

0001 .MAIN

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

;
;
;

RCSL: 44-RT1623
AUTHOR: JJO/HH
EDITED: 77.10.24

;

RC3600 CHECKERBOARD IV

;
;
;

BINARY CARDS : 44-RT1626
BINARY TAPE : 44-RT1625
ASCII TAPE : 44-RT1624

;

KEYWORDS: RC3600, CPU 705, MEMORY-TESTPROGRAM.

0002 .MAIN

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```
*****  
;  
;  
;  
; DESCRIPTION: RC3600 CHECKERBOARD IV  
;  
;  
; REVISION HISTORY:  
;  
; REV. DATE  
;  
; 00 75.04.01  
; 01 77.03.10  
; 02 77.10.24  
;  
;  
; COPYRIGHT (C) A/S REGNOCENTRALEN, 1975.  
; ALL RIGHTS RESERVED.  
*****
```

1 0003 .MAIN

```
01 ;
02 ;
03 ;1, ABSTRACT
04 ;
05 ; CHECKERBOARD IV IS A MAINTENCE PROGRAM DESIGNED
06 ; TO PRODUCE WORST CASE NOISE CONDITION ON THE
07 ; SENSE/INHIBIT WIRES. THE PROGRAM SHOULD BE RUN
08 ; TO INSURE PROPER OPERATION OF SENSE AMPS,INHIBIT
09 ; DRIVERS, AND MEMORY CURRENTS.
10 ;
11 ;2, MACHINE REQUIREMENTS
12 ;2.1 ANY RC3600: PROCESSOR
13 ; 4K READ/WRITE MEMORY
14 ;
15 ;
16 ;3, SWITCH SETTINGS
17 ;3.1 STARTING ADDRESS =000002
18 ;3.2 SWITCH 0(1) =1024 READ/WRITE DI-
19 ; STURB
20 ;3.3 SWITCH 15(1) =INHIBIT HALT ON ER-
21 ; ROR
22 ; 15(0) =ENABLE HALT ON ER-
23 ; ROR
24 ;
25 ;
26 ;4, OPERATING PROCEDURE
27 ;4.1 LOAD THE PROGRAM VIA THE BINARY LOADER
28 ;4.2 SET SWITCHES TO 000002
29 ;4.3 PRESS START
30 ;4.3.1 THE PROGRAM WILL PRINT THE HIGHEST
31 ; LOCATION THE PATTERN IS TO USE
32 ;4.4 IF THE FAILURES ARE MARGINAL, SETTING SWITCH
33 ; 0 MAY AID IN INDUCING A FAILURE TO OCCURE,
34 ;4.5 WHEN SCOPING OR ADJUSTING CURRENT, SETTING
35 ; SWITCH 15 WILL ENABLE THE ERROR HALT, THE
36 ; BELL WILL STILL BE RUNG.
37 ;4.6 PROGRAM MODIFICATIONS
38 ;4.6.1 C(3)=ADR THE STARTING PATTERN
39 ; ADDRESS
40 ;4.6.2 C(5)=INHIBIT INHIBIT THE CHECKER-
41 ; BOARD PATTERN ON
42 ; CLEARED BITS.
43 ;
44 ;
45 ;
46 ;5, PROGRAM OUTPUT/ERROR DISCRPTION
47 ;5.1 AT EACH OCCURANCE OF ERROR,THE TELETYPE-BELL
48 ; WILL BE RUNG, AND IF SWITCH 15 (0)
49 ; THE PROGRAM WILL HALT AT LOCATION "ER".
50 ;5.2 WHEN A ERROR HALT OCCURES:
51 ; AC0 =CORRECT MEMORY PATTERN
52 ; AC1 = THE ERROR WORD
53 ; AC2 =THE ERROR ADDRESS
54 ; AC3 =ADDRESS OF PROGRAM
55 ;5.3 SET SWITCH (15) IF SCOPING, PRESS CONTINUE.
56 ;5.4 SYNC PULSES
57 ;
58 ; A "P" PULSE (A74) IN STORE CYCLE.
59 ;
```

0004 .MAIN

```
01
02      ;6,   PROGRAM DISCRIPTION
03      ;6,0   TEST INTERRUPT FROM T10
04      ;6,1   RELOCATE PROGRAM FROM "BEGIN" TO "CEND" TO A RAN-
05      ;       DOMLY GENERATED PLACE IN CORE.
06      ;6,2   GO TO RANDOMLY GENERATED ADDRESS OF "BEGIN".
07      ;6,3   STORE 32 PATTERNWORDS IN TABLE "TABS". THE FI-
08      ;       GURES IS 16 FLOATING ONES, FOLLOWED BY 16 FLOA-
09      ;       TING ZEROS.
10      ;6,4   FILL THE CORE FROM C(ADR) TO "BEGIN" AND FROM
11      ;       "CEND" TO MEMTOP WITH COPY'S OF THE PATTERN-
12      ;       TABLE.
13      ;6,5   DISTURB MEMORY IF SWITCH 0(1).
14      ;6,6   COMPARE THE MEMORY PATTERNS IN CORE WITH THE FI-
15      ;       GURE IN "TABS".
16      ;6,7   CHANGE PATTERN FIGURE IN "TABS".
17      ;6,8   EXECUTE STEP 6.4 TO 6.7 32 TIMES.
18      ;6,9   GOTO 6.0.
19
20      ;7,   PATTERN DESCRIPTION
21
22      ;       THE PROGRAM SEQUENTIALLY USES:
23      ;       1)   A SLIDING FLOATING ONE BIT
24      ;       2)   A SLIDING FLOATING ZERO BIT
25
26      ;8,   LIMITATIONS
27      ;       NONE
28
29
```

1 0005 .MAIN

01
02 000002 .LOC 2
03 000002 000177 JMP MSIZ

04
05 000003 001000 ADR: 1000
06 000004 007577 FINAL: 7577
07 000005 177777 INH: -1
08 000006 000001 PATT: 1
09 000007 000000 ERET: 0
10 000100 000101 PACOU: 101
11 000111 000361 IMPAS: MEPAS
12 000112 000000 MESCO: 0
13 000113 000200 PACON: 128,

;PATTERN STARTING ADDRESS
;PATTERN FINAL ADDRESS=CHANGED BY PROG.
;MASK FOR INHIBITED BITS

14
15 000040 .LOC 40
16 000040 000101 C101: 101
17 000041 000400 C400: 400
18 000042 000077 C77: 77
19 000043 007777 C7777: 7777
20 000044 000207 C207: 207
21 000045 000060 C60: 60
22 000046 000377 C377: 377
23 000047 177400 C1774: 177400
24 000050 177773 M5: -5
25 000051 070000 C070000:070000
26 000052 000000 MODUAL: 0
27 000053 000000 EDIST: 0
28 000054 001000 K1000: 1000
29 000055 177577 M201: -201
30 000056 000250 CMA: MMA2
31 000057 177402 PSIZE: BEGIN-CEND
32 000060 000000 PLUC: 0
33 000061 000000 BPROG: 0
34 000062 000000 EPROG: 0
35 000063 012345 RANDOM: 12345
36 000064 000020 C20: 20
37 000065 177740 M40: -40
38 000066 000040 C40: 40
39 000067 010000 C10K: 10000
40 000070 000000 WHICH: 0
41 000071 000020 CX: 20
42 000072 000400 CY: 400
43 000073 177777 CMSK: -1
44 000074 000000 FLAG: 0
45 000075 000000 IMES: 0
46 000076 000000 MKTK: 0
47 000077 000736 TABL: TABS
48 001000 000735 TABL0: TABS-1
49 001001 000000 BEGL: 0
50 001002 000400 CBEG: BEGIN
51 001003 000000 RETURN: 0
52

;CHANGED TO -2 BY PROG AFTER 1. PASS

1 0006 .MAIN

```
01
02 00104 004151 SEND: JSK .RAND ;GET A RANDOM # TO AC0
03 00105 000063 RANDOM
04 00106 024004 LDA 1,FINAL ;AC1:= MEMTOP - 200 = TOP
05 00107 030057 LDA 2,PSIZE ;AC2:= - SIZE
06 00110 147000 ADD 2,1 ;AC1:=TOP-SIZE=LAST PAIT, TABLE ADDRESS=
07 ;LASTP
08 00111 101220 MOVZR 0,0 ;MAKE # POS. BY RIGHT SHIFT
09 00112 122422 SUBZ 1,0,SZC ;SKIP IF LASTP>#, AC0:= #-LASTP
10 00113 000112 JMP , -1
11 00114 123000 ADD 1,0 ;AC0=# MODULO C(FINAL) = -N*(LASTP)
12 00115 024065 LDA 1,M40 ;AC1:= 177740
13 00116 123400 AND 1,0
14 00117 024003 LDA 1,ADR
15 00120 122433 SUBZ# 1,0,SNC ;IF # TOO SMALL USE, SKIP IF C(ADR)<#
16 00121 121000 MOV 1,0 ;C(ADR) FOR STARTER
17 00122 040060 STA 0,PLOC ;AC0:= STARTADR
18 00123 040101 STA 0,BEGL
19 00124 010101 ISZ BEGL
20 00125 010101 ISZ BEGL ;BEGL:= STARTADR + 2
21 00126 034065 LDA 3,M40
22 00127 117000 ADD 0,3
23 00130 054061 STA 3,BPROG ;BPROG:= STARTADR - 40
24 00131 142400 SUB 2,0 ;AC0:= STARTADR + SIZE
25 00132 024065 LDA 1,M40
26 00133 122400 SUB 1,0
27 00134 123400 AND 1,0
28 00135 040062 STA 0,EPROG ;EPROG:= STARTADR +SIZE-40
29 00136 145000 MOVE: MOV 2,1
30 00137 030102 LDA 2,CHEG ;AC2:= ADDRESS OF "BEGIN"
31 00140 021000 LDA 0,0,2 ;MOVE A COPY OF
32 00141 041440 STA 0,40,3 ;CHECKERBOARD TO
33 00142 175400 INC 3,3 ;SELECTED SPOT,
34 00143 151400 INC 2,2
35 00144 125404 INC 1,1,SZR ;TEST FOR LAST REG
36 00145 000140 JMP MOVE+2 ;TO BE MOVED,
37 00146 034060 LDA 3,PLOC
38 00147 005401 JSK 1,3 ;EXIT TO RELOCATED PROG.
39 00150 000234 JMP MMA ;RETURN TO HERE FROM LAST INSTR.
40 ;IN UPDT
41
```

```
42 00151 054173 .RAND: STA 3,.UD03 ;GENERATE A RANDOM #
43 00152 010173 ISZ .UD03
44 00153 031400 LDA 2,0,3 ;AC2:= RANDOM
45 00154 021000 LDA 0,0,2 ;AC0:= 12345
46 00155 024176 LDA 1,.UD21
47 00156 044174 STA 1,.UD10
48 00157 105120 MOVZL 0,1
49 00160 125120 MOVZL 1,1
50 00161 014174 DSZ .UD10
51 00162 000160 JMP , -2
52 00163 107000 ADD 0,1
53 00164 125120 MOVZL 1,1
54 00165 125120 MOVZL 1,1
55 00166 123000 ADD 1,0
56 00167 034175 LDA 3,.UD20
57 00170 163000 ADD 3,0 ;AC0:= #
58 00171 041000 STA 0,0,2 ;LOC,(CALL+1):= #
59 00172 002173 JMP 0,.UD03 ;RETURN
60
```

```
61 00173 000000 .UD03: 0
62 00174 000000 .UD10: 0
63 00175 033031 .UD20: 33031
64 00176 000010 .UD21: 10
```

1 0007 ,MAIN

```
01
02 00177 020054 MSIZ: LDA 0,K1000 ;SIZE THE MEMORY
03 00200 115000 MOV 0,3
04 00201 031400 MSIZ1: LDA 2,0,3 ;SAVE C(MEM)
05 00202 055400 STA 3,0,3
06 00203 025400 LDA 1,0,3
07 00204 051400 STA 2,0,3 ;RESTORE MEMORY
08
09 00205 125014 MOV# 1,1,SZR
10 00206 124015 COM# 1,1,SNR
11 00207 000215 JMP MSIZ2 ;END OF MEMORY
12 00210 136414 SUB# 1,3,SZR ;AC1=BAD, AC3=GOOD
13 00211 063077 HALT ;MEMORY FAILED,
14 00212 117000 ADD 0,3
15 00213 175113 MOVL# 3,3,SNC ;INCREMENT MEMORY ADDRESS,
16 00214 000201 JMP MSIZ1 ;TEST FOR 32K
17
18 00215 020055 MSIZ2: LDA 0,M201
19 00216 163000 ADD 3,0
20 00217 040004 STA 0,FINAL ;ADJUST FINAL TO END OF CUR, MEM
21 00220 126440 SUBO 1,1
22 00221 044012 STA 1,MESCO ;INITIALIZE COUNTERS
23 00222 044070 STA 1,WHICH
24 00223 024013 LDA 1,PACON
25 00224 044010 STA 1,PACOU
26 00225 126000 ADC 1,1
27 00226 065135 DOAS 1,35
28 00227 065111 DOAS 1,TTO
29 00230 004262 JSR MESS ;MESS CURRENT MEM.SIZE ON TTY
30 00231 000335 MFSIZE
31 00232 024004 LDA 1,FINAL
32 00233 004300 JSR POCT
33
34 00234 063511 MMA: SKPBZ TTO
35 00235 000234 JMP .-1 ;WAIT FOR TTO DONE,
36 00236 063535 SKPBZ 35
37 00237 000236 JMP .-1 ;WAIT FOR DISP DONE
38 00240 020056 LDA 0,CMA
39 00241 040001 STA 0,1 ;SET INTERRUPT RETURN
40 00242 152520 SUBZL 2,2 ;THIS PROGRAM TEST FOR INTERRUPT
41 00243 102000 MMA1: ADC 0,0 ;ABILITY TO CLEAR MA,
42 00244 040000 STA 0,0
43 00245 025000 LDA 1,0,2
44 00246 060177 NIOS CPU ;ENABLE INTERRUPT
45 00247 005000 JSR 0,2 ;SET BIT INTO MA, INTERRUPT WILL OCCUR
46 ;JUST AFTER JUMP
47
48
49 00250 045000 MMA2: STA 1,0,2
50 00251 020000 LDA 0,0
51 00252 112414 SUB# 0,2,SZR ;AC0=PC STORED
52 00253 000260 JMP .+5 ;AC2=CORRECT, MEMORY FAILED!
53 00254 151120 MOVZL 2,2 ;OK, AC0:= AC2
54 00255 151113 MOVL# 2,2,SNC
55 00256 000243 JMP MMA1 ;TEST INTERRUPT AGAIN
56 00257 000104 JMP SEND ;SIZE MEM.,GOTO "MOVE TEST"-ROUT.
57 00260 063077 HALT ;INTERRUPT ERROR, UNKNOWN INTERRUPT
58 ;OCCURED, OR MEMORY (LOC 0-1) FAILED
59 ;ON THE EXPECTED TTO-INTERRUPT, OR MA-
0 ;REGISTER ISN'T CLEARED BY INTERRUPT,
1
00261 000254 JMP MMA2+4
```

0008 .MAIN

01
02 00262 054325 MESS: STA 3,MESSR ;PRINT A TEXT MESSAGE
03 00263 010325 ISZ MESSR
04 00264 031400 LDA 2,0,3 ;AC(2) POINTS TO MESSAGE
05 00265 020046 MESS0: LDA 0,C377 ;A 8 BIT MASK
06 00266 025000 LDA 1,0,2 ;AC(2)=DATA WORD
07 00267 123405 AND 1,0,SNR
08 00270 002325 JMP 0MESSR
09 00271 004326 JSR TYPE
10 00272 020047 LDA 0,C1774
11 00273 123705 ANDS 1,0,SNR
12 00274 002325 JMP 0MESSR
13 00275 004326 JSR TYPE
14 00276 151400 INC 2,2
15 00277 000265 JMP MESS0

16
17 00300 054325 POCT: STA 3,MESSR ;PRINT OCTAL NUMBER
18 00301 030045 LDA 2,C60
19 00302 102400 SUB 0,0
20 00303 125120 MOVZL 1,1
21 00304 101100 MOVL 0,0
22 00305 143000 ADD 2,0
23 00306 004326 JSR TYPE
24 00307 030050 LDA 2,M5
25 00310 102400 POCT0: SUB 0,0
26 00311 125120 MOVZL 1,1
27 00312 101100 MOVL 0,0
28 00313 125120 MOVZL 1,1
29 00314 101100 MOVL 0,0
30 00315 125120 MOVZL 1,1
31 00316 101100 MOVL 0,0
32 00317 034045 LDA 3,C60
33 00320 163000 ADD 3,0
34 00321 004326 JSR TYPE
35 00322 151404 INC 2,2,SZR
36 00323 000310 JMP POCT0
37 00324 002325 JMP 0MESSR

38
39 00325 000000 MESSR: 0

40
41 00326 063511 TYPE: SKPBZ T10
42 00327 000326 JMP .-1
43 00330 061111 DOAS 0,T10
44 00331 063535 SKPBZ 35
45 00332 000331 JMP .-1
46 00333 061135 DOAS 0,35
47 00334 001400 JMP 0,3

48
49 MESIZE: .TXTE !<15><12>LAST LOC TESTED: !

00335 005215
00336 040714
00337 152123
00340 146240
00341 141717
00342 152240
00343 051705
00344 142724
00345 035104
00346 000240

50 CRLF: .TXTE !<15><12>!

00347 005215
00350 000000

1 0009 .MAIN

01
02 MSOCT: ,TXTE ! (OCT)!

00351 024240
00352 141717
00353 124724
00354 000000

03
04 PAMES: ,TXTE !<15><12>PASS !

00355 005215
00356 040520
00357 051523
00360 000240

05
06 00361 054007 MEPAS: STA 3,ERET
07 00362 004262 JSR MESS
08 00363 000355 PAMES
09 00364 010012 ISZ MESCO
10 00365 101000 MOV 0,0
11 00366 024012 LDA 1,MESCO
12 00367 004300 JSR POCT
13 00370 004262 JSR MESS
14 00371 000351 MSOCT
15 00372 002007 JMP 0ERET
16

0010 .MAIN

;THE FOLLOWING PROG.PART IS RELOCATED IN MEMORY TO TEST
;DIFFERENT AREAS. AFTER A RELOCATIN THE ADDRESS OF
;"BEGIN" IS STOKED IN LOC. "PLOC" (PAGE ZERO).

```
07      000377      .LOC  377
09 00377 063077      HALT                ;OPERATOR ERROR FIX C(ADR)
10 00400 004401 BEGIN: JSR                ,+1-,1
11 00401 054103      STA                3,RETURN
12 00402 014010      DSZ                PACOU ;SKIP IF 128 RUNS
13 00403 000404      JMP                BEG01
14 00404 034013      LDA                3,PACON
15 00405 054010      STA                3,PACOU
16 00406 006011      JSR                0,IMPAS ;MESS: PASS
17 00407 034003 BEG01: LDA                3,ADR ;AC3:= ADDRESS OF FIRST WORD IN FREE
18 00410 030051      LDA                2,C070000 ;AREA
19 00411 020004      LDA                0,FINAL ;AC0:= END OF MEM
20 00412 143400      AND                2,0
21 00413 040053      STA                0,EDIST ;EDIST:= 0 FOR 4K,1 FOR 8K,2 FOR
22                                     ;12K,3 FOR 16K ... 7 FOR 32K
24 00414 173400      AND                3,2
25 00415 050052      STA                2,MODUAL;THE MEMORY MODUAL IN TEST, #
26                                     ;AS ABOVE
27 00416 034077 MKT:  LDA                3,TABL ;AC3:= ADDRESS OF PATTERN TABLE I
28                                     ;(STARTING AT "TABS")
29 00417 024064      LDA                1,C20
30 00420 044076      STA                1,MKTK
31 00421 020006      LDA                0,PATT
32 00422 024005 MKT0: LDA                1,INH
33 00423 107400      AND                0,1
34 00424 045400      STA                1,0,3 ;STORE 40 PATTERNFIGURES IN PATTERN-
35 00425 100000      COM                0,0 ;TABLE I (STARTING TABS)
36 00426 024005      LDA                1,INH
37 00427 107400      AND                0,1
38 00430 045420      STA                1,20,3
39 00431 100000      COM                0,0
40 00432 101122      MOVZL               0,0,SZC
41 00433 101400      INC                0,0
42 00434 175400      INC                3,3
43 00435 014076      DSZ                MKTK
44 00436 000764      JMP                MKT0
45                                     ;40 PATTERNFIGURES STORED, MOVE
46                                     ;THE PATTERN TABLE I TO FREE CO-
47                                     ;RE (STARTING AT "ADR") IN BLOCKS OF
48                                     ;40
49
50 00437 030003 IPAT: LDA                2,ADR
51 00440 102400      SUB                0,0
52 00441 024072      LDA                1,CY ;AC1:= 400 OR 10000
53 00442 147414      AND#               2,1,SZR
54 00443 100000 IPAT0: COM                0,0
55
```

1 0011 ,MAIN

```
01
02 00444 060300 FILL: NIOP 0 ;SYNC AT A74
03 00445 024073 LDA 1,CMSK
04 00446 044075 STA 1,TMES ;TMES:= -1 OR -2
05 00447 024061 LDA 1,BPROG ;HPROG:= STARTADR - 40
06 00450 132033 ADCZ# 1,2,SNC ;SKIP IF C(BPROG)<= C(ADR)
07 00451 000404 JMP FILL1
08 00452 024062 LDA 1,EPROG ;EPROG:= STARTADR - 40 + SIZE
09 00453 132433 SUBZ# 1,2,SNC ;SKIP IF C(EPROG)<= C(ADR)
10 00454 000452 JMP FILL0
11 00455 024064 FILL1: LDA 1,C20
12 00456 034077 FILL2: LDA 3,TABL ;AC3:= ADDRESS OF PATTERN TAB-
13 00457 101014 MOV# 0,0,SZR ;LE I (STARTING AT "TABS")
14 00460 137000 ADD 1,3
15 00461 025400 LDA 1,0,3
16 00462 045000 STA 1,0,2
17 00463 025401 LDA 1,1,3
18 00464 045001 STA 1,1,2
19 00465 025402 LDA 1,2,3
20 00466 045002 STA 1,2,2
21 00467 025403 LDA 1,3,3
22 00470 045003 STA 1,3,2
23 00471 025404 LDA 1,4,3
24 00472 045004 STA 1,4,2
25 00473 025405 LDA 1,5,3
26 00474 045005 STA 1,5,2
27 00475 025406 LDA 1,6,3
28 00476 045006 STA 1,6,2
29 00477 025407 LDA 1,7,3
30 00500 045007 STA 1,7,2
31 00501 025410 LDA 1,10,3
32 00502 045010 STA 1,10,2
33 00503 025411 LDA 1,11,3
34 00504 045011 STA 1,11,2
35 00505 025412 LDA 1,12,3
36 00506 045012 STA 1,12,2
37 00507 025413 LDA 1,13,3
38 00510 045013 STA 1,13,2
39 00511 025414 LDA 1,14,3
40 00512 045014 STA 1,14,2
41 00513 025415 LDA 1,15,3
42 00514 045015 STA 1,15,2
43 00515 025416 LDA 1,16,3
44 00516 045016 STA 1,16,2
45 00517 025417 LDA 1,17,3
46 00520 045017 STA 1,17,2
47 00521 024064 LDA 1,C20
48 00522 133000 ADD 1,2 ;ADD 20 TO WORD POINTER
49 00523 010075 ISZ TMES
50 00524 000732 JMP FILL2
51 00525 000403 JMP FILL3
52 00526 024064 FILL0: LDA 1,C20
53 00527 133000 ADD 1,2 ;ADD 20 TO WORD POINTER
54 00530 024042 FILL3: LDA 1,C77
55 00531 133414 AND# 1,2,SZR
56 00532 000711 JMP IPAT0-.,1
57 00533 024004 LDA 1,FINAL
58 00534 146432 SUBZ# 2,1,SZC ;SKIP IF WORD POINTER > FINAL
59 00535 000703 JMP IPAT+1-.,1 ;(MEANS IF MEM IS FILLED UP)
60
```

0012 ,MAIN

```
01
02 00536 030052 DISTUR: LDA      2,MODUAL;DISTURB MODULE SELECT
03 00537 020043          LDA      0,C7777 ;DISTURB AT LOCATION
04 00540 024053          LDA      1,EDIST ;0101,0202,0303,ETC.
05 00541 123000          ADD      1,0
06 00542 024040          LDA      1,C101 ;EVERY OTHER CORE IN MEMORY
07 00543 133000          ADD      1,2 ;IS DISTURBED AT LEAST
08 00544 074477          READS    3 ;1024 TIMES + INHIBIT DISTURBS,
09 00545 175112          MOVL#    3,3,SZC ;BUT ONLY IF SWITCH 0
10 00546 142433          SUBZ#   2,0,SNC ;IS SET TO A ONE.
11 00547 000406          JMP      ICHK ;END OF DISTURB
12 00550 176400          SUB      3,3
13 00551 025000          LDA      1,0,2
14 00552 175704          INCS    3,3,SZR
15 00553 000776          JMP      .-2
16 00554 000766          JMP      DISTURB+4
```

;COMPARE THE MEM PATTERNS WITH PATTERNABLE I

```
21
22 00555 030003 ICHK:  LDA      2,ADR
23 00556 102400          SUB      0,0
24 00557 024072          LDA      1,CY ;AC1:= 400 OR 10000
25 00560 147414          AND#    2,1,SZR
26 00561 100000 ICHK0: COM      0,0
27 00562 040074          STA      0,FLAG
28
29 00563 024073 ICHK1: LDA      1,CMSK ;AC1:= 17777 OR 177776(AFTER
30 00564 044075          STA      1,IMES ;FIRST PASS)
31 00565 024061          LDA      1,BPROG
32 00566 132033          ADCZ#   1,2,SNC ;SKIP IF C(BPROG)< C(ADR)
33 00567 000404          JMP      CHEK
34 00570 024062          LDA      1,EPROG
35 00571 132433          SUBZ#   1,2,SNC ;SKIP IF C(EPROG) <= C(ADR)
36 00572 000436          JMP      CHEK0
37
38 00573 024064 CHEK:  LDA      1,C20
39 00574 044076          STA      1,MKTK
40 00575 034100          LDA      3,TABL0 ;AC3:= (ADDRESS OF PATTERNABLE I)-1 =
41 ;"TABS"-1
42 00576 167000          ADD      3,1 ;AC1:= "TABS"-1+20
43 00577 054020          STA      3,20
44 00600 044021          STA      1,21
45 00601 020074          LDA      0,FLAG
46 00602 101015          MOV#    0,0,SNR
47 00603 000403          JMP      .+3
48 00604 044020          STA      1,20
49 00605 054021          STA      3,21
50
```

1 0013 .MAIN

```
01
02 00606 022020 CHEK1: LDA 0,020 ;LOAD FROM PATTERN TABLE I
03 00607 025000 LDA 1,0,2 ;LOAD FROM MEMORY IN TEST
04 00610 106434 SUBZ# 0,1,SZR ;COMPARE
05 00611 004467 JSR ERR1 ;ERROR
06 00612 022021 LDA 0,021
07 00613 041000 STA 0,0,2
08 00614 025000 LDA 1,0,2
09 00615 122434 SUBZ# 1,0,SZR
10 00616 004463 JSR ERR2
11 00617 014020 DSZ 20
12 00620 036020 LDA 3,020
13 00621 055000 STA 3,0,2
14 00622 151400 INC 2,2 ;INCREMENT WORD POINTER
15 00623 014076 DSZ MKTK
16 00624 000762 JMP CHEK1
17 00625 010075 ISZ TMES
18 00626 000745 JMP CHEK
19 00627 000403 JMP CHEK3
20 00630 024064 CHEK0: LDA 1,C20
21 00631 133000 ADD 1,2
22 00632 020074 CHEK3: LDA 0,FLAG
23 00633 024042 LDA 1,C77
24 00634 133414 AND# 1,2,SZR
25 00635 000724 JMP ICHK0
26 00636 024004 LDA 1,FINAL
27 00637 146432 SUBZ# 2,1,SZC ;SKIP IF WORD POINTER > END OF CUR. MEM
28 00640 000716 JMP ICHK+1
29
30 00641 024006 UPDT: LDA 1,PATT
31 00642 130000 COM 1,2
32 00643 125122 MOVZL 1,1,SZC
33 00644 125400 INC 1,1
34 00645 020432 LDA 0,UPDTC
35 00646 101233 MOVZR# 0,0,SNC ;SKIP IF C(UPDTC) IS ODD
36 00647 145000 MOV 2,1
37 00650 044006 STA 1,PATT ;CHANGE PATTERN WORD
38 00651 014426 DSZ UPDTC
39 00652 002101 JMP 0BEG
40 00653 024066 LDA 1,C40
41 00654 044423 STA 1,UPDTC ;RESTORE UPDTC
42 00655 020070 LDA 0,WHICH
43 00656 101400 INC 0,0
44 00657 040070 STA 0,WHICH
45 00660 024064 LDA 1,C20
46 00661 030041 LDA 2,C400
47 00662 176000 ADC 3,3 ;AC3:= 177777
48 00663 101223 MOVZR 0,0,SNC ;SKIP IF C(WHICH) IS ODD
49 00664 000404 JMP UPDT0
50 00665 024066 LDA 1,C40
51 00666 030067 LDA 2,C10K
52 00667 176120 ADCZL 3,3 ;AC3:= 177776,CARRY:= 1
53 00670 044071 UPDT0: STA 1,CX
54 00671 050072 STA 2,CY
55 00672 054073 STA 3,CMSK
56 00673 101223 MOVZR 0,0,SNC ;SKIP IF B14 OF C(WHICH) IS ONE
57 00674 002101 JMP 0BEG
58 00675 034103 LDA 3,RETURN
59 00676 001400 JMP 0,3 ;RETURN TO LOC. "MOVE"+10
60
61 00677 000040 UPDTC: 40
62
```

```

1 0014 ,MAIN
01
02 00700 101020 ERR1: MOVZ 0,0 ;DISTURB ENTRY
03 00701 054007 ERR2: STA 3,ERET ;UNDISTURB ENTRY
04 00702 034044 LDA 3,C207
05 00703 063411 SKPBN TTO
06 00704 075111 DOAS 3,TTO ;SET SWITCH 15 TO
07 00705 004411 JSR ALARM
08 00706 074477 READS 3 ;INHIBIT HALT
09 00707 175200 MOVR 3,3
10 00710 034060 LDA 3,PLOC
11 00711 054103 STA 3,RETURN;IF CONTINUE AFTER ERROR-HALT, THEN
12 ;REPEAT CURRENT PASS
13
14 00712 101003 MOV 0,0,SNC ;SKIP IF SW15(1)
15 00713 063077 ER: HALT ;ERROR IN MEMORY - AC0= WORD STORED,
16 ;AC1= ERRORWORD LOADED, AC2= ERROR-
17 ;ADDRESS.
18 00714 114040 COMO 0,3
19 00715 002007 JMP 0,ERET
20
21 00716 054414 ALARM: STA 3,HERR
22 00717 034414 LDA 3,BIT4
23 00720 075032 DOA 3,32
24 00721 014413 DSZ ,CNT
25 00722 000404 JMP ,RTUR
26 00723 176420 SUBZ 3,3
27 00724 075032 DOA 3,32
28 00725 000402 JMP ,+2
29 00726 034407 ,RTUR: LDA 3,C1
30 00727 054405 STA 3,CNT
31 00730 034402 LDA 3,HERR
32 00731 001400 JMP 0,3
33
34
35 00732 000000 ,HERR: 0
36 00733 004000 BIT4: 104
37 00734 000001 ,CNT: 1
38 00735 000001 C1: 1
39 000040 TABS: ,BLK 40 ;TABLE WITH REFERENCE PATTERNS
40
41 00776 000776 CEND: .
42
43

```

I 0015 .MAIN

01

02

.END

0016 .MAIN

ADR	000003	5/05	6/14	10/17	10/50	12/22			
ALARM	000716	14/07	14/21						
BEG01	000407	10/13	10/17						
BEGIN	000400	5/31	5/50	10/10					
BGL	000101	5/49	6/18	6/19	6/20	13/39	13/57		
BIT4	000733	14/22	14/36						
BPROG	000061	5/33	6/23	11/05	12/31				
C0700	000051	5/25	10/18						
C1	000735	14/29	14/38						
C101	000040	5/16	12/06						
C10K	000067	5/39	13/51						
C1774	000047	5/23	8/10						
C20	000064	5/36	10/29	11/11	11/47	11/52	12/38	13/20	13/45
C207	000044	5/20	14/04						
C377	000046	5/22	8/05						
C40	000066	5/38	13/40	13/50					
C400	000041	5/17	13/46						
C60	000045	5/21	8/18	8/32					
C77	000042	5/18	11/54	13/23					
C7777	000043	5/19	12/03						
CREG	000102	5/50	6/30						
CEND	000776	5/31	14/41						
CHEK	000573	12/33	12/38	13/18					
CHEK0	000630	12/36	13/20						
CHEK1	000606	13/02	13/16						
CHEK3	000632	13/19	13/22						
CMA	000056	5/30	7/38						
CMSK	000073	5/43	11/03	12/29	13/55				
CRLF	000347	8/50							
CX	000071	5/41	13/53						
CY	000072	5/42	10/52	12/24	13/54				
DISTU	000536	12/02	12/16						
EDIST	000053	5/27	10/21	12/04					
EPROG	000062	5/34	6/28	11/08	12/34				
ER	000713	14/15							
ERET	000007	5/09	9/06	9/15	14/03	14/19			
ERR1	000700	13/05	14/02						
ERR2	000701	13/10	14/03						
FILL	000444	11/02							
FILL0	000526	11/10	11/52						
FILL1	000455	11/07	11/11						
FILL2	000456	11/12	11/50						
FILL3	000530	11/51	11/54						
FINAL	000004	5/06	6/04	7/20	7/31	10/19	11/57	13/26	
FLAG	000074	5/44	12/27	12/45	13/22				
ICLK	000555	12/11	12/22	13/28					
ICLK0	000561	12/26	13/25						
ICLK1	000563	12/29							
IMPAS	000011	5/11	10/16						
INH	000005	5/07	10/32	10/36					
IPAT	000437	10/50	11/59						
IPAT0	000443	10/54	11/56						
K1000	000054	5/28	7/02						
M201	000055	5/29	7/18						
M40	000065	5/37	6/12	6/21	6/25				
M5	000050	5/24	8/24						
MEPAS	000361	5/11	9/06						
MESCU	000012	5/12	7/22	9/09	9/11				
MESIZ	000335	7/30	8/49						

0017 ,MAIN

MESS	000262	7/29	8/02	9/07	9/13			
MESS0	000265	8/05	8/15					
MESSK	000325	8/02	8/03	8/08	8/12	8/17	8/37	8/39
MKT	000416	10/27						
MKT0	000422	10/32	10/44					
MKTK	000076	5/46	10/30	10/43	12/39	13/15		
MMA	000234	6/39	7/34					
MMA1	000243	7/41	7/55					
MMA2	000250	5/30	7/49	7/62				
MODUA	000052	5/26	10/25	12/02				
MOVE	000136	6/29	6/36					
MSIZ	000177	5/03	7/02					
MSIZ1	000201	7/04	7/16					
MSIZ2	000215	7/11	7/18					
MSOCI	000351	9/02	9/14					
PACON	000013	5/13	7/24	10/14				
PACOU	000010	5/10	7/25	10/12	10/15			
PAMES	000355	9/04	9/08					
PATT	000006	5/08	10/31	13/30	13/37			
PLOC	000060	5/32	6/17	6/37	14/10			
POCT	000300	7/32	8/17	9/12				
POCT0	000310	8/25	8/36					
PSIZE	000057	5/31	6/05					
RANDU	000063	5/35	6/03					
RETUR	000103	5/51	10/11	13/58	14/11			
SEND	000104	6/02	7/56					
TABL	000077	5/47	10/27	11/12				
TABL0	000100	5/48	12/40					
TABS	000736	5/47	5/48	14/39				
TMES	000075	5/45	11/04	11/49	12/30	13/17		
TYPE	000326	8/09	8/13	8/23	8/34	8/41		
UPDT	000641	13/30						
UPDT0	000670	13/49	13/53					
UPDTC	000677	13/34	13/38	13/41	13/61			
WHICH	000070	5/40	7/23	13/42	13/44			
.CNT	000734	14/24	14/30	14/37				
.HERR	000732	14/21	14/31	14/35				
.RAND	000151	6/02	6/42					
.RTUR	000726	14/25	14/29					
.UD03	000173	6/42	6/43	6/59	6/61			
.UD10	000174	6/47	6/50	6/62				
.UD20	000175	6/56	6/63					
.UD21	000176	6/46	7/00					

