



DATA GENERAL
CORPORATION

Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Memory Checkerboard III

TAPES

Binary 095-000031

ABSTRACT

Checkerboard III is a maintenance program designed to produce worst case noise conditions on the sense/inhibit wires. The program should be run to insure proper operation of sense amps, inhibit drivers, and memory currents.

CHECKERBOARD III

;1. ABSTRACT

; ;
; CHECKERBOARD III IS A MAINTENANCE PROGRAM
; DESIGNED TO PRODUCE WORST CASE NOISE CONDITIONS
; ON THE SENSE/INHIBIT WIRES. THE PROGRAM SHOULD BE
; RUN TO INSURE PROPER OPERATION OF SENSE AMPS,
; INHIBIT DRIVERS, AND MEMORY CURRENTS.
;

2. MACHINE REQUIREMENTS

;2.1 NOVA/SUPERNOVA PROCESSOR
;2.2 4K READ/WRITE MEMORY

;3. SWITCH SETTINGS

;3.1 STARTING ADDRESS =000002
;3.2 SWITCH 0(1) =1024 READ/WRITE DISTURBS
;3.3 SWITCH 15(1) =INHIBIT HALT ON ERROR

;4. OPERATING PROCEDURE

;4.1 LOAD THE PROGRAM VIA THE BINARY LOADER
;4.2 SET SWITCHES TO 000002
;4.3 PRESS START
;4.3.1 THE PROGRAM WILL PRINT THE HIGHEST
; LOCATION THE PATTERN IS TO USE.
;4.4 IF THE FAILURES ARE MARGINAL, SETTING SWITCH
; 0 MAY AID IN INDUCING A FAILURE TO OCCUR.
;4.5 WHEN SCOPING OR ADJUSTING CURRENTS, SETTING
; SWITCH 15 WILL INHIBIT THE ERROR HALT. THE
; BELL WILL STILL BE RUNG.
;4.6 PROGRAM MODIFICATIONS
;4.6.1 C(3)=ADR THE STARTING PATTERN ADDRESS
;4.6.2 C(5)=INHIBIT INHIBIT THE CHECKERBOARD
; PATTERN ON CLEARED BITS.

```

;5.          PROGRAM OUTPUT/ERROR DESCRIPTION
;5.1         AT EACH OCCURRENCE OF ERROR, IF THE TELETYPE IS
;           NOT BUSY THE BELL WILL BE RUNG. IF SWITCH (15) IS
;           ZERO THE PROGRAM WILL HALT AT LOCATION "ER".
;5.2         WHEN AN ERROR HALT OCCURRS:
;           C(1)=THE ERROR WORD
;           C(2)=THE ERROR ADDRESS
;           C(3)=ADDRESS OF PROGRAM
;5.3         SET SWITCH (15) IF SCOPING, PRESS CONTINUE.
;5.4         SYNC PULSES
;           A "P" PULSE (A74) IN STORE CYCLE.
;           A "S" PULSE (A52) CHECK ONES PATTERN WORD.
;           A "C" PULSE (A50) CHECK ZEROS PATTERN WORD.

```

```

;6.          PROGRAM DESCRIPTION
;6.1         STORE THE CHECKERBOARD PATTERN
;6.2         IF SWITCH 0(1) DISTURB THE CONTENTS OF MEMORY BY
;           REFERENCING LOCATIONS 0101, 0202, 0303, ETC. 512
;           TIMES. THIS PRODUCES 1024 READ/WRITE DISTURBS.
;6.3         CHECK THE PATTERN WORD
;6.4         COMPLEMENT AND CHECK THE WORD
;6.5         RESTORE THE WORD
;6.6         WHEN THE END OF THE PATTERN IS REACHED THE
;           PROGRAM COMPLEMENTS THE PATTERN WORD AND RE -
;           TURNS TO STEP 6. 1.

```

```

;7.          LIMITATIONS
;           NONE

```

```

000002 .LOC 2
000002 000164 JMP MSIZ
000003 00060020 ADR: 620
000004 007577 FINAL: 7577
000005 177777 INH: -1
000006 000000 PATT: 0
000007 000000 ERET: 0

```

```

;PATTERN STARTING ADDRESS
;PATTERN FINAL ADDRESS
;MASK FOR INHIBITED BITS
;PATTERN WORD

```

000040 .LOC 40

```

000040 000017 C17: 17
000041 000400 C400: 400
000042 000077 C77: 77
000043 007777 C7777: 7777
000044 000207 C207: 207
000045 000101 C101: 101
000046 060200 CNIOC: NIOC 0
000047 060100 CNIOS: NIOS 0
000050 070000 C070000: 070000
000051 000000 MODUAL: 0
000052 000000 EDIST: 0
000053 001000 K1000: 1000
000054 177577 M201: -201
000055 000224 CMA: MMA2
000056 177625 PSIZE: BEGIN-CEND
000057 000000 PLOC: 0
000060 000000 BPROG: 0
000061 000000 EPROG: 0
000062 012345 RANDOM: 12345
000063 177760 M20: -20
000064 000020 C20: 20
000065 000432 CBEG: BEGIN
000066 000000 RETURN: 0

```

```

000067 004126 SEND: JSR .RAND
000070 000062 RANDOM
000071 024004 LDA 1,FINAL
000072 030056 LDA 2,PSIZE
000073 147000 ADD 2,1
000074 101220 MOVER 0,0
000075 122422 SUBE 1,0,SEC
000076 000075 JMP .-1
000077 123900 ADD 1,0
00100 024063 LDA 1,M20
00101 123400 AND 1,0
00102 024003 LDA 1,ADR
00103 122433 SUBE# 1,0,SNC
00104 121000 MOV 1,0

```

```

;GET A RNADOM #
;AC0=# MODULO C(FINAL).
;IF # TO SMALL USE
;C(ADR) FOR STARTER.

```

```

00105 040057 STA 0,PLOC
00106 034063 LDA 3,M20
00107 117000 ADD 0,3
00110 054060 STA 3,BPROG
00111 142400 SUB 2,0
00112 040061 STA 0,EPROG

```


```
00113 145000 MOVE:  MOV 2,1
00114 030065      LDA 2,CBEG
00115 021000      LDA 0,0,2      ;MOVE A COPY OF
00116 041420      STA 0,20,3    ;CHECKERBOARD TO
00117 175400      INC 3,3      ;SELECTED SPOT.
00120 151400      INC 2,2
00121 125404      INC 1,1,SZR    ;TEST FOR LAST REG
00122 000115      JMP MOVE+2     ;TO BE MOVED.
00123 034057      LDA 3,PLOC
00124 005401      JSR 1,3      ;EXIT TO PROG.
00125 000212      JMP MMA

00126 054160 •RAND:  STA 3,•UD03    ;GENERATE A RANDOM #
00127 010160      ISZ •UD03
00130 044156      STA 1,•UD01
00131 050157      STA 2,•UD02
00132 031400      LDA 2,0,3
00133 021000      LDA 0,0,2
00134 004143      JSR •UD50
00135 034162      LDA 3,•UD20
00136 163000      ADD 3,0
00137 041000      STA 0,0,2
00140 024156      LDA 1,•UD01
00141 030157      LDA 2,•UD02
00142 002160      JMP 0,•UD03    ;RETURN

00143 024163 •UD50:  LDA 1,•UD21
00144 044161      STA 1,•UD10
00145 105120      MOVZL 0,1
00146 125120      MOVZL 1,1
00147 014161      DSE •UD10
00150 000146      JMP •-2
00151 107000      ADD 0,1
00152 125120      MOVZL 1,1
00153 125120      MOVZL 1,1
00154 123000      ADD 1,0
00155 001400      JMP 0,3

00156 000000 •UD01:  0
00157 000000 •UD02:  0
00160 000000 •UD03:  0
00161 000000 •UD10:  0
00162 033031 •UD20:  33031

00163 000010 •UD21:  10
```


```
00164 020053 MSIZ: LDA 0,K1000 ;SIZE THE MEMORY
00165 115000 MOV 0,3
00166 031400 MSIZ1: LDA 2,0,3 ;SAVE C(MEM)
00167 055400 STA 3,0,3
00170 025400 LDA 1,0,3
00171 051400 STA 2,0,3 ;RESTORE MEMORY

00172 125014 MOV# 1,1,SZR
00173 124015 COM# 1,1,SNR
00174 000202 JMP MSIZ2 ;END OF MEMORY
00175 136414 SUB# 1,3,SZR ;AC1=BAD, AC3=GOOD
00176 063077 HALT ;MEMORY FAILED.
00177 117000 ADD 0,3
00200 175113 MOVL# 3,3,SNC ;INCREMENT MEMORY ADDRESS.
00201 000166 JMP MSIZ1 ;TEST FOR 32K.

00202 020054 MSIZ2: LDA 0,M201
00203 163000 ADD 3,0
00204 040004 STA 0,FINAL
00205 004372 JSR CRLF
00206 004247 JSR MESS
00207 000234 MESIZE
00210 024004 LDA 1,FINAL
00211 004266 JSR POCT

00212 063611 MMA: SKPDN TTO
00213 000212 JMP -1 ;WAIT FOR TTO DONE.
00214 020055 LDA 0,CMA
00215 040001 STA 0,1 ;SET INTERRUPT RETURN.
00216 152520 SUBZL 2,2 ;THIS PROG TEST FOR INTERRUPT
00217 102000 MMA1: ADC 0,0 ;ABILITY TO CLEAR MA.
00220 040000 STA 0,0
00221 025000 LDA 1,0,2
00222 060177 NIOS CPU ;ENABLE INTERRUPT
00223 005000 JSR 0,2 ;SET BIT INTO MA
00224 045000 MMA2: STA 1,0,2
00225 020000 LDA 0,0
00226 112414 SUB# 0,2,SZR ;AC0=PC STORED
00227 063077 HALT ;AC2=CORRECT. MEMORY FAILED!
00230 151120 MOVZL 2,2
00231 151113 MOVL# 2,2,SNC
00232 000217 JMP MMA1
00233 000067 JMP SEND

00234 040714 MESIZE: .TXTE ILA
00235 152123 ST
00236 146240 L
00237 141717 OC
00240 152101 AT
00241 147711 IO
00242 120116 N
00243 142724 TE
00244 152123 ST
00245 042305 ED
00246 000240 I
```

???

;TTO NON INTERRUPT PACKAGE
;"MESS" PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLR
;"CHAR" PRINTS ASCII CHARACTER, C(0)R,C(0)L MUST BE 0
;WILL RETURN +2 IF C(0)R=0,CORRECTS THE PARITY,33 SIMUL
;"TYPE" PRINTS C(0)R. MUST HAVE PROPER PARITY. RETURN IS
;TO CALL+1.REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF
;"CRLF" PRINTS A CARRIAGE RETURN
;"POCT" PRINTS C(1) IN OCTAL FOLLEWED BY A TAB
;"PDEC" PRINTS C(1) IN DECIMAL,LEADING ZEROS SUPPRESSED,
;FOLLOWED BY A TAB.

```
00247 054546 MESS: STA 3,MESSR ;PRINT A TEXT MESSAGE
00250 010545 ;SZ MESSR
00251 031400 LDA 2,0,3 ;C(2) POINTS TO MESSAGE
00252 024542 LDA 1,C377 ;A 8 BIT MASK
00253 021000 LDA 0,0,2 ;C(2)=DATA WORD
00254 125112 MOVL# 1,1,SZC
00255 123701 ANDS 1,0,SKP
00256 123401 AND 1,0,SKP ;C(0)=DATA CHARACTER RIGHT
00257 151400 INC 2,2 ;INC TO NEXT WORD
00260 124000 COM 1,1 ;FLIP MASK
00261 004343 JSR CHAR ;PRINT
00262 000253 JMP MESS+4 ;ANOTHER
00263 002532 JMP @MESSR ;LAST

00264 020525 ZOCT: LDA 0,CH240
00265 101001 MOV 0,0,SKP

00266 020525 POCT: LDA 0,C60
00267 030322 LDA 2,OCTAB ;PRINT C(1) IN OCTAL
00270 000273 JMP +3
00271 030332 PDEC: LDA 2,DECTB ;PRINT C(1) IN DECIMAL
00272 020517 LDA 0,CH240 ;SUPPRESS LEADING ZEROS
00273 054342 STA 3,RADRET ;BOTH ENTRYS PRINT NUMBER
00274 040341 STA 0,ZSUPP ;THEN TAB TO NEXT POSITION
00275 050276 STA 2,+1
00276 000000 DECOCT: 0 ;A"LDA 2,TABLE" INSTRUCTION
00277 010276 ISZ +-1
00300 034342 LDA 3,RADRET ;SETUP "TAB" AT END
00301 020503 LDA 0,CHTAB
00302 151005 MOV 2,2,SNR ;IF TABLE ENTRY=0
00303 000343 JMP CHAR ;EXIT WITH TAB
00304 034341 LDA 3,ZSUPP ;ZEROS SUPPRESS STUF
00305 102400 SUB 0,0
00306 146512 DECOT: SUBL# 2,1,SEC
00307 000314 JMP DECP
00310 146400 SUB 2,1 ;FORM THE DIGIT
00311 034502 LDA 3,C60
00312 101400 INC 0,0
00313 000306 JMP DECOT
00314 151235 DECP: MOVER# 2,2,SNR
00315 034476 LDA 3,C60
00316 054341 STA 3,ZSUPP ;C(0)=DIGIT
00317 163000 ADD 3,0 ;MAKE ASCII
00320 004343 JSR CHAR ;PRINT
00321 000276 JMP DECOCT ;GET NEXT DIGIT
```

!!!

```
00322 030323  OCTAB:  LDA 2, .+1
00323 100000          100000
00324 010000          10000
00325 001000          1000
00326 000100          100
00327 000010          10
00330 000001          1
00331 000000          0

00332 030333  DECTB: LDA 2, .+1
          000012  .RDX 10
00333 023420          10000
00334 001750          1000
00335 000144          100
00336 000012          10
00337 000001          1
00340 000000          0
          000010  .RDX 8

00341 000000  ESUPP:  0
00342 000000  RADRET: 0

00343 054442  CHAR:   STA 3, CHRET      ;PRINT C(0) RIGHT
00344 101325          MOVES 0, 0, SNR      ;RETURN +2 IF NULL
00345 001401          JMP 1, 3
00346 040440          STA 0, CHSAV
00347 176000          ADC 3, 3          ;COMPUTE THE PARITY
00350 117000          ADD 0, 3
00351 163404          AND 3, 0, SZR
00352 000347          JMP .-3
00353 176660          SUBCR 3, 3      ;COMBIND PARITY WITH CHAR
00354 020432          LDA 0, CHSAV
00355 163300          ADDS 3, 0

00356 034426  CHAR1:  LDA 3, CHTAB      ;IS THIS A TAB
00357 116405          SUB 0, 3, SNR
00360 000363          JMP .+3          ;YES
00361 004435          JSR TYPE        ;NO PRINT IT
00362 002423          JMP 0CHRET      ;EXIT

00363 020424          LDA 0, CHORE
00364 034424          LDA 3, CHAR7
00365 117405          AND 0, 3, SNR
00366 002417          JMP 0CHRET
00367 020422          LDA 0, CH240
00370 004426          JSR TYPE
00371 000363          JMP .-6
```

111

```
00372 054420 CRLF: STA 3,CRLFR ;SAVE RETURN
00373 020410 LDA 0,C215
00374 004343 JSR CHAR ;PRINT CARRIAGE AND LF
00375 020405 LDA 0,C212
00376 004343 JSR CHAR
00377 102400 SUB 0,0
00400 040407 STA 0,CHORZ ;CLEAR HORZ POSISTION
00401 002411 JMP 0CRLFR ;EXIT
```

```
00402 000212 C212: 212
00403 000215 C215: 215
00404 000011 CHTAB: 11
00405 000000 CHRET: 0
00406 000000 CHSAV: 0
00407 000000 CHORZ: 0
00410 000007 CHAR7: 7
00411 000240 CH240: 240
00412 000000 CRLFR: 0
00413 000060 C60: 60
```

```
00414 000377 C377: 377
00415 000000 MESSR: 0
00416 054412 TYPE: STA 3,TYPRET ;TYPE THE C(0)R IF
00417 010770 ISZ CHORZ
00420 074477 READS 3 ;SWITCH 1(0).
00421 175100 MOVL 3,3
00422 175102 MOVL 3,3,SEC
00423 002405 JMP 0TYPRET ;INHIBIT TYPE EXIT.
00424 063511 SKPBZ TIO
00425 000777 JMP --1
00426 061111 DOAS 0,TIO
00427 002401 JMP 0TYPRET
00430 000000 TYPRET: 0
```

???

```
00431 063077          HALT          ;OPERATOR ERROR FIX C(ADR)
00432 004401  BEGIN:  JSR  ++1
00433 054066          STA 3,RETURN
00434 034003          LDA 3,ADR
00435 030050          LDA 2,C070000
00436 020004          LDA 0,FINAL
00437 143400          AND 2,0
00440 040052          STA 0,EDIST
00441 173400          AND 3,2
00442 050051          STA 2,MODUAL  ;THE MEMORY MODUAL

00443 030003  IPAT:  LDA 2,ADR
00444 024041          LDA 1,C400
00445 020006          LDA 0,PATT      ;PRESET PATTERN
00446 147404          AND 2,1,SZR
00447 100000  IPAT1: COM 0,0
00450 024005          LDA 1,INH
00451 123400          AND 1,0         ;MASK INHIBITED BITS
00452 024040          LDA 1,C17

00453 060300  FILL:  NIOP 0          ;SYNC AT A74
00454 034060          LDA 3,BPROG      ;CODE TO AVOID OVERWRITE
00455 156436          SUBZ# 2,3,SEZ    ;OF THE PROGRAM.
00456 000403          JMP  ++3         ;PATT < PROGRAM BEGIN.
00457 034061          LDA 3,EPROG
00460 172436          SUBZ# 3,2,SEZ    ;DONT SKIP IF > PROG END.
00461 041000          STA 0,0.2
00462 034042          LDA 3,C77
00463 151400          INC 2,2        ;PATTERN
00464 133414          AND# 1,2,SZR    ;SKIP EVERY 16 TIMES
00465 000766          JMP FILL
00466 157414          AND# 2,3,SZR    ;SKIP EVERY 64 TIMES
00467 000760          JMP IPAT1
00470 020004          LDA 0,FINAL     ;TEST FOR FINAL ADDRESS
00471 142432          SUBZ# 2,0,SEC   ;EVERY 64 LOC. 4K
00472 000752          JMP IPAT+1      ;FILL TIME=100MS.
```

111

00473 030051	DISTUR:	LDA 2,MODUAL	;DISTURB MODULE SELECT
00474 020043		LDA 0,C7777	;DISTURB AT LOCATION
00475 024052		LDA 1,EDIST	;0101,0202,0303,ETC.
00476 123000		ADD 1,0	
00477 024045		LDA 1,C101	;EVERY OTHER CORE IN MEMORY
00500 133000		ADD 1,2	;IS DISTURBED AT LEAST
00501 074477		READS 3	;1024 TIMES+INHIBIT DISTURBS.
00502 175112		MOVL# 3,3,SEC	;BUT ONLY IF SWITCH 0
00503 142433		SUB# 2,0,SNC	;IS SET TO A ONE.
00504 000406		JMP ICHECK	;END OF DISTURB
00505 176400		SUB 3,3	
00506 025000		LDA 1,0,2	;REFERANCE MEMORY
00507 175704		INCS 3,3,SER	
00510 000776		JMP *-2	
00511 000766		JMP DISTURB+4	
00512 030003	ICHECK:	LDA 2,ADR	;INITIALIZE CHECK CYCLE
00513 024041		LDA 1,C400	
00514 020006		LDA 0,PATT	;X LINE INIT PATTERN
00515 133414		AND# 1,2,SER	
00516 100000	ICK:	COM 0,0	
00517 034005		LDA 3,INM	
00520 163400		AND 3,0	;MASK INHIBITED BITS
00521 024047		LDA 1,CN10S	; "S" PULSE
00522 114044		COMO 0,3,SER	; "C" PULSE
00523 024046		LDA 1,CN10C	;ON 1/0 DISTURB SIGNALS
00524 044412		STA 1,CHECK	
00525 024060		LDA 1,BPROG	;CODE TO PREVENT PATTERN
00526 146436		SUB# 2,1,SEZ	;CHECK OF PROGRAM.
00527 000407		JMP CHECK	;PATT<PROGRAM BEGIN.
00530 024061		LDA 1,EPROG	
00531 132436		SUB# 1,2,SEZ	
00532 000404		JMP CHECK	;PATT>PROGRAM END
00533 024064		LDA 1,C20	
00534 133000		ADD 1,2	
00535 000416		JMP ECHECK	
00536 000000	CHECK:	0	;A SYNC PULSE ISSUED
00537 025000		LDA 1,0,2	; SIGNALS RWV2,RWV1
00540 106414		SUB# 0,1,SER	
00541 004427		JSR ERR1	
00542 055000		STA 3,0,2	
00543 025000		LDA 1,0,2	
00544 136414		SUB# 1,3,SER	; SIGNALS UV1,UV2
00545 004424		JSR ERR2	
00546 041000		STA 0,0,2	
00547 151400		INC 2,2	
00550 034040		LDA 3,C17	;COUNT 16 TIMES
00551 157414		AND# 2,3,SER	
00552 000764		JMP CHECK	

???

00553 034042	ECHECK:	LDA 3,C77	;	CHECK FOR END OF
00554 157414		AND# 2,3,SZR	;	LINE
00555 000741		JMP ICK	;	EVERY 64 TIMES
00556 024004		LDA 1,FINAL	;	CHECK FOR END OF CORE
00557 146432		SUBZ# 2,1,SZC		
00560 000733		JMP ICHECK+1		
00561 020006		LDA 0,PATT	;	COMP THE
00562 100000		COM 0,0	;	PATTERN
00563 040006		STA 0,PATT		
00564 101004		MOV 0,0,SZR		
00565 000647		JMP BEGIN+2		
00566 034066		LDA 3,RETURN		
00567 001400		JMP 0,3		

00570 101020	ERR1:	MOVZ 0,0	;	DISTURB ENTRY
00571 054007	ERR2:	STA 3,ERET	;	UNDISTURB ENTRY
00572 034044		LDA 3,C207	;	C(1)=ERROR WORD
00573 063411		SKPBN TTO	;	C(2)=ERROR ADDRESS
00574 075111		DOAS 3,TTO	;	SET SWITCH 1 TO
00575 074477		READS 3	;	INHIBIT HALT
00576 175200		MOVR 3,3		
00577 034057		LDA 3,PLOC		
00600 054066		STA 3,RETURN		
00601 101003		MOV 0,0,SNC		
00602 063077	ER:	HALT	;	IF TTY NOT BUSY
00603 114040		CONO 0,3	;	TURN OF TTY IF
00604 002007		JMP 0ERET	;	TO NOISEY.
00605 000605	CEND:	.		
	.END			