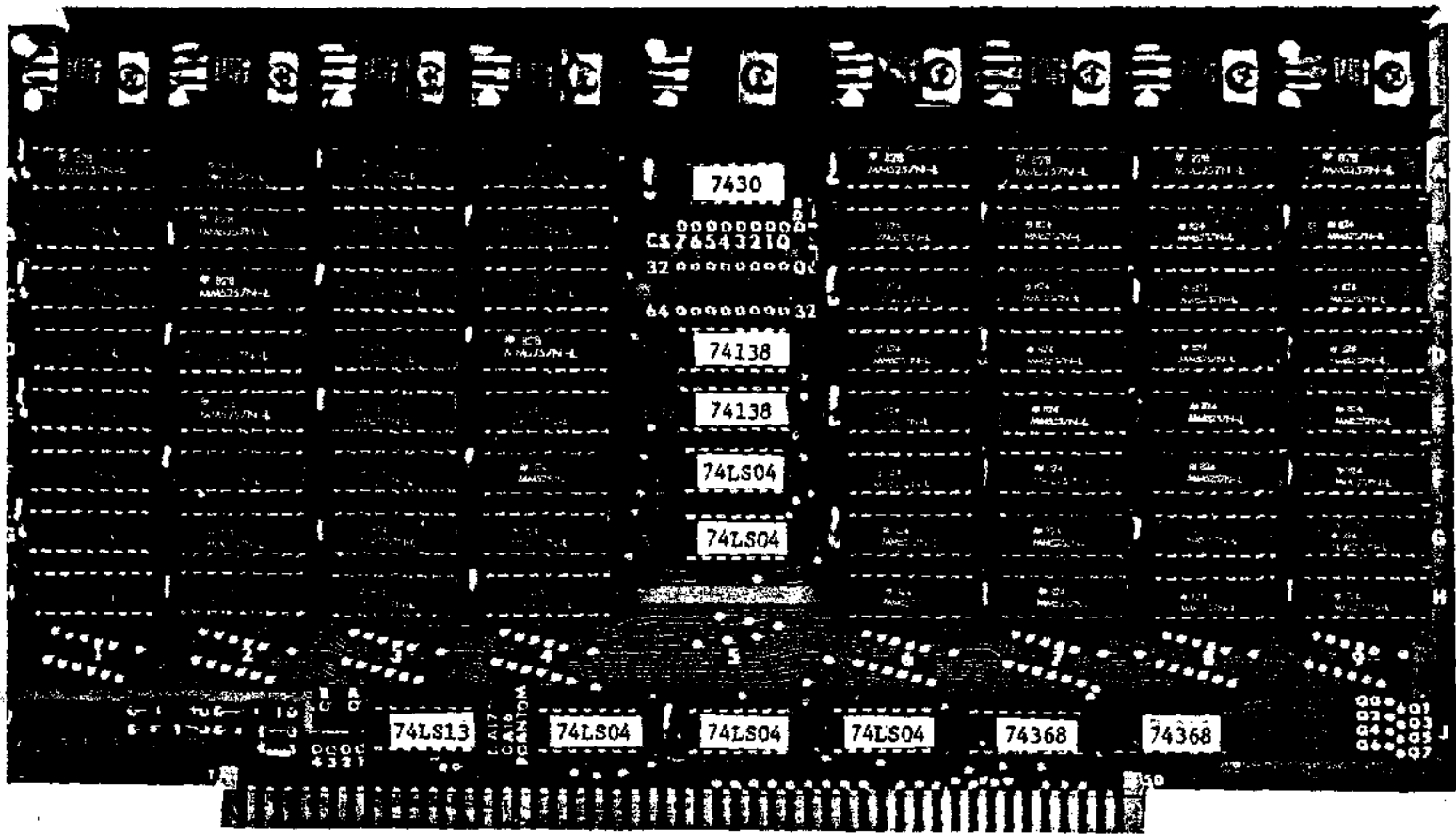
The layout drawing below will help in relating a bad bit pattern to the correct chip.

To completely test for all address open and short combinations, the test should be run over a 4K boundary, 8 times (for 32K, of course). For example the first time, enter 0000 and 0FFF as the starting and ending address. When test is complete, re-run it entering 1000 and 1FFF. Then 2000 and 2FFF and so on. Then run the test from 0000 to 7FFF. There are some subtle things that get missed if you only run the last test (0000 to 7FFF) and not 4K at a time.

If your board passes these tests and will run for an hour on the random numbers test without a problem, look somewhere else for your troubles!



Memory Chip Layout



7430

00000000
CS76543210
32 00000000
64 00000000 32

74138

74138

74LS04

74LS04

74LS13

74LS04

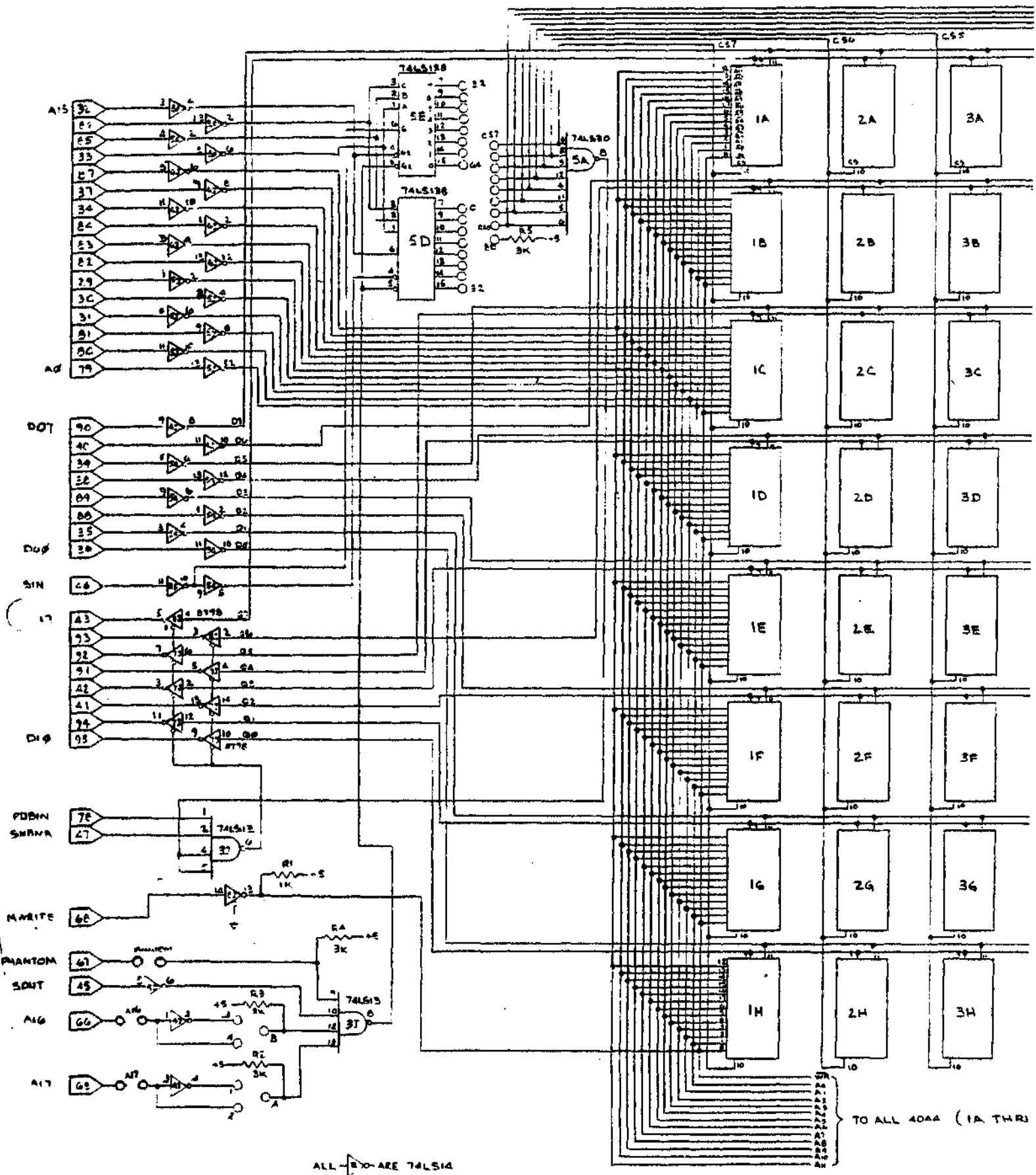
74LS04

74LS04

74368

74368

00 401
02 403
04 405
06 407



ALL 7400 ARE 74LS14

TO ALL 4044 (1A THRU 3H)

APPENDIX A
THE RASMUSSEN TEST FOR DISEASED MEMORY

```

;
;MEMORY TEST PROGRAM
;DELTA PRODUCTS
;BY D. RASMUSSEN
;LAST EDIT 1-10-79
;
;
;
;MEMORY TEST WILL USE DEFAULT ADDRESS IF STARTING AND
;ENDING ADDRESS QUESTIONS ARE ANSWERED WITH "CR"
;MEMORY TEST CAN BE TERMINATED BY "SP"
;ENTERING MEMORY TEST WITH R IN TBUF WILL CAUSE
;ALL BUT RANDOM NUMBERS TEST TO BE SKIPPED
;
;*****
;
0500 = DEFST EQU 500H ;DEFAULT STARTING ADDRESS (INCLUSIVE)
F5FF = DEFEND EQU 0F5FFH ;DEFAULT ENDING ADDRESS (INCLUSIVE)
;
;*****
;
0100 = STACK EQU 0100H ;STACK
00F9 = COUNT EQU STACK-7 ;STORAGE FOR RETRY COUNT
000E = RETRYS EQU 14 ;# OF RETRYS
00F8 = TESTWORD EQU STACK-8 ;STORAGE
00FE = ENDADD EQU STACK-2 ;ENDING ADDRESS
00FC = MEM EQU STACK-4 ;STARTING ADDRESS
00FA = TESTLOC EQU STACK-6
00F7 = TBUF EQU STACK-9 ;OR 82H FOR CPM'S TBUF
00F5 = SEED EQU STACK-11
00F3 = SEEDST EQU STACK-13
0000 = MONITOR EQU 0 ;MONITOR OR WARM BOOT ENTRY POINT
;
0100 ORG 0100H
START:
;
;
0100 C3A904 JMP START1
0103 C30900 JCONI: JMP CONIN
0106 C30600 JCONS: JMP CONST
CONO:
0109 D5 PUSH D
010A CDOC00 JCONO: CALL CONOT
010D D1 POP D
010E C9 RET
;
START2:
;
010F 3E0E MVI A,RETRYS ;SET RETRY COUNT
0111 32F900 STA COUNT
0114 119403 LXI D,SIG ;SIGN ON
0117 CDE601 CALL PMSG ;PRINT IT
011A 110104 LXI D,MSG ;STARTING ADDR. MSG.
011D CDE601 CALL PMSG ;PRINT IT

```



```

0120 CD0003      CALL GETADD      ;GET STARTING ADDR FROM CONIN
0123 D22E01      JNC LDST1       ;FLAG = 0 THEN ENTRY WAS CR
0126 210005      LXI H,DEFST     ;GET DEFAULT STARTING ADDR.
0129 E5          PUSH H
012A CDDA01      CALL PHL        ;PRINT DEFAULT ADDRESS
012D E1          POP H

LDST1:
012E 22FC0C      SHLD MEM        ;SAVE STARTING ADDRESS
0131 112304      LXI D,EMSG     ;ENDING ADDR. MSG.
0134 CDE601      CALL PMSG
0137 CD0003      CALL GETADD     ;GET ENDING ADDRESS FROM KEYBOARD
013A D24501      JNC LDST2       ;FLAG = 0 THEN WAS CR
013D 21FFF5      LXI H,DEFEND    ;DEFAULT END ADDR HIGH BYTE
0140 E5          PUSH H
0141 CDDA01      CALL PHL        ;PRINT DEFAULT ADDRESS
0144 E1          POP H

LDST2:
0145 22FE00      SHLD ENDADD    ;SAVE ENDING ADDRESS

RESTART:
0148 3AF700      LDA TBUF       ;GET TBUF TO SEE IF RANDOM TEST ONLY
014B FE52        CPI 'R'         ;IF R THEN GO DIRECTLY TO RANDOM TEST
014D CA4003      JZ RNDW        ;GO DO RANDOM NUMBERS
0150 115C04      LXI D,TEST1
0153 CDE601      CALL PMSG     ;PRINT TEST
0156 06FF        MVI B,0FFH    ;FF TEST START
0158 CDF101      CALL TESTW    ;WRITE TEST BYTE
015B CD0802      CALL TESTR    ;CHECK TEST BYTE
015E 0600        MVI B,0       ;ZEROS TEST
0160 CDF101      CALL TESTW    ;WRITE TEST BYTE 0
0163 CD0802      CALL TESTR    ;TEST
0166 116D04      LXI D,TEST3
0169 CDE601      CALL PMSG     ;PRINT BIT SHORT TEST
016C 3E01        MVI A,1       ;ROTATE BIT TEST

LOOP2:
016E 47          MOV B,A ;MAKE TEST BYTE
016F CD6602      CALL ROTST    ;DO TEST
0172 D26E01      JNC LOOP2    ;DONE WITH 8 BITS?

;
;ADDRESS LINE TEST
;
0175 118004      LXI D,TEST4
0178 CDE601      CALL PMSG     ;PRINT ADDR. LINE TEST
017B 0655        MVI B,55H    ;TEST BYTE
017D CD7202      CALL ADTEST   ;TEST
0180 06AA        MVI B,0AAH   ;TEST BYTE
0182 CD7202      CALL ADTEST   ;TEST
0185 3E52        MVI A,'R'
0187 32F700      STA TBUF     ;PUT R IN TBUF - LOOP ON RANDOM TEST
018A C34003      JMP RNDW     ;DO RANDOM # TEST

DONER:
018D 11F103      LXI D,MSGOK   ;POINT AT TEST COMPLEAT MESSAGE

DONEA:
0190 CDE601      CALL PMSG
0193 C34801      JMP RESTART  ;START TEST AGAIN WITH OLD PRAMETERS

;
;
;

```

```

ERROR:
0196 C5      PUSH B
0197 D5      PUSH D
0198 E5      PUSH H
0199 F5      PUSH PSW      ;SAVE ALL REGS.
019A 11A403  LXI D,MSG1
019D CDE601  CALL PMSG
01A0 13      INX D
01A1 3AF900  LDA COUNT
01A4 FE0E    CPI RETRYS
01A6 CCE601  CZ PMSG      ;PRINT LABLES
01A9 CDDA01  CALL PHL     ;PRINT HL REG
01AC 50      MOV D,B      ;GET SHOULD BE DATA
01AD CD4702  CALL CNVT    ;PRINT HEX
01B0 78      MOV A,B
01B1 CDOC03  CALL CNVTB
01B4 F1      POP PSW
01B5 F5      PUSH PSW
01B6 57      MOV D,A      ;GET WAS DATA
01B7 CD4702  CALL CNVT    ;PRINT HEX
01BA F1      POP PSW
01BB CDOC03  CALL CNVTB
01BE 3AF900  LDA COUNT    ;GET ERROR COUNT
01C1 3D      DCR A      ;ONE LESS
01C2 32F900  STA COUNT    ;PUT BACK COUNT
01C5 C2D601  JNZ MORET
01C8 11CB03  LXI D,MSG4
01CB CDE601  CALL PMSG
01CE CD0003  CALL GETADD  ;SEE IF STOP OR CONTINUE
01D1 3E0E    MVI A,RETRYS
01D3 32F900  STA COUNT    ;RESET COUNT

MORET:
01D6 E1      POP H      ;GET ALL REGS. BACK AND CONTINUE TEST
01D7 D1      POP D
01D8 C1      POP B
01D9 C9      RET

;
;PRINT H,L REGISTER
;
PHL:
01DA 54      MOV D,H
01DB CD4702  CALL CNVT    ;MAKE ADDRESS OF ERROR ASCII
01DE 55      MOV D,L
01DF CD4702  CALL CNVT
01E2 CD2E03  CALL TAB
01E5 C9      RET

;
;PRINT STRING POINTED TO BY D,E
;
PMSG:
01E6 1A      LDAX D      ;GET BYTE TO BE PRINTED
01E7 B7      ORA A      ;END OF TEXT?
01E8 C8      RZ
01E9 4F      MOV C,A    ;CONVENTION
01EA CD0901  CALL CONO   ;GOTO OUTPUT
01ED 13      INX D
01EE C3E601  JMP PMSG

```

```

;
;FILL TEST MEMORY WITH TEST WORD
;
TESTW:
01F1 CD0002      CALL LOAD
WLOOP:
01F4 CD3902      CALL ADCK
01F7 D8          RC
01F8 70          MOV M,B          ;PUT TEST WORD IN MEMORY
01F9 CD4002      CALL INXH
01FC D8          RC
01FD C3F401      JMP WLOOP
LOAD:
0200 2AFE00      LHLD ENDADD      ;GET ENDING ADDRESS
0203 EB          XCHG          ;PUT ENDING ADDRESS IN DE
0204 2AFC00      LHLD MEM          ;GET STARTING ADDRESS
0207 C9          RET
;
;READ AND CHECK TEST MEMORY FOR TEST WORD
;
TESTR:
0208 2AFC00      LHLD MEM
020B 22FA00      SHLD TESTLOC
TESTADR:
020E CD2102      CALL CSTAT
RLOOP:
0211 CD3902      CALL ADCK
0214 D8          RC
0215 7E          MOV A,M
0216 B8          CMP B          ;CHECK SHOULD BE
0217 C49601      CNZ ERROR
021A CD4002      CALL INXH
021D D8          RC
021E C31102      JMP RLOOP
;
;QUERY CONSOLE FOR STATUS
;
CSTAT:
0221 E5          PUSH H
0222 C5          PUSH B
0223 CD0601      CALL JCONS      ;SEE IF KEY STRUCK
0226 B7          ORA A          ;SET FLAGS
0227 CA3202      JZ NOKEY
022A CD0301      CALL JCONI      ;GET KEY
022D FE20        CPI ' '          ;EXIT IF SPACE
022F CA8E03      JZ RETURN
NOKEY:
0232 C1          POP B
0233 2AFE00      LHLD ENDADD      ;GET STARTING ADDRESS
0236 EB          XCHG
0237 E1          POP H
0238 C9          RET
;
;ADDRESS LIMIT CHECK
;
ADCK:
0239 7A          MOV A,D

```

```

023A 94          SUB H
023B D8          RC
023C C0          RNZ
023D 7B          MOV A,E
023E 95          SUB L
023F C9          RET

INXH:
0240 C5          PUSH B
0241 010100     LXI B,1
0244 09          DAD B
0245 C1          POP B
0246 C9          RET

;
;CONVERT REG. D TO HEX ASCII
;
CNVT:
0247 7A          MOV A,D
0248 0F          RRC          ;GET HIGH NIBBLE
0249 0F          RRC
024A 0F          RRC
024B 0F          RRC
024C E60F       ANI 0FH
024E CD5802     CALL CNV          ;MAKE NIBBLE ASCII
0251 3E0F       MVI A,0FH
0253 A2          ANA D          ;GET LOW NIBBLE
0254 CD5802     CALL CNV          ;MAKE ASCII NIBBLE
0257 C9          RET

CNV:
0258 C630       ADI 30H
025A FE3A       CPI 3AH
025C FA6102     JM DONE
025F C607       ADI 7

DONE:
0261 4F          MOV C,A
0262 CD0901     CALL CONO
0265 C9          RET

;
ROTTST:
0266 F5          PUSH PSW
0267 CDF101     CALL TESTW
026A F1          POP PSW
026B F5          PUSH PSW
026C CD0802     CALL TESTR
026F F1          POP PSW
0270 07          RLC
0271 C9          RET

ADTEST:
0272 CDF101     CALL TESTW          ;FILL MEMORY WITH TEST WORD
0275 78          MOV A,B
0276 32F800     STA TESTWORD      ;SAVE TEST WORD
0279 CD0002     CALL LOAD
027C 2F          CMA
027D 77          MOV M,A
027E 22FA00     SHLD TESTLOC
0281 23          INX H
0282 CD0E02     CALL TESTADR
0285 010100     LXI B,1          ;START ADDR. LSB ON

```

```

LOOP4:
0288 2AFC00      LHLD MEM
028B 09          DAD B
028C D8          RC
028D 7A          MOV A,D          ;ROLLED OVER 65K
028E 94          SUB H           ;GET TEST ADDR
028F D8          RC
0290 C29602     JNZ INXHA      ;PAST END?
0293 7B          MOV A,E
0294 95          SUB L
0295 D8          RC
0296 3AF800     INXHA:      LDA TESTWORD
0299 C5          PUSH B
029A 47          MOV B,A
029B 2F          CMA
029C 77          MOV M,A      ;INVERT BITS
029D 22FA00     SHLD TESTLOC ;WRITE TO TEST MEMORY
02A0 CD4002     CALL INXH
02A3 DAB102     JC OVERFL
02A6 CD0E02     CALL TESTADR ;TEST
02A9 E1          POP H
02AA 29          DAD H
02AB D8          RC
02AC 44          MOV B,H
02AD 4D          MOV C,L
02AE C38802     JMP LOOP4
02B1 C1          OVERFL:  POP B
02B2 C9          RET
;
GETBYT:
02B3 CDD702     CALL CHAR
02B6 C8          RZ
02B7 FE20       CPI ' '          ;EXIT IF SPACE
02B9 CA8E03     JZ RETURN
02BC CDE502     CALL ATOH
02BF F2F902     JP REENTER
02C2 0F          RRC
02C3 0F          RRC
02C4 0F          RRC
02C5 0F          RRC
02C6 5F          MOV E,A
02C7 D5          PUSH D
02C8 CDD702     CALL CHAR
02CB D1          POP D
02CC CAF902     JZ REENTER
02CF CDE502     CALL ATOH
02D2 F2F902     JP REENTER
02D5 B3          ORA E
02D6 C9          RET
CHAR:
02D7 CD0301     CALL JCONI
02DA FE0D       CPI 0DH
02DC 37          STC
02DD C8          RZ
02DE F5          PUSH PSW
;SET CARRY FLAG (MAYBE CR)

```

```

02DF 4F          MOV C,A
02E0 CD0901     CALL CONO
02E3 F1         POP PSW
02E4 C9         RET

      ATOH:
02E5 D630      SUI 30H
02E7 FAF702    JM BAD
02EA FEOA      CPI OAH
02EC F8        RM
02ED D607      SUI 7
02EF FEOA      CPI OAH
02F1 FAF702    JM BAD
02F4 FE10      CPI 10H
02F6 F8        RM

      BAD:
02F7 AF        XRA A          ;SET ZERO FLAG
02F8 C9        RET

      REENTER:
02F9 E1        POP H          ;FIX STACK
02FA 114504    LXI D,IEMSG
02FD CDE601    CALL PMSG

;
;INPUT 4 HEX VALUES FROM CONSOLE
;
      GETADD:
0300 CDB302    CALL GETBYT
0303 D8        RC
0304 67        MOV H,A
0305 E5        PUSH H
0306 CDB302    CALL GETBYT
0309 E1        POP H
030A 6F        MOV L,A
030B C9        RET

;
;PRINT REG-A IN BINARY
;
      CNVTB:
030C 5F        MOV E,A          ;SAVE A
030D 1602      MVI D,2          ;TWO SPACES
030F CD3003    CALL TLOP
0312 0602      MVI B,2          ;NIBBLE COUNT

      PNIBB:
0314 1604      MVI D,4

      ALLBIT:
0316 7B        MOV A,E          ;RESTORE A
0317 17        RAL
0318 5F        MOV E,A
0319 0E30      MVI C,30H        ;0
031B D22003    JNC ZERO
031E 0E31      MVI C,31H        ;1

      ZERO:
0320 CD0901    CALL CONO
0323 15        DCR D
0324 C21603    JNZ ALLBIT
0327 CD3803    CALL SPACE          ;PRINT SPACE
032A 05        DCR B
032B C21403    JNZ PNIBB

```

```

TAB:
032E 1604      MVI D,4
TLOP:
0330 CD3803    CALL SPACE
0333 15        DCR D
0334 C23003    JNZ TLOP
0337 C9        RET
SPACE:
0338 0E20      MVI C,' '
033A C5        PUSH B
033B CD0901    CALL CONO
033E C1        POP B
033F C9        RET
;
;RANDOM NUMBER TEST
;
;WRITE RANDOM NUMBERS
RNDW:
0340 119404    LXI D,RMSG
0343 CDE601    CALL PMSG
0346 2AF500    LHLD SEED      ;GET OLD SEED
0349 22F300    SHLD SEEDST   ;SAVE IT FOR LATER
034C 22FA00    SHLD TESTLOC
034F CD0002    CALL LOAD
RNDWL:
0352 CD7E03    CALL RND
0355 70        MOV M,B
0356 CD3902    CALL ADCK
0359 CA6003    JZ RNRD
035C 23        INX H
035D C35203    JMP RNDWL
;
;READ AND CHECK RANDOM NUMBERS
RNRD:
0360 CD2102    CALL CSTAT
0363 2AF300    LHLD SEEDST   ;START WITH SAME SEED
0366 22F500    SHLD SEED     ;PASS SEED TO GENERATOR
0369 CD0002    CALL LOAD     ;GET TEST ADDRESS
RNRDL:
036C CD7E03    CALL RND
036F 7E        MOV A,M
0370 B8        CMP B
0371 C49601    CNZ ERROR
0374 CD3902    CALL ADCK
0377 CA8D01    JZ DONER
037A 23        INX H
037B C36C03    JMP RNRDL
;GENERATE RANDOM NUMBER FORM SEED
;
RND:
037E E5        PUSH H
037F 2AF500    LHLD SEED
0382 7D        MOV A,L
0383 AC        XRA H
0384 47        MOV B,A
0385 07        RLC
0386 6F        MOV L,A

```

```

0387 84          ADD H
0388 67          MOV H,A
0389 22F500     SHLD SEED
038C E1         POP H
038D C9         RET

                RETURN:
038E 31F100     LXI SP,STACK-15 ;FIX STACK
0391 C30000     JMP MONITOR

;
;MESSAGES
;
0394 0A0D4D454DSIG: DB OAH,ODH,'MEMORY TEST',ODH,OAH,0
03A4 0A0D004C4FMSG1: DB OAH,ODH,0,'LOCATION'
03AF 202053484F  DB ' SHOULD BE          WAS'
03C8 0A0D00     DB OAH,ODH,0
03CB 0A0D224352MSG4: DB OAH,ODH, '"CR" TO CONTINUE TEST "SP" TO STOP ',0
03F1 0A0D544553MSGOK: DB OAH,ODH,'TEST COMPLETE',0
0401 0A0D535441SMSG: DB OAH,ODH,'STARTING ADDRESS (HEX OR "CR") ',0
0423 0A0D454E44EMSG: DB OAH,ODH,'ENDING ADDRESS (HEX OR "CR") ',0
0445 0A0D494E50IEMSG: DB OAH,ODH,'INPUT ERROR RETYPE- ',0
045C 0A0D424954TEST1: DB OAH,ODH,'BIT STUCK TEST',0
046D 0A0D53484FTEST3: DB OAH,ODH,'SHORTED BIT TEST',0
0480 0A0D414444TEST4: DB OAH,ODH,'ADDRESS LINE TEST',0
0494 0A0D52414ERMSG: DB OAH,ODH,'RANDOM NUMBER TEST',0

;
;*****
;
;INSERT CUSTOM I/O ROUTINES HERE
START1:
;AND ANY INITIALIZATION ROUTINES HERE
;
04A9 31F100     LXI SP,STACK-15 ;SET STACK
;
;*****
;THESE ROUTINES ARE FOR CPM USERS
0006 =          CONST EQU 06H ;CONSOLE STATUS TO REG-A
;                ;A=FF,CHAR. READY A=0,CHAR. NOT READY
0009 =          CONIN EQU 09H ;CONSOLE CHARACTER TO REG-A
;
000C =          CONOT EQU 0CH ;CHARACTED FROM REG-C TO CONSOLE
;
04AC 3A020C     LDA 2 ;GET BASE OF JUMP TABLE
04AF 320801     STA JCONS+2
04B2 320501     STA JCONI+2
04B5 320C01     STA JCONO+2
;*****
04B8 C30F01     JMP START2
04BB           END START

```