

TC11, TU56

TC2-TC11 TEST #2
CZTCBEO

AH-9378E-MC
FICHE 1 OF 1

JUN 1980
COPYRIGHT © 72.80
MADE IN USA



The main body of the document contains a large grid of data, likely a test log or a data table. The grid is organized into approximately 15 columns and 25 rows. Each cell in the grid contains small, faint text or numerical data, which is difficult to read due to the low resolution and high contrast of the image. The data appears to be organized in a structured manner, possibly representing test results for different components or conditions.



000000

IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-9377E-MC
PRODUCT NAME: CZTCBEO TC2 - TC11 TEST 2
DATE: MAR 1980
MAINTAINER: ENGINEERING
AUTHOR: L. R. KOLLER

COPYRIGHT (C) 1972, 1980 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

1.0 GENERAL PROGRAM INFORMATION

1.1 ABSTRACT

TC2 - TC11 TEST2 IS USED TO TEST THE TC11 DECTAPE CONTROL. TC2 USES THE MAINTENANCE BIT FEATURE OF THE TC11 CONTROL TO CHECK THE TC11 CONTROL WITHOUT DEPENDING ON DECTAPE TRANSPORT MOVEMENT. PRIOR TO ACTUAL USE OF THE MAINTENANCE BIT FEATURE, CORRECT OPERATION OF THE INTERRUPT CIRCUITS IS CHECKED, AND THE MAINTENANCE BIT ITSELF IS CHECKED. IF CACHE MEMORY IS PRESENT IN THE PROCESSOR, IT WILL BE TURNED OFF BEFORE THE TESTS ARE RUN.

1.1 SYSTEM REQUIREMENTS HARDWARE PEQUIREMENTS

- A) PDP-11 SYSTEM (12k CORE).
- B) ASR33/35 TELETYPE.
- C) TC11 DECTAPE CONTROL AND AT LEAST ONE TU56 DUAL TRANSPORT.

THE TELETYPE AND TC11 CONTROL MUST HAVE THEIR STANDARD PERIPHERAL ADDRESSES, INTERRUPT LEVELS, AND INTERRUPT VECTOR ADDRESSES. REFER TO SECTION 7.2 IF YOUR SYSTEM DOES NOT HAVE STANDARD PERIPHERAL ADDRESSES.

1.2 SOFTWARE REQUIREMENTS

THIS PROGRAM IS ABLE TO RUN "STAND ALONE" OR UNDER CONTROL OF THE ACT11 MONITOR

1.3 RELATED DOCUMENTS AND STANDARDS

SEE THE ACT11/XXDP PROGRAMERS GUIDE FOR INFORMATION ON RUNNING UNDER ACT 11

1.4 SUGESTED PREREQUISITES

IT IS RECOMENDED THAT ALL MAINDECS THAT CHECK OUT THE BASIC CPU BE RUN BEFORE TC2.

1.5 FAILURE ASSUMPTIONS

THROUGHOUT THIS PROGRAM IT IS ASSUMED THAT THE BASIC CPU IS IN GOOD RUNNING ORDER. IF IT IS NOT THE INFORMATION GAINED BY RUNNING THIS PROGRAM IS LIKELY TO BE FALSE (OR NONEXISTANT IF THE PROGRAM WILL NOT RUN).

2.0 OPERATING INSTRUCTIONS

2.10 LOADING PROCEDURES

THIS PROGRAM'S OBJECT TAPE IS PUNCHED IN ABSOLUTE FORMAT. THE ABS LOADER IS USED TO LOAD THE PROGRAM UNDER STAND ALONE MODE. FOR FOR INFORMATION ON PROGRAM LOADING UNDER CONTROL OF THE VARIOUS MONITOR SYSTEMS, REFER TO THE DOCUMENTS NAMES IN SECTION 1.3 ABOVE. UNDER STAND ALONE MODE, AFTER ACERTAINING THAT THE ABS LOADER PROGRAM IS IN THE PDP-11, FOLLOW THESE STEPS TO LOAD TC2:

- A) PUT THE TC2 BINARY TAPE INTO THE PAPER TAPE READER
- B) SET THE PDP-11 CONSOLE SWITCHES TO 37750
- C) DEPRESS LOAD ADDRESS
- D) DEPRES START (TAPE SHOULD READ IN)

2.2 STARTING PROCEDURE

- A) UNIT 0: REMOTE/WRITE LOCK/. ALL OTHER UNITS OFF.
- B) WALL SWITCH ON, WRTM SWITCH OFF. WALL SWITCH IS LOCATED BEHIND BLANK PANEL ABOVE TU56.
- C) LOAD ADDRESS 000200.
- D) PRESS START.
- E) THE PROGRAM IDENTIFIES ITSELF, TYPES SETUP INSTRUCTIONS, SR OPTIONS MESSAGE, AND HALTS.
- F) MAKE SURE THAT THE SETUP (STEPS A AND B) HAS BEEN PROPERLY DONE, AND SELECT DESIRED SR OPTIONS, IF ANY. NORMAL SR SETTING IS 000000.
- G) PRESS CONT. THE PROGRAM BEGINS EXECUTION.
- H) AT THE END OF EACH PASS THE PASS COUNT IS PRINTED
- I) REFER TO SECTION 6.2 IF ERROR PRINTOUTS OCCUR.

2.3 EXECUTION TIME

EXECUTION TIME IS DEPENDENT ON WHICH MODEL OF PDP11 THE PROGRAM IS TO BE RUN ON. ANY TIMES GIVEN APPLY TO THE PDP-11 MODEL 40 UNLESS OTHERWISE STATED

- A) ONE NORMAL ERROR FREE PASS TAKES APPROXIMATELY 10 SECONDS
- B) ONE SINGLE ITERATION PASS (SR11=1) TAKES ABOUT 5 SECONDS.

*****NOTE*****

THE SINGLE ITERATION PASS IS A CONVENIENT WAY TO QUICKLY DETERMINE IF ANY SOLID PROBLEMS EXIST. FOR A THOROUGH TEST, THE NORMAL ITERATION PASS SHOULD BE RUN.

3.0 ERROR INFORMATION

ERRORS ARE REPORTED IN THIS PROGRAM BY THE FOLLOWING METHODS:

- A) UNCONDITIONAL ERROR HALTS, OR
- B) ERROR PRINTOUT FOLLOWED BY OPTIONAL ERROR HALT.

3.1 UNCONDITIONAL ERROR HALTS

AN UNCONDITIONAL ERROR HALT WILL OCCUR AT THE ADDRESSES LISTED BELOW IF THROUGH HARDWARE OR SOFTWARE FAILURE, PROGRAM CONTROL IS TRANSFERRED TO AN UNEXPECTED AREA BETWEEN 000000 AND 000176.

000002 RESERVED AREA

000016 DEBUG TRAP

000022 IUT TRAP

000040 THROUGH 000176 - SYSTEM SOFTWARE AND INTERRUPT VECTOR AREA
TO FIND OUT WHERE THE PROGRAM WAS AT THE TIME THE FAILURE OCCURRED,

- A. EXAMINE CONTENTS OF REGISTER 6. (ADDRESS 177706).
- B. TRANSFER THE CONTENTS OF REG 6 TO THE SR, LOAD ADDRESS AND EXAMINE.
- C) THE DATA SHOWN IN THE DATA LIGHTS IS THE VALUE OF THE PC WHEN THE FAILURE OCCURRED.
- D) LOCATE IN PROGRAM LISTING THE DISPLAYED PC VALUE.
- E) THE INSTRUCTION THAT IMMEDIATELY PRECEDES THE ONE REFERENCED BY THE DISPLAYED PC VALUE IS THE INSTRUCTION THAT WAS/WAS BEING

EXECUTED WHEN THE FAILURE OCCURRED.

AN UNCONDITIONAL ERROR HALT FAILURE IS AN ABNORMAL CONDITION INDICATING A HARDWARE FAILURE, OR MOST UNLIKELY, A PROGRAM FAILURE. THIS PROGRAM ASSUMES THAT THE PROCESSOR IS IN OPERATING CONDITION IN ORDER TO PERFORM ITS TESTS. ANY FURTHER STEPS REQUIRED TO DIAGNOSE AN UNCONDITIONAL ERROR HALT ARE NOT WITHIN THE SCOPE OF THIS PROGRAM.

3.2 ERROR PRINTOUTS

THERE ARE 2 TYPES OF ERROR PRINTOUTS, NORMAL ERROR PRINTOUTS AND FATAL ERROR PRINTOUTS. EACH TYPE IS GENERATED BY THE SYSMAC .\$ERROR SUBROUTINE. THE '\$ERROR' SUBROUTINE IS CALLED BY AN 'ERROR NN(TRAP+N)' STATEMENT IN THE PROGRAM LISTING. A NORMAL ERROR PRINTOUT LOOKS AS FOLLOWS:

```
PC      SP      PS      TEST      TCCM      TCST      ADDITIONAL INFO
XXXXXX XXXXX/ XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
```

WHERE:

PC
XXXXXX IS THE ADDRESS OF THE ERROR CALL

SP
XXXXXX IS THE VALUE OF THE STACK POINTER

PS
XXXXXX IS THE VALUE OF THE PROCESSOR STATUS WORD

TEST
XXXXXX IS THE NUMBER OF THE FAILING ROUTINE

TCCM
XXXXXX IS THE VALUE OF THE DECTAPE COMMAND REGISTER

TCST
XXXXXX IS THE CONTENTS OF THE DECTAPE STATUS REGISTER

ADDITIONAL INFORMATION CAN VARY FROM TEST TO TEST AND FURTHER DESCRIBES THE ERROR. AFTER THE PRINTOUT IS COMPLETED, THE PROGRAM WILL HALT AT COMMON ERROR HALT IF SR15 IS SET. WHEN AN ERROR PRINTOUT OCCURS:

- A) LOOK UP THE ADDRESS REFERENCED BY PC OYYYYY IN THE LISTING.
- B) OPPOSITE THE PC VALUE AN 'ERROR' STATEMENT WILL BE FOUND, AND IN THE COMMENTS SECTION, A DESCRIPTION OF THE ERROR.
- C) AT THE BEGINNING OF THE TEST ROUTINE A DESCRIPTION OF THE TEST WILL BE FOUND.

FATAL ERRORS ARE UNEXPECTED TRAPS TO EITHER LOCATION 4 OR TO LOCATION 10. WHEN THESE OCCUR A FATAL ERROR MESSAGE IS PRINTED OUT IN THE FOLLOWING FORMAT.

FATAL ERROR TRAP TO LOC XX FROM LOCATION XXXXXX

WHERE X IS THE TRAP VECTOR LOCATION(4 OR 10) AND XXXXX IS THE PLACE THAT THE PROGRAM WAS EXECUTING AT WHEN THE FATAL ERROR TRAP OCCURRED. AFTER THE MESSAGE IS PRINTED THE PROGRAM ATTEMPTS TO RESTART ITSELF AT LOCATION 000200 THE STANDARD SR OPTIONS ARE DESCRIBED HERE.

SR15 HALT ON ERROR. WITH SR15 SET TO A 1, THE PROGRAM WILL HALT AFTER AN ERROR OCCURS. PRESSING CONT WILL CAUSE PROGRAM TO RESUME OPERATION.

SR14 SCOPE. THIS OPTION CAUSES THE PROGRAM TO REMAIN IN THE CURRENT TEST ROUTINE. WHEN THE OPTION IS REMOVED, THE PROGRAM WILL COMPLETE THE CURRENT ROUTINE, AND WILL THEN GO ON TO THE NEXT ROUTINE.

SR13 INHIBIT ERROR PRINTOUT. THIS OPTION IF SET, WILL REMOVE ALL ERROR PRINTOUTS.

SR11 PROGRAM TO EXECUTE EACH TEST ONLY ONCE, INSTEAD OF THE NORMAL NUMBER OF ITERATIONS SELECTED FOR EACH TEST. THIS ALLOWS FOR A 'QUICK CHECK' OF THE TC11 HARDWARE.

SR10 BELL ON ERROR. SETTING THIS SWITCH TO A 1 WILL CAUSE THE PROGRAM TO SOUND THE BELL WHEN AN ERROR IS FOUND. THIS SWITCH DOES NOT INTERFERE WITH THE FUNCTIONS OF SW15 AND SW13

SR08 SELECT ROUTINE. WITH SR8 SET, THE PROGRAM WILL RUN NORMALLY UNTIL THE ROUTINE SPECIFIED IN SR7 THROUGH SR0 IS ENCOUNTERED. THE PROGRAM WILL REMAIN LOOPING IN THE SPECIFIED ROUTINE, UNTIL EITHER SR8 IS CHANGED, OR UNTIL THE VALUE OF SWITCHES SR7 THROUGH SR0 CHANGES

SR7-SR0 TEST SELLECT. THE NUMBER SET IN THESE SWITCHES IS THE NUMBER OF THE TEST THAT WILL BE LOCKED ONTO IF SR8 IS SET IF SR8 IS SET TO A 0 THEN SR7 THROUGH SR0 HAVE NO EFFECT ON THE OPERATION OF THE PROGRAM

4.0 TESTING TC11 AT NON-STANDARD ADDRESSES AND/OR VECTORS

THIS PROGRAM CAN TEST THE TC11 AT NON-STANDARD ADDRESSES AND VECTORS PROVIDED THOSE ADDRESSES AND VECTORS ARE PROVIDED TO THE PROGRAM AS FOLLOWS:

A) AFTER LOADING PROGRAM REFER TO PROGRAM LISTING AND CHANGE LOCATIONS 001004 THROUGH 001020 TO REFLECT THE NEW TC11 ADDRESSES AND VECTORS.

B) PROCEED TO USE THE PROGRAM

ECO TABLE

CHGE1 - ADDED ROUTINE TO DISABLE CACHE

5.0 PROGRAM LISTING

```
1  
2  
3 .ABS  
4 .ENABL AMA  
5 .LIST MC,MD,BIN,LD,SEQ,ME  
6 .NLIST CND  
7 167400 $SWR=167400  
8 000000 $TN=0  
9 .ENABL ABS  
10 .MCALL .HEADER,.$SCATCH,.$SEOP,.$EQUAT  
11 .MCALL .SWRHI,.$SWRLO,.$SCOPE,.$SETUP  
12 .MCALL .$TYPOCT,.$TYPDEC,.$STRAP,.$POWER  
13 .MCALL .$ERROR,.$TYPE,STARS,.$ERRTYP  
14 .MCALL .SCMTAG  
15 .SETUP <.$SCOPE,.$SEOP,.$POWER,.$STRAP,.$ERROR>  
16 .LIST  
17 000000 .HEADER <CZTCBEO TC2-TC11 TEST #2>,<1972,1979>,<J. COMEAU>  
18 .TITLE CZTCBEO TC2-TC11 TEST #2  
19 ;*COPYRIGHT (C) 1972,1979  
20 ;*DIGITAL EQUIPMENT CORP.  
21 ;*MAYNARD, MASS. 01754  
22 ;*  
23 ;*PROGRAM BY J. COMEAU  
24 ;*  
25 ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC  
26 ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.  
27 000000 ;*  
28 .SWRHI  
29 .SBTTL OPERATIONAL SWITCH SETTINGS  
30 ;*  
31 ;* SWITCH USE  
32 ;* -----  
33 ;* 15 HALT ON ERROR  
34 ;* 14 LOOP ON TEST  
35 ;* 13 INHIBIT ERROR TYPEOUTS  
36 ;* 11 INHIBIT ITERATIONS  
37 ;* 10 BELL ON ERROR  
38 ;* 9 LOOP ON ERROR  
39 .LIST  
40 ;* 8 LOOP ON TEST IN SWR<7:0>  
41 .MACRO .SWRLO S07,S06,S05,S04,S03,S02,S01,S00  
42 .IIF NB <S07>.* 7 S07  
43 .IIF NB <S06>.* 6 S06  
44 .IIF NB <S05>.* 5 S05  
45 .IIF NB <S04>.* 4 S04  
46 .IIF NB <S03>.* 3 S03  
47 .IIF NB <S02>.* 2 S02  
48 .IIF NB <S01>.* 1 S01  
49 .IIF NB <S00>.* 0 S00  
50 .ENDM .SWRLO  
51 000000 ;* 7-0 # OF TEST TO LOOP ON IF SWR<8> IS SET  
52 .EQUAT  
53 .SBTTL BASIC DEFINITIONS  
54 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***  
55 001100 STACK= 1100  
56 .EQUIV EMT,ERROR ;;BASIC DEFINITION OF ERROR CALL
```

```
57 .EQUIV IOT,SCOPE ;;BASIC DEFINITION OF SCOPE CALL
58
59 ;*MISCELLANEOUS DEFINITIONS
60 000011 HT= 11 ;;CODE FOR HORIZONTAL TAB
61 000012 LF= 12 ;;CODE FOR LINE FEED
62 000015 CR= 15 ;;CODE FOR CARRIAGE RETURN
63 000200 CRLF= 200 ;;CODE FOR CARRIAGE RETURN-LINE FEED
64 177776 PS= 177776 ;;PROCESSOR STATUS WORD
65 .EQUIV PS,PSW
66 177774 STKLM= 177774 ;;STACK LIMIT REGISTER
67 177772 PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER
68 177570 DSWR= 177570 ;;HARDWARE SWITCH REGISTER
69 177570 DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
70
71 ;*GENERAL PURPOSE REGISTER DEFINITIONS
72 000000 R0= %0 ;;GENERAL REGISTER
73 000001 R1= %1 ;;GENERAL REGISTER
74 000002 R2= %2 ;;GENERAL REGISTER
75 000003 R3= %3 ;;GENERAL REGISTER
76 000004 R4= %4 ;;GENERAL REGISTER
77 000005 R5= %5 ;;GENERAL REGISTER
78 000006 R6= %6 ;;GENERAL REGISTER
79 000007 R7= %7 ;;GENERAL REGISTER
80 000006 SP= %6 ;;STACK POINTER
81 000007 PC= %7 ;;PROGRAM COUNTER
82
83 ;*PRIORITY LEVEL DEFINITIONS
84 000000 PR0= 0 ;;PRIORITY LEVEL 0
85 000040 PR1= 40 ;;PRIORITY LEVEL 1
86 000100 PR2= 100 ;;PRIORITY LEVEL 2
87 000140 PR3= 140 ;;PRIORITY LEVEL 3
88 000200 PR4= 200 ;;PRIORITY LEVEL 4
89 000240 PR5= 240 ;;PRIORITY LEVEL 5
90 000300 PR6= 300 ;;PRIORITY LEVEL 6
91 000340 PR7= 340 ;;PRIORITY LEVEL 7
92
93 ;*'SWITCH REGISTER' SWITCH DEFINITIONS
94 100000 SW15= 100000
95 040000 SW14= 40000
96 020000 SW13= 20000
97 010000 SW12= 10000
98 004000 SW11= 4000
99 002000 SW10= 2000
100 001000 SW09= 1000
101 000400 SW08= 400
102 000200 SW07= 200
103 000100 SW06= 100
104 000040 SW05= 40
105 000020 SW04= 20
106 000010 SW03= 10
107 000004 SW02= 4
108 000002 SW01= 2
109 000001 SW00= 1
110 .EQUIV SW09,SW9
111 .EQUIV SW08,SW8
112 .EQUIV SW07,SW7
```



```
113 .EQUIV SW06,SW6
114 .EQUIV SW05,SW5
115 .EQUIV SW04,SW4
116 .EQUIV SW03,SW3
117 .EQUIV SW02,SW2
118 .EQUIV SW01,SW1
119 .EQUIV SW00,SW0
120
121 ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
122 100000 BIT15= 100000
123 040000 BIT14= 40000
124 020000 BIT13= 20000
125 010000 BIT12= 10000
126 004000 BIT11= 4000
127 002000 BIT10= 2000
128 001000 BIT09= 1000
129 000400 BIT08= 400
130 000200 BIT07= 200
131 000100 BIT06= 100
132 000040 BIT05= 40
133 000020 BIT04= 20
134 000010 BIT03= 10
135 000004 BIT02= 4
136 000002 BIT01= 2
137 000001 BIT00= 1
138 .EQUIV BIT09,BIT9
139 .EQUIV BIT08,BIT8
140 .EQUIV BIT07,BIT7
141 .EQUIV BIT06,BIT6
142 .EQUIV BIT05,BIT5
143 .EQUIV BIT04,BIT4
144 .EQUIV BIT03,BIT3
145 .EQUIV BIT02,BIT2
146 .EQUIV BIT01,BIT1
147 .EQUIV BIT00,BIT0
148
149 ;*BASIC 'CPU' TRAP VECTOR ADDRESSES
150 000004 ERRVEC= 4 ;:TIME OUT AND OTHER ERRORS
151 000010 RESVEC= 10 ;:RESERVED AND ILLEGAL INSTRUCTIONS
152 000014 TBITVEC=14 ;: 'T' BIT
153 000014 TRTVEC= 14 ;:TRACE TRAP
154 000014 BPTVEC= 14 ;:BREAKPOINT TRAP (BPT)
155 000020 IOTVEC= 20 ;:INPUT/OUTPUT TRAP (IOT) **SCOPE**
156 000024 PWRVEC= 24 ;:POWER FAIL
157 000030 EMTVEC= 30 ;:EMULATOR TRAP (EMT) **ERROR**
158 000034 TRAPVEC=34 ;: 'TRAP' TRAP
159 000060 TKVEC= 60 ;:TTY KEYBOARD VECTOR
160 000064 TPVEC= 64 ;:TTY PRINTER VECTOR
161 000240 PIRQVEC=240 ;:PROGRAM INTERRUPT REQUEST VECTOR
162 .LIST
163 ;MISCELLANEOUS EQUATES
164 000000 .SCATCH START
165 .SBTTL TRAP CATCHER
166
167 000000 .=0
168 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A '+2,HALT'
```

```
169 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
170 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
171 .LIST
172      000174
173 000174 000000
174 000176 000000
175
176 000200 000137 002314
177
178      001000
179      000240
180      000000
181      100000
182      005746
183      024646
184      005726
185      022626
186      000007
187      177777
188      000003
189      000207
190      000040
191      177777
192      100000
193      040000
194      020000
195      000000
196      000004
197      000010
198      000014
199      000020
200      000024
201      000030
202      000034
203      020000
204      010000
205      004000
206      000000
207 000204 000240
208      000000
209      000400
210      001000
211      001400
212      002000
213      002400
214      003000
215      003400
216      000100
217      000000
218      000002
219      000004
220      000006
221      000010
222      000012
223      000014
224      000016

      .=174
DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0          ;;SOFTWARE SWITCH REGISTER
.SBTTL   STARTING ADDRESS(ES)
      JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM

;EQUATES
      SPBOT=1000
      NOP=240
      OPEN=0
      MANUAL=BIT15
      PUSH=005746
      PUSH2=024646
      POPSP=005726
      POPSP2=022626
      BELL=007
      TLAST=-1
      TRC=3
      RTSPC=207
      I=40
      X=-1
      A=BIT15
      B=BIT14
      C=BIT13
      V0=0
      V1=4
      V2=10
      V3=14
      V4=20
      V5=24
      V6=30
      V7=34
      MAINT=BIT13
      DINH=BIT12
      REV=BIT11
      FWD=0
      NOP
      U0=0
      U1=BIT8
      U2=BIT9
      U3=BIT9!BIT8
      U4=BIT10
      U5=BIT10!BIT8
      U6=BIT10!BIT9
      U7=BIT10!BIT9!BIT8
      IE=BIT6
      SAT=0
      RNUM=BIT1
      RDATA=BIT2
      RALL=BIT2!BIT1
      SST=BIT3
      WRTM=BIT3!BIT1
      WDATA=BIT3!BIT2
      WALL=BIT3!BIT2!BIT1
```

```

225          000001          DO=BIT0
226          000000          EMTX=0
227          .MACRO          ADITAG
228          TCST:          177340          ;TC11 STATUS REGISTER.
229          TCCM:          177342          ;TC11 COMMAND REGISTER.
230          TCWC:          177344          ;TC11 WORD COUNT REGISTER.
231          TCBA:          177346          ;TC11 BUS ADDRESS REGISTER.
232          TCDT:          177350          ;TC11 DATA REGISTER.
233          TCVTR:          214           ;TC11 INTERRUPT VECTOR
234          TCLVL:          300          ;TC11 INTERRUPT PRIORITY LEVEL.
235          TPS:           177564          ;LSP CSR
236          TPB:           177566          ;LSP BUFFER
237          CODCAL:        OPEN
238          RTNNO:        OPEN
239          NXTST:        OPEN
240          CURTST:        OPEN
241          CRBUF:        OPEN
242          CRBUFA:        OPEN
243          CTPA:          OPEN
244          SBDAT1:        50505
245                      127272
246          SBDAT2:        72727
247                      105050
248          SBDAT3:        72727
249                      105056
250          POWPUS:        .WORD          000000
251          POWPOP:        .WORD          000000
252          TCCMT:        OPEN
253          TCSTT:        OPEN
254          CCR:           177746          ;CACHE CONTROL REGISTER
255          .ENDM          ADITAG
256
257          ;
258          .MACRO          SETRAP
259          MOV             #TRAP0,a#4          ;SETUP FATAL TRAP VECTOR JUST IN CASE
260          MOV             #340,a#6          ;NO INTERRUPTS WHILE SERVICING FATAL ERRORS
261          .ENDM          SETRAP
262          .MACR          C55
263          .BYTE          1,0,1,1,0,1          ;MTK CODE 55. REV END ZONE MARK.
264          .ENDM
265          .MACR          C25
266          .BYTE          0,1,0,1,0,1          ;MTK CODE 25. EXTENSION MARK.
267          .ENDM
268          .MACR          C26          B0,B1,B2,B3,B4,B5
269          .BYTE          0!B0,I!B1,0!B2,I!B3,I!B4,0!B5          ;FWD BLOCK MARK.
270          .ENDM
271          .MACR          C32          B0,B1,B2,B3,B4,B5
272          .BYTE          0!B0,I!B1,I!B2,0!B3,I!B4,0!B5          ;REV GUARD.
273          .ENDM
274          .MACR          C10          B0,B1,B2,B3,B4,B5
275          .BYTE          0!B0,0!B1,I!B2,0!B3,0!B4,0!B5          ;MTK CODE 10.
276          .ENDM
277          .MACR          C70          B0,B1,B2,B3,B4,B5
278          .BYTE          I!B0,I!B1,I!B2,0!B3,0!B4,0!B5          ;MTK CODE 70. DATA MARK.
279          .ENDM
280          .MACR          C73          B0,B1,B2,B3,B4,B5

```

```
281 .BYTE I!B0,I!B1,I!B2,0!B3,I!B4,I.B5 ;MTK CODE 73. DATA MARK.
282 .ENDM
283 .MACR C51 B0,B1,B2,B3,B4,B5
284 .BYTE I!B0,0!B1,I!B2,0!B3,0!B4,I!B5 ;MTK CODE 51. FWD GUARD.
285 .ENDM
286 .MACR C45 B0,B1,B2,B3,B4,B5
287 .BYTE I!B0,0!B1,0!B2,I!B3,0!B4,I!B5 ;MTK CODE 45. REV BLOCK MARK.
288 .ENDM
289 .MACR C22
290 .BYTE 0,I,0,0,I,0 ;MTK CODE 22. FWD END ZONE.
291 .ENDM
292 .MACR CEND
293 .BYTE -1
294 .ENDM
295 .MACR EMTE
296 .BYTE I,I,I,0,0,I
297 .ENDM
298 .MACR MTCOD MTADR,CNT
299 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
300 MTADR ;ADDRESS OF MARK TRACK CODES.
301 CNT ;MARK TRACK CODE COUNT.
302 .ENDM
303 .MACR MTCOE CALADR,MTADR,CNT
304 JSR R5,LMTCOE ;CALL LOAD MT CODES SUBROUTINE.
305 CALADR ;ADDR TO GO AFTER EACH CODE PASSED.
306 MTADR ;ADDRESS OF MARK TRACK CODES.
307 CNT ;MARK TRACK CODE COUNT.
308 .ENDM
309 .MACR EMTDEF NAMEA,NAMEB
310 .WORD NAMEB ;POINTER FOR EMT CALL NAMEA
311 .NLIST
312 NAMEA=EMT+EMTX
313 EMTX=EMTX+1
314 .LIST
315 .ENDM
316 .MACRO SCOMAC
317 CLR @TCCM
318 CLR 2(SP) ;PS TO =0 AFTER WE EXIT THE SCOPE ROUTINE
319 JSR PC,SRSETT
320 JSR PC,RSTMTK
321 .ENDM SCOMAC
322 000050 .=50
323 000050 000000 .WORD 0
324 000052 000000 .WORD 0
325 000054 .SCMTAG 10,10,ADITAG,1100
326 .MACRO $$CMREG A,B
327 $REG'A: .WORD 0 ;;CONTAINS (($REGAD)+'B)
328 .NLIST
329 $CM1=$CM1+1
330 $CM2=$CM2+2
331 .LIST
332 .ENDM $$CMREG
333 .MACRO $$CMTMP A
334 $TMP'A: .WORD 0 ;;USER DEFINED
335 .NLIST
336 $CM4=$CM4+1
```

CZTCBEO TC2-TC11 TEST #2
CZTCBE.P11 10-APR-80 15:17

MACY11 30A(1052) 10-APR-80 15:18 M 1
STARTING ADDRESS(ES) PAGE 8

SEQ 0012

337
338
339

.LIST
.ENDM \$\$CMTMP
.PAGE


```
340 .SBTTL COMMON TAGS
341
342 000054 STARS
343 ::*****
344 ::*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
345 ::*USED IN THE PROGRAM.
346
347 001100 . =1100
348 001100 $CMTAG: .START OF COMMON TAGS
349 001100 000000 $PASS: .WORD 0 ::CONTAINS PASS COUNT
350 001102 000 $STNM: .BYTE 0 ::CONTAINS THE TEST NUMBER
351 001103 000 $ERFLG: .BYTE 0 ::CONTAINS ERROR FLAG
352 001104 000000 $ICNT: .WORD 0 ::CONTAINS SUBTEST ITERATION COUNT
353 001106 000000 $LPADR: .WORD 0 ::CONTAINS SCOPE LOOP ADDRESS
354 001110 000000 $LPERR: .WORD 0 ::CONTAINS SCOPE RETURN FOR ERRORS
355 001112 000000 $ERTTL: .WORD 0 ::CONTAINS TOTAL ERRORS DETECTED
356 001114 000 $ITEMB: .BYTE 0 ::CONTAINS ITEM CONTROL BYTE
357 001115 001 $ERMAX: .BYTE 1 ::CONTAINS MAX. ERRORS PER TEST
358 001116 000000 $ERRPC: .WORD 0 ::CONTAINS PC OF LAST ERROR INSTRUCTION
359 001120 000000 $GDADR: .WORD 0 ::CONTAINS ADDRESS OF 'GOOD' DATA
360 001122 000000 $BDADR: .WORD 0 ::CONTAINS ADDRESS OF 'BAD' DATA
361 001124 000000 $GDDAT: .WORD 0 ::CONTAINS 'GOOD' DATA
362 001126 000000 $BDDAT: .WORD 0 ::CONTAINS 'BAD' DATA
363 001130 000000 .WORD 0 ::RESERVED--NOT TO BE USED
364 001132 000000 .WORD 0
365 001134 000 $AUTOB: .BYTE 0 ::AUTOMATIC MODE INDICATOR
366 001135 000 $INTAG: .BYTE 0 ::INTERRUPT MODE INDICATOR
367 001136 000000 .WORD 0
368 001140 177570 $SWR: .WORD DSWR ::ADDRESS OF SWITCH REGISTER
369 001142 177570 $DISPLAY: .WORD DDISP ::ADDRESS OF DISPLAY REGISTER
370 001144 177560 $TKS: 177560 ::TTY KBD STATUS
371 001146 177562 $TKB: 177562 ::TTY KBD BUFFER
372 001150 177564 $TPS: 177564 ::TTY PRINTER STATUS REG. ADDRESS
373 001152 177566 $TPB: 177566 ::TTY PRINTER BUFFER REG. ADDRESS
374 001154 000 $NULL: .BYTE 0 ::CONTAINS NULL CHARACTER FOR FILLS
375 001155 002 $FILLS: .BYTE 2 ::CONTAINS # OF FILLER CHARACTERS REQUIRED
376 001156 012 $FILLC: .BYTE 12 ::INSERT FILL CHARS. AFTER A 'LINE FEED'
377 001157 000 $TPFLG: .BYTE 0 ::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
378 .LIST
379 001160 000000 $REGAD: .WORD 0 ::CONTAINS THE ADDRESS FROM
380 WHICH ($REGO) WAS OBTAINED
381 .LIST
382 000010 .REPT $CM3
383 $$CMREG \SCM1,\SCM2
384 .ENDR
385 001162 $$CMREG \SCM1,\SCM2
386 001162 000000 $REGO: .WORD 0 ::CONTAINS (($REGAD)+0)
387 .LIST
388 001164 $$CMREG \SCM1,\SCM2
389 001164 000000 $REG1: .WORD 0 ::CONTAINS (($REGAD)+2)
390 .LIST
391 001166 $$CMREG \SCM1,\SCM2
392 001166 000000 $REG2: .WORD 0 ::CONTAINS (($REGAD)+4)
393 .LIST
394 001170 $$CMREG \SCM1,\SCM2
395 001170 000000 $REG3: .WORD 0 ::CONTAINS (($REGAD)+6)
```

```
396 .LIST
397 001172
398 001172 000000 $REG4: .WORD 0 ;;CONTAINS (($REGAD)+10)
399 .LIST
400 001174
401 001174 000000 $REG5: .WORD 0 ;;CONTAINS (($REGAD)+12)
402 .LIST
403 001176
404 001176 000000 $REG6: .WORD 0 ;;CONTAINS (($REGAD)+14)
405 .LIST
406 001200
407 001200 000000 $REG7: .WORD 0 ;;CONTAINS (($REGAD)+16)
408 .LIST
409 .LIST
410 000010 .REPT 10
411 $SCMTMP \SCM4
412 .ENDR
413 001202
414 001202 000000 $SCMTMP \SCM4
415 .LIST STMP0: .WORD 0 ;;USER DEFINED
416 001204
417 001204 000000 $SCMTMP \SCM4
418 .LIST STMP1: .WORD 0 ;;USER DEFINED
419 001206
420 001206 000000 $SCMTMP \SCM4
421 .LIST STMP2: .WORD 0 ;;USER DEFINED
422 001210
423 001210 000000 $SCMTMP \SCM4
424 .LIST STMP3: .WORD 0 ;;USER DEFINED
425 001212
426 001212 000000 $SCMTMP \SCM4
427 .LIST STMP4: .WORD 0 ;;USER DEFINED
428 001214
429 001214 000000 $SCMTMP \SCM4
430 .LIST STMP5: .WORD 0 ;;USER DEFINED
431 001216
432 001216 000000 $SCMTMP \SCM4
433 .LIST STMP6: .WORD 0 ;;USER DEFINED
434 001220
435 001220 000000 $SCMTMP \SCM4
436 .LIST STMP7: .WORD 0 ;;USER DEFINED
437 001222 000000 STIMES: 0 ;;MAX. NUMBER OF ITERATIONS
438 001224 000000 $ESCAPE: 0 ;;ESCAPE ON ERROR ADDRESS
439 001226 177607 000377 $BELL: .ASCIZ <207><377><377> ;;CODE FOR BELL
440 001232 077 $QUES: .ASCII /?/ ;;QUESTION MARK
441 001233 015 $CRLF: .ASCII <15> ;;CARRIAGE RETURN
442 001234 000012 $LF: .ASCIZ <12> ;;LINE FEED
443 001236 STARS
444 ;:*****
445 .IRP A,<ADITAG>
446 A
447 .ENDM
448 001236 ADITAG
449 001236 177340 TCST: 177340 ;;TC11 STATUS REGISTER.
450 001240 177342 TCCM: 177342 ;;TC11 COMMAND REGISTER.
451 001242 177344 TCWC: 177344 ;;TC11 WORD COUNT REGISTER.
```

452	001244	177346	TCBA:	177346		:TC11 BUS ADDRESS REGISTER.
453	001246	177350	TCDT:	177350		:TC11 DATA REGISTER.
454	001250	000214	TCVTR:	214		:TC11 INTERRUPT VECTOR
455	001252	000300	TCLVL:	300		:TC11 INTEPRUPT PRIORITY LEVEL.
456	001254	177564	TPS:	177564		:LSP CSR
457	001256	177566	TPB:	177566		:LSP BUFFER
458	001260	000000	CODCAL:	OPEN		
459	001262	000000	RTNNO:	OPEN		
460	001264	000000	NXTST:	OPEN		
461	001266	000000	CURTST:	OPEN		
462	001270	000000	CRBUF:	OPEN		
463	001272	000000	CRBUFA:	OPEN		
464	001274	000000	CTRA:	OPEN		
465	001276	050505	SBDAT1:	50505		
466	001300	127272		127272		
467	001302	072727	SBDAT2:	72727		
468	001304	105050		105050		
469	001306	072727	SBDAT3:	72727		
470	001310	105056		105056		
471	001312	000000	POWPUS:	.WORD	000000	
472	001314	000000	POWPOP:	.WORD	000000	
473	001316	000000	TCCMT:	OPEN		
474	001320	000000	TCSTT:	OPEN		
475	001322	177746	CCR:	177746		:CACHE CONTROL REGISTER
476			.PAGE			

477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532

.SBTTL ERROR POINTER TABLE

:*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
:*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
:*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
:*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
:*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

::* EM ::POINTS TO THE ERROR MESSAGE
::* DH ::POINTS TO THE DATA HEADER
::* DT ::POINTS TO THE DATA
::* DF ::POINTS TO THE DATA FORMAT

\$ERRTB:

EM1	::'SAT (STOP ALL TRANSPORTS) COMMAND DID NOT CLEAR READY''
EH1	::' PC SP PS TEST# TCCM TCST''
ET1	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	
EM2	::'SST (STOP SELECTED TRANSPORT) DID NOT CLEAR READY''
EH2	::' PC SP PS TEST# TCCM TCST''
ET2	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	
EM3	::'READY BIT DID NOT CAUSE AN INTERRUPT''
EH3	::' PC SP PS TEST# TCCM TCST''
ET3	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	
EM4	::'READY BIT CAUSED AN INTERRUPT WITH PROCESSOR AND TC11 AT SAME PRIORITY
EH4	::' PC SP PS TEST# TCCM TCST''
ET4	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	
EM5	::'TC11 FAILED TO INTERRUPT''
EH5	::' PC SP PS TEST# TCCM TCST''
ET5	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	
EM6	::'TC11 DID NOT DROP INTERRUPT REQUEST AFTER IT WAS ACKNOWLEDGED''
EH6	::' PC SP PS TEST# TCCM TCST''
ET6	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	
EM7	::'DOING A RESET INSTRUCTION DID NOT CLEAR UPS''
EH7	::' PC SP PS TEST# TCCM TCST''
ET7	::\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
000000	

001324
001324 016237
001326 016325
001330 016404
001332 000000
001334 016422
001336 016504
001340 016562
001342 000000
001344 016600
001346 016645
001350 016724
001352 000000
001354 016742
001356 017051
001360 017130
001362 000000
001364 017146
001366 017177
001370 017256
001372 000000
001374 017274
001376 017371
001400 017450
001402 000000
001404 017466
001406 017542
001410 017620
001412 000000

589	001524	021562	EM21	;'ILO ERROR SETTING DID NOT CAUSE THE 'ERROR' BIT TO SET''
590	001526	021651	EH21	;' PC SP PS TEST# TCCM TCST''
591	001530	021730	ET21	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
592	001532	000000	000000	
593				
594				
595	001534	021746	EM22	;'CLEARING ERROR BIT ALSO CLEARED ILO ERROR''
596	001536	022020	EH22	;' PC SP PS TEST# TCCM TCST''
597	001540	022076	ET22	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
598	001542	000000	000000	
599				
600				
601	001544	022114	EM23	;'THE 'ERROR' BIT DID NOT SET''
602	001546	022150	EH23	;' PC SP PS TEST# TCCM TCST''
603	001550	022226	ET23	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
604	001552	000000	000000	
605				
606				
607	001554	022244	EM24	;'THE 'ERROR' BIT SET DID NOT CAUSE AN INTERRUPT''
608	001556	022323	EH24	;' PC SP PS TEST# TCCM TCST''
609	001560	022402	ET24	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
610	001562	000000	000000	
611				
612				
613	001564	022420	EM25	;'DOING A RESET INSTRUCTION DID NOT SET THE READY BIT'
614	001566	022504	EH25	;' PC SP PS TEST# TCCM TCST''
615	001570	022562	ET25	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
616	001572	000000	000000	
617				
618				
619	001574	022600	EM26	;'TEST EXECUTION IS OUT OF ORDER''
620	001576	022637	EH26	;' PC SP PS TEST# TEST# S/B''
621	001600	022712	ET26	;\$ERRPC,\$REG6,\$REG7,\$REG5,TEST# S/B
622	001602	000000	000000	
623				
624				
625	001604	022726	EM27	;'ERROR TRYING TO READ A BLOCK MARK''
626	001606	022770	EH27	;' PC SP PS TEST# TCCM TCST''
627	001610	023046	ET27	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
628	001612	000000	000000	
629				
630				
631	001614	023064	EM30	;'READY WAS NOT SET AFTER BLOCK MARK WAS SHIFTED INTO THE WINDOW REGISTE
632	001616	023174	EH30	;' PC SP PS TEST# TCCM TCST''
633	001620	023252	ET30	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
634	001622	000000	000000	
635				
636				
637	001624	023270	EM31	;'INCORRECT BLOCK # IN DATA REG AFTER BLOCK MARK WAS DETECTED''
638	001626	023364	EH31	;' PC SP PS TEST# TCCM TCST''
639	001630	023464	ET31	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
640	001632	000000	000000	
641				
642				
643	001634	023506	EM32	;'MTE WAS NOT SET BY AN ILLEGAL MARK TRACK CODE''
644	001636	023564	EH32	;' PC SP PS TEST# TCCM TCST''

701	001754	025436	EM44	;'WORD COUNT INCREMENTED IMPROPERLY''					
702	001756	025500	EH44	;' PC SP PS TEST# TCCM TCST TCWC''					
703	001760	025602	ET44	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,TCWC					
704	001762	000000							
705									
706									
707	001764	025624	EM45	;'TCBA INCREMENTED IMPROPERLY''					
708	001766	025660	EH45	;' PC SP PS TEST# TCCM TCST TCBA''					
709	001770	025762	ET45	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,TCBA					
710	001772	000000							
711									
712									
713	001774	026004	EM46	;'PARITY ERROR''					
714	001776	026021	EH46	;' PC SP PS TEST# TCCM TCST''					
715	002000	026100	ET46	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
716	002002	000000							
717									
718									
719	002004	026116	EM47	;'READY DID NOT SET AFTER READING WAS COMPLETED''					
720	002006	026174	EH47	;' PC SP PS TEST# TCCM TCST''					
721	002010	026252	ET47	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
722	002012	000000							
723									
724									
725	002014	026270	EM50	;'TRANSFERED TOO MANY WORDS''					
726	002016	026322	EH50	;' PC SP PS TEST# TCCM TCST RBUF+2''					
727	002020	026412	ET50	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,RBUF+2					
728	002022	000000							
729									
730									
731	002024	026432	EM51	;'TCBA CONTAINS AN INCORRECT ADDRESS''					
732	002026	026475	EH51	;' PC SP PS TEST# TCCM TCST TCBA TCBA S/B''					
733	002030	026576	ET51	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,TCBA TCBA S/B					
734	002032	000000							
735									
736									
737	002034	026620	EM52	;'PARRITY ERROR WAS NOT DETECTED''					
738	002036	026657	EH52	;' PC SP PS TEST# TCCM TCST''					
739	002040	026746	ET52	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
740	002042	000000							
741									
742									
743	002044	026766	EM53	;'PARITY ERROR DID NOT SET THE 'ERROR' BIT''					
744	002046	027037	EH53	;' PC SP PS TEST# TCCM TCST''					
745	002050	027126	ET53	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
746	002052	000000							
747									
748									
749	002054	027146	EM54	;'PARITY ERROR BIT WILL NOT CLEAR''					
750	002056	027206	EH54	;' PC SP PS TEST# TCCM TCST''					
751	002060	027274	ET54	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
752	002062	000000							
753									
754									
755	002064	027314	EM55	;'BLOCK MISS SHOULD NOT HAVE SET''					
756	002066	027353	EH55	;' PC SP PS TEST# TCCM TCST''					

757	002070	027442	ET55	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
758	002072	000000	000000	
759				
760				
761	002074	027462	EM56	;'RDATA WAS ISSUED BUT BLOCK MISS FAILED TO SET''
762	002076	027540	EH56	;' PC SP PS TEST# TCCM TCST''
763	002100	027626	ET56	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
764	002102	000000	000000	
765				
766				
767	002104	027646	EM57	;'BLOCK MISS SETTING DID NOT SET THE 'ERROR' BIT''
768	002106	027725	EH57	;' PC SP PS TEST# TCCM TCST''
769	002110	030014	ET57	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
770	002112	000000	000000	
771				
772				
773	002114	030034	EM60	;'CLEARING ERROR BIT FAILED TO CLEAR BLOCK MISS''
774	002116	030112	EH60	;' PC SP PS TEST# TCCM TCST''
775	002120	030200	ET60	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
776	002122	000000	000000	
777				
778				
779	002124	030220	EM61	;'FORWARD CHECKSUM WAS WRITTEN INCORRECTLY INTO CORE''
780	002126	030303	EH61	;' PC SP PS TEST# TCCM TCST RBUF+514''
781	002130	030402	ET61	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,RBUF+514
782	002132	000000	000000	
783				
784				
785	002134	030424	EM62	;'TCWC WAS MODIFIED DURING RAL''
786	002136	030462	EH62	;' PC SP PS TEST# TCCM TCST TCWC''
787	002140	030550	ET62	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,TCWC
788	002142	000000	000000	
789				
790				
791	002144	030570	EM63	;'TCBA WAS MODIFIED DURING RAL''
792	002146	030626	EH63	;' PC SP PS TEST# TCCM TCST TCBA''
793	002150	030730	ET63	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,TCBA
794	002152	000000	000000	
795				
796				
797	002154	030750	EM64	;'DATA MISS DID NOT SET''
798	002156	030776	EH64	;' PC SP PS TEST# TCCM TCST''
799	002160	031064	ET64	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
800	002162	000000	000000	
801				
802				
803	002164	031104	EM65	;'DATA MISS SETTING DID NOT CAUSE THE 'ERROR' BIT TO SET''
804	002166	031173	EH65	;' PC SP PS TEST# TCCM TCST''
805	002170	031252	ET65	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
806	002172	000000	000000	
807				
808				
809	002174	031270	EM66	;'CLEARING THE 'ERROR' BIT DID NOT CAUSE DATA MISS TO BE CLEARED''
810	002176	031367	EH66	;' PC SP PS TEST# TCCM TCST''
811	002200	031446	ET66	;\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
812	002202	000000	000000	


```

869      .SBTTL INITIALIZE THE COMMON TAGS
870      ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
871 002324 012706 001100      MOV    # $CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
872 002330 005026              CLR    (R6)+            ;;CLEAR MEMORY LOCATION
873 002332 022706 001140      CMP    #SWR,R6 ;;DONE?
874 002336 001374              BNE    -6              ;;LOOP BACK IF NO
875 002340 012706 001000      MOV    #1000,SP      ;;SETUP THE STACK POINTER
876      ;;INITIALIZE A FEW VECTORS
877 002344 012737 013362 000020  MOV    # $SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
878 002352 012737 000340 000022  MOV    #340,@IOTVEC+2 ;;LEVEL 7
879 002360 012737 013652 000030  MOV    # $ERROR,@EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
880 002366 012737 000340 000032  MOV    #340,@EMTVEC+2 ;;LEVEL 7
881 002374 012737 015314 000034  MOV    # $TRAP,@TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
882 002402 012737 000340 000036  MOV    #340,@TRAPVEC+2;LEVEL 7
883 002410 012737 014074 000024  MOV    # $PWRDN,@PWRVEC ;;POWER FAILURE VECTOR
884 002416 012737 000340 000026  MOV    #340,@PWRVEC+2 ;;LEVEL 7
885 002424 013737 011604 011576  MOV    $ENDCT,$EOPCT  ;;SETUP END-OF-PROGRAM COUNTER
886 002432 005037 001222              CLR    $TIMES        ;;INITIALIZE NUMBER OF ITERATIONS
887 002436 005037 001224              CLR    $ESCAPE      ;;CLEAR THE ESCAPE ON ERROR ADDRESS
888 002442 112737 000001 001115  MOV    #1,$ERMAX     ;;ALLOW ONE ERROR PER TEST
889 002450 012737 002450 001106  MOV    #.,$SLPADR    ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
890 002456 012737 002456 001110  MOV    #.,$SLPERR    ;;SETUP THE ERROR LOOP ADDRESS
891
892 002464      .LIST
893      SWRSU
894      ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
895      ;;EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
896 002464 013746 000004              MOV    @#ERRVEC,-(SP) ;;SAVE ERROR VECTOR
897 002470 012737 002524 000004  MOV    #64,$@#ERRVEC  ;;SET UP ERROR VECTOR
898 002476 012737 177570 001140  MOV    #DSWR,$SWR     ;;SETUP FOR A HARDWARE SWICH REGISTER
899 002504 012737 177570 001142  MOV    #DDISP,$DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
900 002512 022777 177777 176420  CMP    #-1,$SWR      ;;TRY TO REFERENCE HARDWARE SWR
901 002520 001012              BNE    66$           ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
902 002522 000403              BR     65$           ;;AND THE HARDWARE SWR IS NOT = -1
903 002524 012716 002532 64$:    MOV    #65$,(SP)    ;;BRANCH IF NO TIMEOUT
904 002530 000002              RTI                    ;;SET UP FOR TRAP RETURN
905 002532 012737 000176 001140 65$:    MOV    #SWREG,$SWR   ;;POINT TO SOFTWARE SWR
906 002540 012737 000174 001142  MOV    #DISPREG,$DISPLAY
907 002546 012637 000004 66$:    MOV    (SP)+,@#ERRVEC ;;RESTORE ERROR VECTOR
908
909 002552 012706 001000      MOV    #1000,SP      ;SET BOTTOM OF SP STACK.
910 002556 005037 001262      CLR    RTNNO
911 002562 104401 001233      TYPE  ,$CRLF
912 002566 104401 015366      TYPE  ,$TIMES       ;PRINTOUT STARTUP MESSAGE
913 002572 000240      NOP
914
915 002574 013746 000004      ;*****
916 002600 012737 002616 000004  CHGE1: MOV    @#ERRVEC,-(SP) ;SAVE ERROR VECTOR
917 002606 012777 000014 176506  MOV    #1$,@#ERRVEC  ;SET UP ERROR VECTOR
918 002614 000403              MOV    #14,@CCR     ;DISABLE CACHE, IF THERE
919 002616 012716 002624      BR     2$           ;GET AROUND INTERRUPT SERV. ROUTINE
920 002622 000002 1$:    MOV    #2$,(SP)    ;NO CACHE MEM,SETUP TO RESTORE SP
921 002624 012637 000004      RTI                    ;RESTORE SP
922 2$:    MOV    (SP)+,@#ERRVEC ;RESTORE ERROR VECTOR
923 002630 000000      ;*****
924 002632 005737 000042      HALT
      STARTX: TST    42 ;HERE IS YOUR CHANCE TO SET THE SWITCH REGISTER

```

```
925 002636 001401          BEQ      GETRDY
926 002640 000005          RESET
927 002642 005037 177776  GETRDY: CLR      PSW
928 002646 012706 001000  MOV      #1000,SP      ;SET BOTTOM OF STACK.
929 002652 004737 012220  JSR      PC,SRSETT    ;ISSUE RESET.
930 002656 004737 012242  JSR      PC,RSTMTK    ;RESTORE MARK TRACK.
931 002662 000137 002666  JMP      T0001
932
933 .SBTTL T0001
934 ;CHECK THAT THE TCCM REGISTER CAN BE ACCESSED WITHOUT A TRAP OCCURING
935 ;*****
935 002666 000004  T0001: SCOPE
936 002670 012737 002734 000004  MOV      #A0001,#4    ;SETUP THE FATAL TRAP VECTOR
937 002676 012737 000340 000006  MOV      #340,#6     ;MAKE SURE WE GET NO INTERRUPTS IF WE TRAP
938 002704 012706 001000  MOV      #1000,SP    ;SETUP THE STACK POINTER
939 002710 004737 011766  JSR      PC,TORDER   ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
940 002714 000001 00001 00001    ;HERE LIES THE NUMBER OF THIS TEST
941 002716 012706 001000  R0001: MOV      #1000,SP ;INIT THE STACK POINTER
942 002722 005777 176312  TST      @TCCM       ;TRY TO READ THE TCCM
943 002726 005077 176306  CLR      @TCCM       ;TRY TO MODIFY THE TCCM
944 002732 000401          BR       T0002       ;NO ERRORS. GO ON TO THE NEXT TEXT
945 002734 104074  A0001: ERROR 74      ;COULD NOT ACCESS TCCM
946
947 .SBTTL T0002
948 ;CHECK THAT THE TCST REGISTER CAN BE ACCESSED WITHOUT A TRAP OCCURING
949 ;*****
949 002736 000004  T0002: SCOPE
950 002740 012737 003004 000004  MOV      #A0002,#4    ;SETUP THE FATAL TRAP VECTOR
951 002746 012737 000340 000006  MOV      #340,#6     ;MAKE SURE WE GET NO INTERRUPTS IF WE TRAP
952 002754 012706 001000  MOV      #1000,SP    ;SETUP THE STACK POINTER
953 002760 004737 011766  JSR      PC,TORDER   ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
954 002764 000002 00002 00002    ;HERE LIES THE NUMBER OF THIS TEST
955 002766 012706 001000  R0002: MOV      #1000,SP ;INIT THE STACK POINTER
956 002772 005777 176240  TST      @TCST       ;TRY TO READ THE TCST
957 002776 005077 176234  CLR      @TCST       ;TRY TO MODIFY THE TCST
958 003002 000401          BR       T0003       ;NO ERRORS. GO ON TO THE NEXT TEXT
959 003004 104075  A0002: ERROR 75      ;COULD NOT ACCESS TCST
960
961 .SBTTL T0003
962 ;CHECK THAT THE TCWC REGISTER CAN BE ACCESSED WITHOUT A TRAP OCCURING
963 ;*****
963 003006 000004  T0003: SCOPE
964 003010 012737 003054 000004  MOV      #A0003,#4    ;SETUP THE FATAL TRAP VECTOR
965 003016 012737 000340 000006  MOV      #340,#6     ;MAKE SURE WE GET NO INTERRUPTS IF WE TRAP
966 003024 012706 001000  MOV      #1000,SP    ;SETUP THE STACK POINTER
967 003030 004737 011766  JSR      PC,TORDER   ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
968 003034 000003 00003 00003    ;HERE LIES THE NUMBER OF THIS TEST
969 003036 012706 001000  R0003: MOV      #1000,SP ;INIT THE STACK POINTER
970 003042 005777 176174  TST      @TCWC       ;TRY TO READ THE TCWC
971 003046 005077 176170  CLR      @TCWC       ;TRY TO MODIFY THE TCWC
972 003052 000401          BR       T0004       ;NO ERRORS. GO ON TO THE NEXT TEXT
973 003054 104076  A0003: ERROR 76      ;COULD NOT ACCESS TCWC
974
975 .SBTTL T0004
976 ;CHECK THAT THE TCBA REGISTER CAN BE ACCESSED WITHOUT A TRAP OCCURING
977 ;*****
977 003056 000004  T0004: SCOPE
978 003060 012737 003124 000004  MOV      #A0004,#4    ;SETUP THE FATAL TRAP VECTOR
979 003066 012737 000340 000006  MOV      #340,#6     ;MAKE SURE WE GET NO INTERRUPTS IF WE TRAP
980 003074 012706 001000  MOV      #1000,SP    ;SETUP THE STACK POINTER
```

```
981 003100 004737 011766      JSR    PC,TORDER      ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
982 003104 000004              00004      ;HERE LIES THE NUMBER OF THIS TEST
983 003106 012706 001000      R0004:  MOV    #1000,SP  ;INIT THE STACK POINTER
984 003112 005777 176126      TST    @TCBA          ;TRY TO READ THE TCBA
985 003116 005077 176122      CLR    @TCBA          ;TRY TO MODIFY THE TCBA
986 003122 000401              BR      T0005          ;NO ERRORS. GO ON TO THE NEXT TEXT
987 003124 104077              A0004:  ERROR  77      ;COULD NOT ACCESS TCBA
988
989
990      ;CHECK THAT ISSUING A SAT COMMAND (STOP ALL TRANSPORTS) CAUSES READY BIT
991      ;TO CLEAR IMMEDIATELY (TCCM BIT 7).
992      .SBTTL T0005
993      ;*****
993 003126 000004              T0005:  SCOPE
994 003130 012737 011702 000004      MOV    #TRAP4,4      ;SETUP FATAL TRAP VECTORS
995 003136 012737 011672 000010      MOV    #TRAP10,10
996 003144 012737 000340 000006      MOV    #340,6
997 003152 012737 000340 000012      MOV    #340,12
998 003160 012706 001000              MOV    #1000,SP      ;SETUP THE STACK POINTER
999 003164 004737 011766      JSR    PC,TORDER      ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1000 003170 000005              00005      ;HERE LIES THE NUMBER OF THIS TEST
1001 003172 013700 001240      R0005:  MOV    TCCM,R0    ;TCCM ADDR TO R0.
1002 003176 005010              CLR    (0)            ;SELECT U0, FUNCTION 0.(SAT COMMAND).
1003 003200 005210              INC    (0)            ;DO.
1004 003202 105710              TSTB   (0)            ;SEE IF READY IS SET.
1005 003204 100001              BPL    A0005          ;BR IF READY NOT SET. (OK).
1006 003206 104001              ERROR  1              ;SAT COMMAND FAILED TO CLEAR READY.
1007 003210
1008 003210 012706 001000      A0005:  MOV    #1000,SP      ;RESTORE THE STACK POINTER
1009 003214 000400              BR      T0006          ;GO ON TO THE NEXT TEST
1010      ;CHECK THAT ISSUING SST COMMAND (STOP SELECTED TRANSPORT) CAUSES READY
1011      ;BIT TO CLEAR IMMEDIATELY (TCCM BIT 7)
1012      .SBTTL T0006
1013      ;*****
1014 003216 000004              T0006:  SCOPE
1015 003220 012706 001000      MOV    #1000,SP      ;SETUP THE STACK POINTER
1016 003224 004737 011766      JSR    PC,TORDER      ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1017 003230 000006              00006      ;HERE LIES THE NUMBER OF THIS TEST
1018 003232 013700 001240      R0006:  MOV    TCCM,R0    ;TCCM ADDR TO R0.
1019 003236 012710 000010      MOV    #10,(0)        ;SELECT U0,FUNCTION 100. (SST COMMAND).
1020 003242 005210              INC    (0)            ;DO.
1021 003244 105710              TSTB   (0)            ;SEE IF READY IS SET.
1022 003246 100001              BPL    A0006          ;BR IF READY NOT SET. (OK).
1023 003250 104002              ERROR  2              ;SST COMMAND FAILED TO CLEAR READY.
1024 003252
1025 003252 012706 001000      A0006:  MOV    #1000,SP      ;RESTORE THE STACK POINTER
1026 003256 000400              BR      T0007          ;GO ON TO THE NEXT TEST
1027      ;TEST THAT READY BIT CAN CAUSE AN INTERRUPT. IF THE INTERRUPT IS SERVICED,
1028      ;IT WILL HAVE OCCURRED AT THE CORRECT VECTOR.
1029      .SBTTL T0007
1030      ;*****
1031 003260 000004              T0007:  SCOPE
1032 003262 012706 001000      MOV    #1000,SP      ;SETUP THE STACK POINTER
1033 003266 004737 011766      JSR    PC,TORDER      ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1034 003272 000007              00007      ;HERE LIES THE NUMBER OF THIS TEST
1035 003274 004737 012174      R0007:  JSR    PC,STTCV  ;SET INTERRUPT VECTOR TO CB.
1036 003300 003324
```

```
1037 003302 005077 175732          CLR      @TCCM          ;DISABLE TC11 INTERRUPTS.
1038 003306 005037 177776          CLR      PSW           ;SET PROCESSOR PRIORITY 0.
1039 003312 052777 000100 175720  BIS      #BIT6,@TCCM   ;ENABLE TC11 INTERRUPTS.
1040 003320 000240                NOP
1041 003322 104003                ERROR 3                ;READY DID NOT INTERRUPT.
1042 003324                A0007:
1043 003324 012706 001000          MOV      #1000,SP      ;RESTORE THE STACK POINTER
1044 003330 000400                BR       T0010        ;GO ON TO THE NEXT TEST
;TEST THAT READY DOES NOT CAUSE INTERRUPT WITH PROCESSOR AT SAME PRIORITY
;LEVEL AS THE TC11 INTERRUPT PRIORITY.
1045                .SBTTL T0010
1046                ;*****
1047                ;*****
1048                ;*****
1049 003332 000004                T0010: SCOPE
1050 003334 012706 001000          MOV      #1000,SP      ;SETUP THE STACK POINTER
1051 003340 004737 011766          JSR      PC,TORDER    ;MAKE SURE TESTS ARE IN P.POER SEQUENCE
1052 003344 000010                00010                ;HERE LIES THE NUMBER OF THIS TEST
1053 003346 004737 012174          R0010: JSR      PC,STTCV ;SET INTERRUPT VECTOR TO DC.
1054 003352 003410                B0010
1055 003354 013737 001252 177776  MOV      TCLVL,PSW    ;SET PROCESSOR TO SAME PRTY AS TC11.
1056 003362 005077 175652          CLR      @TCCM        ;DISABLE TC11 INTERRUPTS.
1057 003366 052777 000100 175644  BIS      #BIT6,@TCCM  ;ENABLE TC11 INTERRUPTS.
1058 003374 000240                NOP
1059 003376 005077 175636          A0010: CLR      @TCCM  ;DISABLE TC11 INTERRUPTS. (OK).
1060
1061 003402 012706 001000          MOV      #1000,SP      ;RESTORE THE STACK POINTER
1062 003406 000402                BR       T0011        ;GO ON TO THE NEXT TEST
1063 003410 104000                B0010: ERROR          ;HERE IF INT. OCCURS.
1064                ;TC11 INTERRUPTED. WITH PROCESSOR AT SAME
1065 003412 000771                BR       A0010        ;PRTY AS TC11 INTERRUPT PRTY.
1066                ;TEST THAT TC11 INTERRUPTS WHEN PROCESSOR IS AT PRIORITY ONE LEVEL LOWER
1067                ;THAN THE TC11 INTERRUPT PRIORITY.
1068                .SBTTL T0011
1069                ;*****
1070                ;*****
1071 003414 000004                T0011: SCOPE
1072 003416 012706 001000          MOV      #1000,SP      ;SETUP THE STACK POINTER
1073 003422 004737 011766          JSR      PC,TORDER    ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1074 003426 000011                00011                ;HERE LIES THE NUMBER OF THIS TEST
1075 003430 004737 012174          R0011: JSR      PC,STTCV ;SET INTERRUPT VECTOR TO EB.
1076 003434 003472                A0011
1077 003436 005077 175576          CLR      @TCCM        ;DISABLE TC11 INTERRUPTS.
1078 003442 013737 001252 177776  MOV      TCLVL,PSW    ;SET PROCESSOR TO PRTY ONE LEVEL LOWER
1079 003450 162737 000040 177776  SUB      #40,PSW       ;THAN TC11 INTEPRUPT PRTY.
1080 003456 052777 000100 175554  BIS      #BIT6,@TCCM  ;ENABLE TC11 INTERRUPTS.
1081 003464 000240                NOP
1082 003466 104003                ERROR 3                ;TC11 FAILED TO INT. WITH PROCESSOR AT
1083 003470 000401                BR       B0011        ;PRTY ONE LEVEL LOWER THAN TC11 INT. PRTY.
1084 003472 022626          A0011: POPSP2          ;HERE IF INT. OCCURS. POP STACK TWICE.
1085 003474 005077 175540          B0011: CLR      @TCCM  ;DISABLE TC11 INTERRUPTS.
1086
1087 003500 012706 001000          MOV      #1000,SP      ;RESTORE THE STACK POINTER
1088 003504 000400                BR       T0012        ;GO ON TO THE NEXT TEST
1089
1090                ;TEST TC11 DOES NOT REINTERRUPT AFTEH .NITIAL I. INTERRUPT HAS BEEN SERVICED.
1091                .SBTTL T0012
1092                ;*****
```

```
1093 003506 000004 T0012: SCOPE
1094 003510 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1095 003514 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1096 003520 000012 00012 ;HERE LIES THE NUMBER OF THIS TEST
1097 003522 004737 012174 R0012: JSR PC,STTCV ;SET INTERRUPT VECTOR TO FC.
1098 003526 003564 B0012
1099 003530 005077 175504 CLR @TCCM ;DISABLE TC11 INTERRUPTS.
1100 003534 005037 177776 CLR PSW ;SET PROCESSOR PRTY 0.
1101 003540 052777 000100 175472 BIS #BIT6,@TCCM ;ENABLE TC11 INTERRUPTS.
1102 003546 000240 NOP
1103 003550 104005 ERROR 5 ;TC11 FAILED TO INTERRUPT.
1104 003552 005077 175462 A0012: CLR @TCCM ;DISABLE TC11 INTERRUPTS.
1105
1106 003556 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1107 003562 000413 BR T0013 ;GO ON TO THE NEXT TEST
1108 003564 012777 003604 175456 B0012: MOV #D0012,@TCVTR ;CHANGE INT POINTER TO FE.
1109 003572 012716 003600 MOV #C0012,@SP ;CHANGE INT EXIT POINTER TO FD.
1110 003576 000002 RTI ;EXIT INTERRUPT.
1111 003600 000240 C0012: NOP ;OK IF NO INT. REOCCURS.
1112 003602 000763 BR A0012
1113 003604 022626 D0012: POPSP2 ;HERE IF REINTERRUPT OCCURS.
1114 003606 104006 ERROR 6 ;TC11 REINTERRUPTED AFTER RTI.
1115 003610 000760 BR A0012
1116 ;TEST THAT SETTING MAINTENANCE BIT (TCCM BIT 13) SETS UPS BIT (TCST BIT 7)
1117 ;THAT CLEARING MAINTENANCE BIT C LARS UPS, AND THAT RESET CLEARS UPS.
1118 .SBTTL T0013
1119 :*****
1120 003612 000004 T0013: SCOPE
1121 003614 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1122 003620 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1123 003624 000013 00013 ;HERE LIES THE NUMBER OF THIS TEST
1124 003626 032777 000200 175402 R0013: BIT #BIT7,@TCST ;SEE IF UPS IS CLEAR.
1125 003634 001402 BEQ A0013 ;BR IF UPS IS CLEAR.
1126 003636 104007 ERROR 7 ;RESET FAILED TO CLEAR UPS.
1127 003640 000421 BR C0013
1128 003642 052777 020000 175370 A0013: BIS #BIT13,@TCCM ;SET MAINTENALCE BIT.
1129 003650 032777 000200 175360 BIT #BIT7,@TCST ;SEE IF UPS IS SET.
1130 003656 001002 BNE B0013 ;BR IF UPS IS SET.
1131 003660 104010 ERROR 10 ;MAINT BIT FAILED TO SET UPS.
1132 003662 000410 BR C0013
1133 003664 042777 020000 175346 B0013: BIC #BIT13,@TCCM ;CLEAR MAINT BIT.
1134 003672 032777 000200 175336 BIT #BIT7,@TCST ;SEE IF UPS IS CLEAR.
1135 003700 001401 BEQ C0013 ;BR IF UPS IS CLEAR.
1136 003702 104011 ERROR 11 ;CLEARING MAINT. BIT FAILED TO CLEAR UPS.
1137 003704 052777 020000 175326 C0013: BIS #BIT13,@TCCM ;SET MAINT BIT TO SET UPS.
1138 003712 004737 012220 JSR PC,SRSETT ;ISSUE RESET TO CLEAR MAINT AND UPS BITS.
1139
1140 003716 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1141 003722 000400 BR T0014 ;GO ON TO THE NEXT TEST
1142 ;TEST THAT SETTING MAINT. BIT DISABLES LOADING XD16 (TCST BIT 0).
1143 .SBTTL T0014
1144 :*****
1145 003724 000004 T0014: SCOPE
1146 003726 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1147 003732 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1148 003736 000014 00014 ;HERE LIES THE NUMBER OF THIS TFST
```



```
1149 003740 052777 020000 175272 R0014: BIS #BIT13,@TCCM ;SET MAINTENANCE BIT
1150 003746 052777 000001 175262 BIS #BIT0,@TCST ;TRY SETTING XD16.
1151 003754 032777 000001 175254 BIT #BIT0,@TCST ;SEE IF XD16 IS SET.
1152 003762 001401 BEQ A0014 ;BR IF XD16 IS CLEAR.
1153 003764 104012 ERROR 12 ;MAINT BIT SET FAILS TO PREVENT LOADING
1154 003766 004737 012220 A0014: JSR PC,SRSETT ;OF XD16.
1155
1156 003772 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1157 003776 000400 BR T0015 ;GO ON TO THE NEXT TEST
1158 ;TEST THAT SETTING MAINT. BIT DISABLES LOADING XD17 (TCST BIT 1).
1159 .SBTTL T0015
1160 *****
1161 004000 000004 T0015: SCOPE
1162 004002 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1163 004006 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PROPER SEQUENCE
1164 004012 000015 00015 ;HERE LIES THE NUMBER OF THIS TEST
1165 004014 052777 020000 175216 R0015: BIS #BIT13,@TCCM ;SET MAINTENANCE BIT.
1166 004022 052777 000002 175206 BIS #BIT1,@TCST ;TRY SETTING XD17.
1167 004030 032777 000002 175200 BIT #BIT1,@TCST ;SEE IF XD17 IS SET.
1168 004036 001401 BEQ A0015 ;BR IF XD17 IS CLEAR.
1169 004040 104013 ERROR 13 ;MAINT BIT FAILED TO PREVENT SETTING
1170 004042 004737 012220 A0015: JSR PC,SRSETT ;OF XD17.
1171
1172 004046 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1173 004052 000400 BR T0016 ;GO ON TO THE NEXT TEST
1174
1175 ;CHECK THAT ISSUING WRTM COMMAND WITH WRTM SWITCH OFF CAUSES AN ILO ERROR.
1176 ;(ILLEGAL OP- TCST BIT 12), AND THAT ERROR BIT SETS. (TCCM BIT 15).
1177 ;TEST DONE WITH MAINTENANCE BIT SET.
1178 .SBTTL T0016
1179 *****
1180 004054 000004 T0016: SCOPE
1181 004056 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1182 004062 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PROPER SEQUENCE
1183 004066 000016 00016 ;HERE LIES THE NUMBER OF THIS TEST
1184 004070 012777 020012 175142 R0016: MOV #MAINT!FWD!UO!WRTM,@TCCM
1185 004076 000240 NOP
1186 004100 032777 010000 175130 BIT #BIT12,@TCST ;SEE IF ILO ERROR IS SET.
1187 004106 001002 BNE A0016 ;BR IF ILO ERR IS SET.
1188 004110 104014 ERROR 14 ;WRTM COMMAND WITH WRTM SWITCH DISABLED
1189 004112 000421 BR D0016 ;FAILED TO SET ILO ERROR.
1190 004114 005777 175120 A0016: TST @TCCM ;SEE IF ERROR BIT IS SET.
1191 004120 100402 BMI B0016 ;BR IF ERROR BIT IS SET.
1192 004122 104015 ERROR 15 ;ILO ERR FAILED TO SET ERROR BIT.
1193 004124 000414 BR D0016
1194 004126 005077 175106 B0016: CLR @TCCM ;CLEAR ILLEGAL COMMAND.
1195 004132 032777 010000 175076 BIT #BIT12,@TCST ;SEE IF ILO ERROR IS SET.
1196 004140 001402 BEQ C0016 ;BR IF ILO ERROR IS CLEAR.
1197 004142 104016 ERROR 16 ;CLEARING ILLEGAL OP FAILED TO CLEAR
1198 004144 000404 BR D0016 ;ILO ERROR.
1199 004146 005777 175066 C0016: TST @TCCM ;SEE IF ERROR BIT IS CLEAR.
1200 004152 100001 BPL D0016 ;BR IF ERROR IS CLEAR.
1201 004154 104017 ERROR 17 ;CLEARING ILLEGAL OP FAILED TO
1202 004156 004737 012220 D0016: JSR PC,SRSETT ;CLEAR ERROR BIT.
1203
1204 004162 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
```

```
1205 004166 000400 BR T0017 ;GO ON TO THE NEXT TEST
1206 ;CHECK THAT ISSUING WRTM COMMAND (WRITE TIMING AND MARK) WITH WRTM SWITCH
1207 ;OFF CAUSES AN ILO ERROR(ILLEGAL OP- TCST BIT 12),ALD THAT ERROR BIT SETS.
1208 ;(TCCM BIT 15). TEST DONE WITH MAINTENANCE BIT SET.
1209 .SBTTL T0017
1210 :*****
1211 004170 000004 T0017: SCOPE
1212 004172 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1213 004176 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1214 004202 000017 00017 ;HERE LIES THE NUMBER OF THIS TEST
1215 004204 012777 020012 175026 R0017: MOV #MAINT!FWD!UO!WRTM,@TCCM
1216 004212 000240 NOP
1217 004214 032777 010000 175014 BIT #BIT12,@TCST ;SEE IF ILO ERR IS SET.
1218 004222 001002 BNE A0017 ;BR IF ILO SET.
1219 004224 104020 ERROR 20 ;WRTM WITH WRTM SW OFF FAILED TO SET
1220 004226 000421 BR D0017 ;ILO ERROR.
1221 004230 005777 175004 A0017: TST @TCCM ;ERROR BIT SET?
1222 004234 100402 BMI B0017 ;BR IF ERROR BIT SET.
1223 004236 104021 ERROR 21 ;ERROR BIT NOT SET WITH ILO ERR SET.
1224 004240 000414 BR D0017
1225 004242 005077 174772 B0017: CLR @TCCM ;CLEAR ILLEGAL COMMAND.
1226 004246 032777 010000 174762 BIT #BIT12,@TCST ;SEE IF ILO ER IS CLEAR.
1227 004254 001402 BEQ C0017 ;BR ID ILO ERR IS CLEAR.
1228 004256 104016 ERROR 16 ;CLEARING ILLEGAL OP FAILED TO
1229 004260 000404 BR D0017 ;CLEAR ILO ERR.
1230 004262 005777 174752 C0017: TST @TCCM ;ERROR BIT SET?
1231 004266 100001 BPL D0017 ;BR IF ERROR BIT IS CLEAR.
1232 004270 104017 ERROR 17 ;CLEARING ILLEGAL OP FAILED TO
1233 004272 004737 012220 D0017: JSR PC,SRSETT ;CLEAR ERROR BIT.
1234
1235 004276 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1236 004302 000400 BR T0020 ;GO ON TO THE NEXT TEST
1237
1238 ;TEST THAT CLEARING ERROR BIT DOES NOT CLEAR ILO ERROR.
1239 .SBTTL T0020
1240 :*****
1241 004304 000004 T0020: SCOPE
1242 004306 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1243 004312 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1244 004316 000020 00020 ;HERE LIES THE NUMBER OF THIS TEST
1245 004320 012777 020012 174712 R0020: MOV #MAINT!FWD!UO!WRTM,@TCCM
1246 004326 000240 NOP
1247 004330 005777 174704 TST @TCCM ;ERROR SET?
1248 004334 100402 BMI A0020 ;BR IF ERROR BIT IS SET.
1249 004336 104023 ERROR 23 ;ERROR BIT FAILED TO SET.
1250 004340 000410 BR B0020
1251 004342 042777 100000 174670 A0020: BIC #BIT15,@TCCM ;TRY CLEARING ERROR BIT.
1252 004350 032777 010000 174660 BIT #BIT12,@TCST ;ILO SET?
1253 004356 001001 BNE B0020 ;BR IF ILO IS SET.
1254 004360 104022 ERROR 22 ;O TO ERROR BIT CLEARED ILO ERROR.
1255 004362 004737 012220 B0020: JSR PC,SRSETT ;RESET.
1256
1257 004366 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1258 004372 000400 BR T0021 ;GO ON TO THE NEXT TEST
1259 ;TEST THAT ERROR BIT (TCCM BIT15) IS ABLE TO CAUSE AN INTERRUPT.
1260 .SBTTL T0021
```

```
1261
1262 004374 000004
1263 004376 012706 001000
1264 004402 004737 011766
1265 004406 000021
1266 004410 004737 012174
1267 004414 004442
1268 004416 005077 174616
1269 004422 005037 177776
1270 004426 052777 000100 174604
1271 004434 000240
1272 004436 104005
1273 004440 000415
1274 004442 012777 004472 174600 A0021:
1275 004450 012716 004456 MOV #C0021,@TCVTR ;CHANGE INT VECTOR TO MD.
1276 004454 000002 MOV #B0021,@SP ;CHANGE INT EXIT POINTER TO MC.
1277 004456 052777 020012 174554 B0021: RTI ;EXIT INTERRUPT.
1278 004464 000240 BIS #MAINT!FWD!UO!WRTM,@TCCM
1279 004466 104024 NOP
1280 004470 000401 ERROR 24 ;ERROR BIT FAILED TO INTERRUPT.
1281 004472 022626 BR D0021
1282 004474 005077 174540 C0021: POPSP2 ;HERE IF ERROR INTERRUPTS.
1283 D0021: CLR @TCCM ;DISABLE INT. CLEAR ILLEGAL OP.
1284 004500 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1285 004504 000400 BR T0022 ;GO ON TO THE NEXT TEST
1286
1287 ;TEST THAT ISSUING RNUM COMMAND (READ BLOCK #) CLEARS READY BIT.
1288 ;RESET INSTRUCTION SHOULD SET READY. TEST DONE WITH MAINT. BIT SET.
1289 .SBTTL T0022
1290
1291 004506 000004
1292 004510 012706 001000
1293 004514 004737 011766
1294 004520 000022
1295 004522 105777 174512 R0022:
1296 004526 100402 TSTB @TCCM ;READY SET?
1297 004530 104025 BMI A0022 ;BR IF READY IS SET.
1298 004532 000407 ERROR 25 ;RESET DID NOT FORCE READY TO SET.
1299 004534 012777 020003 174476 A0022: BR B0022
1300 004542 105777 174472 MOV #MAINT!UO!FWD!RNUM!DO,@TCCM
1301 004546 100001 TSTB @TCCM ;READY CLEAR?
1302 004550 104076 BPL B0022 ;BR IF READY IS CLEAR.
1303 004552 004737 012220 B0022: ERROR 76 ;RNUM,DO, FAILED TO CLEAR READY.
1304 JSR PC,SRSETT ;ISSUE RESET TO FORCE READY TO SET.
1305 004556 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1306 004562 000400 BR T0023 ;GO ON TO THE NEXT TEST
1307 ;TEST THAT TC11 CONTROL CAN RECOGNIZE A BLOCK MARK. WITH MAINT BIT SET,
1308 ;RNUM COMMAND IS ISSUED. A SUBROUTINE PROVIDES TIMING AND MARK DATA.
1309 ;WHEN THE BLOCK MARK HAS BEEN SHIFTED INTO THE WINDOW, THE READY BIT SHOULD SET.
1310 .SBTTL T0023
1311
1312 004564 000004
1313 004566 012706 001000
1314 004572 004737 011766
1315 004576 000023
1316 004600 005077 174434 R0023:
T0021: SCOPE
MOV #1000,SP ;SETUP THE STACK POINTER
JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
00021 ;HERE LIES THE NUMBER OF THIS TEST
R0021: JSR PC,STTCV ;SET TC11 INT. VECTOR TO MB.
A0021: CLR @TCCM ;DISABLE TC11 INTERRUPTS.
CLR PSW ;SET PRTY 0.
BIS #BIT6,@TCCM ;ENABLE TC11 INTERRUPTS.
NOP
ERROR 5 ;TC11 FAILED TO INTERRUPT.
BR D0021
A0021: MOV #C0021,@TCVTR ;CHANGE INT VECTOR TO MD.
MOV #B0021,@SP ;CHANGE INT EXIT POINTER TO MC.
RTI ;EXIT INTERRUPT.
B0021: BIS #MAINT!FWD!UO!WRTM,@TCCM
NOP
ERROR 24 ;ERROR BIT FAILED TO INTERRUPT.
BR D0021
C0021: POPSP2 ;HERE IF ERROR INTERRUPTS.
D0021: CLR @TCCM ;DISABLE INT. CLEAR ILLEGAL OP.
MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0022 ;GO ON TO THE NEXT TEST
T0022: SCOPE
MOV #1000,SP ;SFTUP THE STACK POINTER
JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
00022 ;HERE LIES THE NUMBER OF THIS TEST
R0022: TSTB @TCCM ;READY SET?
BMI A0022 ;BR IF READY IS SET.
ERROR 25 ;RESET DID NOT FORCE READY TO SET.
BR B0022
A0022: MOV #MAINT!UO!FWD!RNUM!DO,@TCCM
TSTB @TCCM ;READY CLEAR?
BPL B0022 ;BR IF READY IS CLEAR.
ERROR 76 ;RNUM,DO, FAILED TO CLEAR READY.
B0022: JSR PC,SRSETT ;ISSUE RESET TO FORCE READY TO SET.
MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0023 ;GO ON TO THE NEXT TEST
T0023: SCOPE
MOV #1000,SP ;SETUP THE STACK POINTER
JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
00023 ;HERE LIES THE NUMBER OF THIS TEST
R0023: CLR @TCCM
```

```
1317 004604 012777 020003 174426      MOV      #MAINT!UO!FWD!RNUM!DO,@TCCM
1318 004612                                     MTCOD    MTK7,6
1319 004612 004537 012674      JSR      R5,LMTCOD      ;CALL LOAD MT CODES SUB.
1320 004616 033306                                     MTK7     ;ADDRESS OF MARK TRACK CODES.
1321 004620 000006      6         ;MARK TRACK CODE COUNT.
1322 004622 005777 174412      TST      @TCCM          ;ERROR BIT SET?
1323 004626 100002      BPL      A0023          ;BR IF NO ERROR.
1324 004630 104027      ERROR 27 ;ERROR BIT SET. EXAMINE TCST OR LIGHT PANEL.
1325 004632 000404      BR       B0023
1326 004634 105777 174400      A0023: TSTB    @TCCM          ;READY BIT SET?
1327 004640 100401      BMI     B0023          ;BR IF READY IS SET.
1328 004642 104030      ERROR 30 ;READY NOT SET AFTER BLOCK MARK WAS
1329                                     ;SHIFTED INTO WINDOW REG WITH RNUM COMMAND
1330                                     ;IN EFFECT.EVERYTHING IS SUSPECT AT THIS
1331                                     ;POINT. ABILITY TO SHIFT TIMING AND MARK
1332                                     ;DATA WHILE IN MAINT MODE HAS NOT BEEN
1333 004644                                     B0023:
1334 004644 012706 001000      MOV      #1000,SP      ;RESTORE THE STACK POINTER
1335 004650 000400      BR       T0024          ;GO ON TO THE NEXT TEST
1336                                     ;TEST THAT TC11 CONTROL TRANSFERS THE BLOCK NUMBER TO THE DATA REGISTER
1337                                     ;WHEN BLOCK MARK IS DETECTED AND CONTROL IS DOING RNUM COMMAND. A SUBROUTINE
1338                                     ;PROVIDES TIMING, MARK, AND DATA. WHEN THE READY BIT SETS, THE BLOCK #
1339                                     ;EXPECTED IN THE DATA REGISTER IS 000525.
1340                                     .SBTTL T0024
1341                                     ;*****
1342 004652 000004      T0024: SCOPE
1343 004654 012706 001000      MOV      #1000,SP      ;SETUP THE STACK POINTER
1344 004660 004737 011766      JSR      PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1345 004664 000024      00024    ;HERE LIES THE NUMBER OF THIS TEST
1346 004666 005077 174346      R0024: CLR      @TCCM
1347 004672 012777 020003 174340      MOV      #MAINT!UO!FWD!RNUM!DO,@TCCM
1348 004700                                     MTCOD    MTK7,6
1349 004700 004537 012674      JSR      R5,LMTCOD      ;CALL LOAD MT CODES SUB.
1350 004704 033306                                     MTK7     ;ADDRESS OF MARK TRACK CODES.
1351 004706 000006      6         ;MARK TRACK CODE COUNT.
1352 004710 022777 052525 174330      CMP      #52525,@TCDT  ;TCDT=52525?
1353 004716 001415      BEQ      A0024          ;BR IF TCDT CORRECT.
1354 004720 017737 174322 001172      MOV      @TCDT,$REG4   ;SETUP BLOBK # FOR PRINTOUT
1355 004726 012737 052525 001162      MOV      #52525,$REG0  ;SETUP GOOD BLOBK # TO PRINTOUT
1356 004734 017737 174306 001172      MOV      @TCDT,$REG4   ;SETUP BLOBK # FOR PRINTOUT
1357 004742 012737 052525 001162      MOV      #52525,$REG0  ;SETUP GOOD BLOBK # TO PRINTOUT
1358 004750 104031      ERROR 31 ;ERROR.BLOCK # IN TCDT NOT 52525. EXAMINE TCDT.
1359 004752                                     A0024:
1360 004752 012706 001000      MOV      #1000,SP      ;RESTORE THE STACK POINTER
1361 004756 000400      BR       T0025          ;GO ON TO THE NEXT TEST
1362                                     ;TEST THAT TC11 CONTROL IS ABLE TO DETECT AN INCORRECT MARK TRACK CODE.
1363                                     ;A SUBROUTINE PROVIDES TIMING AND MARK DATA WHILE CONTROL IS IN RNUM
1364                                     ;COMMAND. WHEN THE INCORRECT MARK IS SHIFTED, THE MTE AND ERR BITS SHOULD SET.
1365                                     .SBTTL T0025
1366                                     ;*****
1367 004760 000004      T0025: SCOPE
1368 004762 012706 001000      MOV      #1000,SP      ;SETUP THE STACK POINTER
1369 004766 004737 011766      JSR      PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1370 004772 000025      00025    ;HERE LIES THE NUMBER OF THIS TEST
1371 004774 004537 012646      R0025: JSR      R5,BMOVE    ;SET INVALID CODE IN MARK TRACK.
1372 005000                                     MTKER
```

```
1373 005002 033360 MTKVAR  
1374 005004 000006 6  
1375 005006 012777 020003 174224 MOV #MAINT!UO!FWD!RNUM!DO,@TCCM  
1376 005014 MTCOD MTK7,9.  
1377 005014 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.  
1378 005020 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.  
1379 005022 000011 9. ;MARK TRACK CODE COUNT.  
1380 005024 032777 020000 174204 BIT #BIT13,@TCST ;MTE BIT SET? (MARK TRACK ERROR).  
1381 005032 001002 BNE A0025 ;BR IF MTE BIT IS SET.  
1382 005034 104032 ERROR 32 ;INVALID MARK TRACK CODE FAILED TO SET MTE.  
1383 005036 000404 BR B0025  
1384 005040 005777 174174 A0025: TST @TCCM ;ERROR BIT SET?  
1385 005044 100401 BMI B0025 ;BR IF ERROR BIT IS SET.  
1386 005046 104033 ERROR 33 ;MTE BIT FAILED TO SET ERROR BIT.  
1387 005050 B0025:  
1388 005050 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER  
1389 005054 000400 BR T0026 ;GO ON TO THE NEXT TEST  
1390 ;TEST THAT TC11 CONTROL DETECTS END ZONE MARK CODES. A SUBROUTINE PROVIDES  
1391 ;TIMING AND MARK DATA WHILE CONTROL IS IN RNUM COMMAND. WHEN THE ENDZ  
1392 ;MARK CODE IS SHIFTED INTO THE WINDOW, THE ENDZ AND ERROR BITS SHOULD SET.  
1393 .SBTTL T0026  
1394 *****  
1395 005056 000004 T0026: SCOPE  
1396 005060 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER  
1397 005064 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PROPER SEQUENCE  
1398 005070 000026 00026 ;HERE LIES THE NUMBER OF THIS TEST  
1399 005072 004537 012646 R0026: JSR R5,BMOVE ;SET END CODE IN MARK TRACK.  
1400 005076 033264 MTKEND  
1401 005100 033336 MTK5  
1402 005102 000006 6  
1403 005104 012777 020003 174126 MOV #MAINT!UO.FWD!RNUM!DO,@TCCM  
1404 005112 MTCOD MTK7,5  
1405 005112 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.  
1406 005116 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.  
1407 005120 000005 5 ;MARK TRACK CODE COUNT.  
1408 005122 005777 174110 TST @TCST ;ENDZ BIT SET?  
1409 005126 100402 BMI A0026 ;BR IF ENDZ BIT IS SET.  
1410 005130 104034 ERROR 34 ;ENDZ MARK FAILED TO SET ENDZ BIT.  
1411 005132 000404 BR B0026  
1412 005134 005777 174100 A0026: TST @TCCM ;ERROR BIT SET?  
1413 005140 100401 BMI B0026 ;BR IF ERROR BIT IS SET.  
1414 005142 104035 ERROR 35 ;ENDZ BIT FAILED TO SET ERROR BIT.  
1415 005144 B0026:  
1416 005144 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER  
1417 005150 000400 BR T0027 ;GO ON TO THE NEXT TEST  
1418 ;TEST THAT TC11 CONTROL DOES NOT RECOGNIZE MARK TRACK CODE 55 AS END ZONE  
1419 ;BLOCK MARK. SUBROUTINE PROVIDES TIMING AND MARK DATA.  
1420 .SBTTL T0027  
1421 *****  
1422 005152 000004 T0027: SCOPE  
1423 005154 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER  
1424 005160 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PROPER SEQUENCE  
1425 005164 000027 00027 ;HERE LIES THE NUMBER OF THIS TEST  
1426 005166 004537 012646 R0027: JSR R5,BMOVE ;SET CODE 55 IN MARK TRACK.  
1427 005172 033272 MTK55  
1428 005174 033336 MTK5
```

```
1429 005176 000006
1430 005200 005077 174034
1431 005204 012777 020003 174026
1432 005212
1433 005212 004537 012674
1434 005216 033306
1435 005220 000005
1436 005222 005777 174010
1437 005226 100002
1438 005230 104036
1439 005232 000404
1440 005234 005777 174000
1441 005240 100001
1442 005242 104037
1443 005244
1444 005244 012706 001000
1445 005250 000400
1446
1447
1448
1449
1450 005252 000004
1451 005254 012706 001000
1452 005260 004737 011766
1453 005264 000030
1454 005266 004737 012174
1455 005272 005332
1456 005274 005077 173740
1457 005300 012777 020103 173732
1458 005306
1459 005306 004537 012674
1460 005312 033306
1461 005314 000004
1462 005316 105777 173716
1463 005322 100402
1464 005324 104040
1465 005326 000401
1466 005330 104003
1467 005332
1468 005332 012706 001000
1469 005336 000400
1470
1471
1472
1473
1474
1475 005340 000004
1476 005342 012706 001000
1477 005346 004737 011766
1478 005352 000031
1479 005354 004737 012470
1480 005360 004737 012174
1481 005364 005530
1482 005366 005077 173646
1483 005372 012777 020103 173640
1484 005400
```

```
6
CLR @TCCM
MOV #MAINT!UO!FWD.RNUM!DO,@TCCM
MTCOD MTK7,5
JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
MTK7 ;ADDRESS OF MARK TRACK CODES.
5 ;MARK TRACK CODE COUNT.
TST @TCST ;ENDZ BIT SET?
BPL A0027 ;BR IF NOT SET.
ERROR 36 ;MARK CODE 55 INTERPRETED AS END ZONE.
BR B0027
A0027: TST @TCCM ;ERROR BIT SET?
BPL B0027 ;BR IF NO ERROR.
ERROR 37 ;ERROR BIT SET. EXAMINE TCST.
B0027:
MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0030 ;GO ON TO THE NEXT TEST
;TEST THAT TC11 INTERRUPTS. RNUM COMMAND IS ISSUED. SUBROUTINE PROVIDES
;TIMING AND MARK. WHEN BLOCK IS FOUND INTERRUPT SHOULD OCCUR.
.SBTTL T0030
;*****
T0030: SCOPE
MOV #1000,SP ;SETUP THE STACK POINTER
JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
00030 ;HERE LIES THE NUMBER OF THIS TEST
R0030: JSR PC,STTCV ;SET INTERRUPT VECTOR TO UE.
D0030
CLR @TCCM
MOV #MAINT!UO!FWD!IE!RNUM!DO,@TCCM
MTCOD MTK7,4
JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
MTK7 ;ADDRESS OF MARK TRACK CODES.
4 ;MARK TRACK CODE COUNT.
TSTB @TCCM ;READY SET?
BMI A0030 ;BR IF READY SET.
ERROR 40 ;READY DID NOT SET.
BR D0030
A0030: ERROR 3 ;READY FAILED TO INTERRUPT.
D0030:
MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0031 ;GO ON TO THE NEXT TEST
;TEST THAT TC11 IS ABLE TO TRANSFER ONE WORD TO CORE STORAGE. SUBROUTINE
;PROVIDES TIMING AND MARK. AFTER BLOCK IS 'FOUND' TEST SWITCHES TO
;RDATA COMMAND WITH WORD COUNT OF -1.
.SBTTL T0031
;*****
T0031: SCOPE
MOV #1000,SP ;SETUP THE STACK POINTER
JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
00031 ;HERE LIES THE NUMBER OF THIS TEST
R0031: JSR PC,CLRBUF ;CLEAR READ BUFFER.
JSR PC,STTCV ;SET INTERRUPT VECTOR TO VG.
G0031
CLR @TCCM
MOV #MAINT!UO!FWD!IE.RNUM!DO,@TCCM
MTCOD MTK7,7
```

1485	005400	004537	012674			JSR	R5,LMTCOD	:CALL LOAD MT CODES SUB.
1486	005404	033306				MTK7		:ADDRESS OF MARK TRACK CODES.
1487	005406	000007				7		:MARK TRACK CODE COUNT.
1488	005410	005777	173624			TST	@TCCM	:ERROR BIT SET?
1489	005414	100002				BPL	A0031	:BR IF NO ERROR.
1490	005416	104041				ERROR 41		:ERROR BIT SET. EXAMINE TCST.
1491	005420	000440				BR	F0031	
1492	005422	105777	173612	A0031:		TSTB	@TCCM	:READY BIT SET?
1493	005426	100002				BPL	B0031	:BR IF READY NOT SET.
1494	005430	104042				ERROR 42		:READY SHOULD NOT BE SET.
1495	005432	000433				BR	F0031	
1496	005434	022737	050505	036430	B0031:	CMP	#50505,RBUF	:SEE IF 1ST WORD IN RBUF IS 50505.
1497	005442	001405				BEQ	C0031	:BR IF WORD IS 50505.
1498	005444	012737	050505	001162		MOV	#50505,\$REG0	:GOOD DATA FOR PRINTOUT
1499	005452	104043				ERROR 43		:WORD IN RBUF IS NOT 50505. EXAMINE RBUF.
1500	005454	000422				BR	F0031	:TRANSFER MAY NOT HAVE OCCURRED.
1501	005456	005777	173560		C0031:	TST	@TCWC	:WORD COUNT 0?
1502	005462	001407				BEQ	D0031	:BR IF WORD COUNT IS 0.
1503	005464	017737	173552	001172		MOV	@TCWC,\$REG4	:PREPARE ERONIOUS WORD COUNT FOR PRINTOUT
1504	005472	005077	173464			CLR	@\$REG0	:PREPARE GOOD WORD COUNT INFO FOR PRINTOUT
1505	005476	104044				ERROR 44		:WORD COUNT NOT 0.
1506	005500	000410				BR	F0031	
1507	005502	022777	036432	173534	D0031:	CMP	#RBUF+2,@TCBA	:DID BUS ADDRESS INCREMENT CORRECTLY?
1508	005510	001404				BEQ	F0031	:BR IF TCBA IS CORRECT.
1509	005512	017737	173526	001172		MOV	@TCBA,\$REG4	
1510	005520	104045				ERROR 45		:TCBA DID NOT INCREMENT OR DID IT INCORRECTLY.
1511	005522				F0031:			
1512	005522	012706	001000			MOV	#1000,SP	:RESTORE THE STACK POINTER
1513	005526	000421				BR	T0032	:GO ON TO THE NEXT TEST
1514	005530	005777	173504		G0031:	TST	@TCCM	:HERE WHEN RNUM INTERRUPTS. ERROR BIT SET?
1515	005534	100004				BPL	I0031	:BR IF NO ERROR.
1516	005536	104041				ERROR 41		:ERROR BIT SET. EXAMINE TCST.
1517								
1518	005540	012706	001000			MOV	#1000,SP	:RESTORE THE STACK POINTER
1519	005544	000412				BR	T0032	:GO ON TO THE NEXT TEST
1520	005546	012777	177777	173466	I0031:	MOV	#-1,@TCWC	:SET WORD COUNT TO -1.
1521	005554	012777	036430	173462		MOV	#RBUF,@TCBA	:SET BUS ADDR TO RBUF.
1522	005562	112777	000005	173450		MOVB	#RDATA!DO,@TCCM	:READ DATA COMMAND.
1523	005570	000002				RTI		:EXIT INTERRUPT.
1524								:TEST THAT RDATA COMMAND WITH WORD COUNT SET TO -1 TRANSFERS ONLY ONE WORD.
1525								:THAT READY IS SET WHEN THE ENTIRE 256 WORD BLOCK HAS BEEN PROCESSED, AND
1526								:THAT NO PARITY ERROR OCCURS. TEST DONE UNDER MAINTENANCE MODE.
1527								.SBTTL T0032
1528								:*****
1529	005572	000004			T0032:	SCOPE		
1530	005574	012706	001000			MOV	#1000,SP	:SETUP THE STACK POINTER
1531	005600	004737	011766			JSR	PC,TORDER	:MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1532	005604	000032				00032		:HERE LIES THE NUMBER OF THIS TEST
1533	005606	004737	012470		R0032:	JSR	PC,CLRBUF	:CLEAR READ BUFFER.
1534	005612	004737	012174			JSR	PC,STTCV	:SET INTERRUPT VECTOR TO WI.
1535	005616	006002				I0032		
1536	005620	005077	173414			CLR	@TCCM	
1537	005624	012777	020103	173406		MOV	#MAINT!UO!FWD!IE!RNUM!DO,@TCCM	
1538	005632					MTCOD	MTK7,267.	
1539	005632	004537	012674			JSR	R5,LMTCOD	:CALL LOAD MT CODES SUB.
1540	005636	033306				MTK7		:ADDRESS OF MARK TRACK CODES.


```
1541 005640 000413 267. ;MARK TRACK CODE COUNT.
1542 005642 005777 173372 TST @TCCM ;ERROR BIT SET?
1543 005646 100010 BPL B0032 ;BR IF NO ERROR.
1544 005650 032777 040000 173360 BIT #BIT14,@TCST ;WAS IT PARITY ERROR?
1545 005656 001402 BEQ A0032 ;BR IF NOT PARITY ERROR.
1546 005660 104046 ERROR 46 ;PARITY ERROR.
1547 005662 000444 BR H0032
1548 005664 104041 A0032: ERROR 41 ;ERROR BIT SET. NOT DUE TO PARITY ERROR.
1549 005666 000442 BR H0032
1550 005670 105777 173344 B0032: TSTB @TCCM ;READY BIT SET?
1551 005674 100402 BMI C0032 ;BR IF READY IS SET.
1552 005676 104047 ERROR 47 ;READY FAILED TO SET AFTER COMPLETION
1553 005700 000435 BR H0032 ;OF RDATA COMMAND.
1554 005702 022737 050505 036430 C0032: CMP #50505,RBUF ;1ST WORD EQUAL 50505?
1555 005710 001405 BEQ D0032 ;BR IF WORD IS 50505.
1556 005712 012737 050505 001162 MOV #50505,$REG0 ;GOOD DATA FOR PRINTOUT
1557 005720 104043 ERROR 43 ;1ST WORD DID NOT TRANSFER TO RBUF CORRECTLY.
1558 005722 000424 BR H0032
1559 005724 005737 036432 D0032: TST RBUF+2 ;RBUF+2 EQUAL 0?
1560 005730 001402 BEQ F0032 ;BR IF RBUF+2 EQUAL 0.
1561 005732 104050 ERROR 50 ;RBUF+2 NOT 0. NO DATA SHOULD HAVE
1562 005734 000417 BR H0032 ;TRANSFERRED TO IT.
1563 005736 005777 173300 F0032: TST @TCWC ;WORD COUNT 0?
1564 005742 001407 BEQ G0032 ;BR IF WORD COUNT IS 0.
1565 005744 017737 173272 001172 MOV @TCWC,$REG4 ;PREPARE ERONIOUS WORD COUNT FOR PRINTOUT
1566 005752 005077 173204 CLR @REG0 ;PREPARE GOOD WORD COUNT INFO FOR PRINTOUT
1567 005756 104044 ERROR 44 ;WORD COUNT NOT 0.
1568 005760 000405 BR H0032
1569 005762 022777 036432 173254 G0032: CMP #RBUF+2,@TCBA ;IS BUS ADDR CORRECT?
1570 005770 001401 BEQ H0032 ;BR IF TCBA OK.
1571 005772 104045 ERROR 45 ;TCBA ADDR INCORRECT. SHOULD CONTAIN RBUF+2.
1572 005774 H0032:
1573 005774 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1574 006000 000421 BR T0033 ;GO ON TO THE NEXT TEST
1575 006002 005777 173232 I0032: TST @TCCM ;HERE WHEN RNUM INTERRUPTS. ERROR?
1576 006006 100004 BPL K0032 ;BR IF NO ERROR.
1577 006010 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
1578
1579 006012 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1580 006016 000412 BR T0033 ;GO ON TO THE NEXT TEST
1581 006020 012777 177777 173214 K0032: MOV #-1,@TCWC ;SET WORD COUNT TO -1.
1582 006026 012777 036430 173210 MOV #RBUF,@TCBA ;SET BUS ADDR TO RBUF.
1583 006034 112777 000005 173176 MOVB #RDATA!DO,@TCCM ;READ DATA COMMAND.
1584 006042 000002 RTI ;EXIT INTERRUPT.
1585 ;TEST THAT TC11 IS ABLE TO DETECT INCORRECT PARITY. RDATA COMMAND IS ISSUED.
1586 ;TCWC=-1. BLOCK TO BE READ CONTAINS BAD CHECKSUM. TEST DONE IN MAINT. MODE.
1587 .SBTTL T0033
1588 ;*****
1589 006044 000004 T0033: SCOPE
1590 006046 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1591 006052 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1592 006056 000033 00033 ;HERE LIES THE NUMBER OF THIS TEST
1593 006060 004737 013152 R0033: JSR PC,MBCKSM ;BAD CHECKSUM TO FCKSM.
1594 006064 004737 012470 JSR PC,CLRBUF ;CLEAR READ BUFFER.
1595 006070 004737 012174 JSR PC,STTCV ;SET INTERRUPT VECTOR TO XE.
1596 006074 006212 D0033
```

```

1597 006076 005077 173136 CLR @TCCM
1598 006102 012777 020103 173130 MOV #MAINT!UO!FWD!IE!RNUM!DO,@TCCM
1599 006110 MTCOD MTK7,267.
1600 006110 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
1601 006114 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.
1602 006116 000413 267. ;MARK TRACK CODE COUNT.
1603 006120 032777 040000 173110 BIT #BIT14,@TCST ;PARITY ERROR SET?
1604 006126 001005 BNE A0033 ;BR IF PARITY ERROR SET.
1605 006130 017737 173106 001162 MOV @TCWC,$REGO
1606 006136 104052 ERROR 52 ;PARITY ERROR NOT DETECTED.(BIT NOT SET).
1607 006140 000421 BR C0033
1608 006142 005777 173072 A0033: TST @TCCM ;ERROR BIT SET?
1609 006146 100405 BMI B0033 ;BR IF ERROR BIT SET.
1610 006150 017737 173066 001162 MOV @TCWC,$REGO
1611 006156 104053 ERROR 53 ;PARITY ERROR DID NOT SET ERROR BIT.
1612 006160 000411 BR C0033
1613 006162 005077 173052 B0033: CLR @TCCM ;CLEAR COMMAND REGISTER.
1614 006166 005777 173046 TST @TCCM ;ERROR BIT CLEAR?
1615 006172 100004 BPL C0033 ;BR IF ERROR BIT IS CLEAR.
1616 006174 017737 173042 001162 MOV @TCWC,$REGO
1617 006202 104054 ERROR 54 ;CLEARING TCCM FAILED TO CLEAR PARITY ERROR.
1618 006204 C0033:
1619 006204 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1620 006210 000421 BR T0034 ;GO ON TO THE NEXT TEST
1621 006212 005777 173022 D0033: TST @TCCM ;HERE WHEN RNUM INTERRUPTS. ERROR?
1622 006216 100004 BPL G0033 ;BR IF NO ERROR.
1623 006220 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
1624
1625 006222 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1626 006226 000412 BR T0034 ;GO ON TO THE NEXT TEST
1627 006230 012777 177777 173004 G0033: MOV #-1,@TCWC ;-1 TO WORD COUNT.
1628 006236 012777 036430 173000 MOV #RBUF,@TCBA ;SET BUS ADDR TO RBUF.
1629 006244 112777 000005 172766 MOVB #RDATA!DO,@TCCM ;RDATA COMMAND.
1630 006252 000002 RTI ;EXIT INTERRUPT.
1631 ;READ 256 WORDS WITH RDATA COMMAND UNDER MAINTENANCE MODE. ALL DATA SHOULD
1632 ;TRANSFER CORRECTLY. NO CONTROL ERRORS SHOULD OCCUR.
1633 .SBTTL T0034
1634 *****
1635 006254 000004 T0034: SCOPE
1636 006256 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1637 006262 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1638 006266 000034 00034 ;HERE LIES THE NUMBER OF THIS TEST
1639 006270 004737 012470 R0034: JSR PC,CLRBUF ;CLEAR READ BUFFER.
1640 006274 004737 012174 JSR PC,STTCV ;SET INTERRUPT VECTOR TO YF.
1641 006300 006416 F0034
1642 006302 005077 172732 CLR @TCCM
1643 006306 012777 020103 172724 MOV #MAINT!UO!FWD!IE!RNUM!DO,@TCCM
1644 006314 MTCOD MTK7,267.
1645 006314 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
1646 006320 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.
1647 006322 000413 267. ;MARK TRACK CODE COUNT.
1648 006324 005777 172710 TST @TCCM ;ERROR BIT SET?
1649 006330 100002 BPL A0034 ;BR IF NO ERROR.
1650 006332 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
1651 006334 000425 BR D0034
1652 006336 005777 172700 A0034: TST @TCWC ;WORD COUNT 0?
  
```

```
1653 006342 001407 BEQ B0034 ;BR IF WORD COUNT IS 0.
1654 006344 017737 172672 001172 MOV @TCWC,$REG4 ;PREPERE ERONIOUS WORD COUNT FOR PRINTOUT
1655 006352 005077 172604 CLR @$REG0 ;PREPARE GOOD WORD COUNT INFO FOR PRINTOUT
1656 006356 104044 ERROR 44 ;WORD COUNT NOT 0.
1657 006360 000413 BR D0034
1658 006362 022777 037430 172654 B0034: CMP #RBUF+512.,@TCBA;BUS ADDR CORRECT?
1659 006370 001402 BEQ C0034 ;BR IF TCBA OK.
1660 006372 104051 ERROR 51 ;TCBA INCORRECT. SHOULD BE EQUAL TO
1661 006374 000405 BR D0034 ;RBUF+512.
1662 006376 004537 013166 C0034: JSR R5,CKDAT ;COMPARE 256 WORDS STARTING AT RBUF.
1663 006402 001276 SBDAT1 ;REPORT ANY ERRORS.
1664 006404 036430 RBUF
1665 006406 000400 256.
1666 006410 D0034:
1667 006410 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1668 006414 000421 BR T0035 ;GO ON TO THE NEXT TEST
1669 006416 005777 172616 F0034: TST @TCCM ;HERE WHEN RNUM INTERRUPTS. ERROR?
1670 006422 100004 BPL H0034 ;BR IF NO ERROR.
1671 006424 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
1672
1673 006426 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
1674 006432 000412 BR T0035 ;GO ON TO THE NEXT TEST
1675 006434 012777 177400 172600 H0034: MOV #-256.,@TCWC ;:-256 TO WORD COUNT.
1676 006442 012777 036430 172574 MOV #RBUF,@TCBA ;SET BUS ADDR TO RBUF.
1677 006450 112777 000005 172562 MOVB #RDATA!DO,@TCCM ;READ DATA COMMAND.
1678 006456 000002 RTI ;EXIT INTERRUPT.
1679 ;READ 2 DATA BLOCKS (512 WORDS) WITH RDATA COMMAND UNDER MAINTENANCE MODE.
1680 ;ALL DATA SHOULD TRANSFER CORRECTLY. NO ERRORS SHOULD OCCUR.
1681 .SBTTL T0035
1682 :*****
1683 006460 000004 T0035: SCOPE
1684 006462 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1685 006466 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1686 006472 000035 00035 ;HERE LIES THE NUMBER OF THIS TEST
1687 006474 004737 012470 R0035: JSR PC,CLRBUF ;CLEAR READ BUFFER.
1688 006500 004737 012174 JSR PC,STTCV ;SET INTERRUPT VECTOR TOZF.
1689 006504 006622 F0035
1690 006506 005077 172526 CLR @TCCM
1691 006512 012777 020103 172520 MOV #MAJNT!UO!FWD!IE!RNUM!DO,@TCCM
1692 006520 MTCOD MTK7,534.
1693 006520 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
1694 006524 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.
1695 006526 001026 534. ;MARK TRACK CODE COUNT.
1696 006530 005777 172504 TST @TCCM ;ERROR BIT SET?
1697 006534 100002 BPL A0035 ;BR IF NO ERROR.
1698 006536 104041 ERROR 41 ;ERROR BIT SET EXAMINE TCST.
1699 006540 000425 BR D0035
1700 006542 005777 172474 A0035: TST @TCWC ;WORD COUNT 0?
1701 006546 001407 BEQ B0035 ;BR IF WORD COUNT IS 0.
1702 006550 017737 172466 001172 MOV @TCWC,$REG4 ;PREPERE ERONIOUS WORD COUNT FOR PRINTOUT
1703 006556 005077 172400 CLR @$REG0 ;PREPARE GOOD WORD COUNT INFO FOR PRINTOUT
1704 006562 104044 ERROR 44 ;WORD COUNT NOT 0.
1705 006564 000413 BR D0035
1706 006566 022777 040430 172450 B0035: CMP #RBUF+1024.,@TCBA;TCBA CORRECT?
1707 006574 001402 BEQ C0035 ;BR IF TCBA IS OK.
1708 006576 104045 ERROR 45 ;TCBA INCORRECT. SHOULD BE RBUF+1024.
```

```

1709 006600 000405
1710 006602 004537 013166
1711 006606 001276
1712 006610 036430
1713 006612 001000
1714 006614
1715 006614 012706 001000
1716 006620 000421
1717 006622 005777 172412
1718 006626 100004
1719 006630 104041
1720
1721 006632 012706 001000
1722 006636 000412
1723 006640 012777 177000 172374
1724 006646 012777 036430 172370
1725 006654 112777 000005 172356
1726 006662 000002
1727
1728
1729
1730
1731 006664 000004
1732 006666 012706 001000
1733 006672 004737 011766
1734 006676 000036
1735 006700 004737 012470
1736 006704 004737 012174
1737 006710 007024
1738 006712 005077 172322
1739 006716 012777 020103 172314
1740 006724
1741 006724 004537 012674
1742 006730 033306
1743 006732 001026
1744 006734 005777 172300
1745 006740 100002
1746 006742 104041
1747 006744 000424
1748 006746 005777 172270
1749 006752 001407
1750 006754 017737 172262 001172
1751 006762 005077 172174
1752 006766 104044
1753 006770 000412
1754 006772 022777 040030 172244
1755 007000 001401
1756 007002 104045
1757 007004 004537 013166
1758 007010 001276
1759 007012 036430
1760 007014 000600
1761 007016
1762 007016 012706 001000
1763 007022 000421
1764 007024 005777 172210

C0035: BR D0035
JSR R5,CKDAT ;COMPARE 512 WORDS STARTING AT RBUF.
SBDAT1 ;REPORT ANY ERRORS.
RBUF
512.

D0035: MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0036 ;GO ON TO THE NEXT TEST
F0035: TST @TCCM ;HERE WHEN RNUM INTERRUPTS. ERROR?
BPL H0035 ;BR IF NO ERROR.
ERROR 41 ;ERROR BIT SET. EXAMINE TCST.

H0035: MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0036 ;GO ON TO THE NEXT TEST
MOV #-512.,@TCWC ;-512 TO WORD COUNT.
MOV #RBUF,@TCBA ;SET BUS ADDR TO RBUF.
MOVB #RDATA!DO,@TCCM ;READ DATA COMMAND.
RTI ;EXIT INTERRUPT.
;READ 1.5 BLOCKS (384 WORDS) WITH RDATA COMMAND UNDER MAINTENANCE MODE.
;ALL DATA SHOULD TRANSFER CORRECTLY. NO ERRORS SHOULD OCCUR.
.SBTTL T0036
;*****
T0036: SCOPE
MOV #1000,SP ;SETUP THE STACK POINTER
JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
00036 ;HERE LIES THE NUMBER OF THIS TEST
R0036: JSR PC,CLRBUF ;CLEAR READ BUFFER.
JSR PC,STTCV ;SET INTERRUPT VECTOR TO A1F.
F0036
CLR @TCCM
MOV #MAINT!UO!FWD!IE!RNUM!DO,@TCCM
MTCOD MTK7,534.
JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
MTK7 ;ADDRESS OF MARK TRACK CODES.
534. ;MARK TRACK CODE COUNT.
TST @TCCM ;ERROR BIT SET?
BPL A0036 ;BR IF NO ERROR.
ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
BR D0036
A0036: TST @TCWC ;WORD COUNT 0?
BEQ B0036 ;BR IF WORD COUNT 0.
MOV @TCWC,$REG4 ;PREPERE ERONIOUS WORD COUNT FOR PRINTOUT
CLR @SREG0 ;PREPARE GOOD WORD COUNT INFO FOR PRINTOUT
ERROR 44 ;WORD COUNT NOT 0.
BR D0036
B0036: CMP #RBUF+768.,@TCBA ;TCBA CORRECT?
BEQ C0036 ;BR IF TCBA OK.
ERROR 45 ;TCBA INCORRECT. SHOULD BE RBUF+768.
C0036: JSR R5,CKDAT ;COMPARE 384 WORDS STARTING AT RBUF.
SBDAT1 ;REPORT ANY ERRORS.
RBUF
384.

D0036: MOV #1000,SP ;RESTORE THE STACK POINTER
BR T0037 ;GO ON TO THE NEXT TEST
F0036: TST @TCCM ;HERE WHEN RNUM INTERRUPTS. ERROR?

```

```
1765 007030 100004          BPL      H0036          ;BR IF NO ERROR.
1766 007032 104041          ERROR 41          ;ERROR BIT SET. EXAMINE TCST.
1767
1768 007034 012706 001000    MOV      #1000,SP      ;RESTORE THE STACK POINTER
1769 007040 000412          BR       T0037        ;GO ON TO THE NEXT TEST
1770 007042 012777 177200 172172 H0036: MOV      #-384.,@TCWC ; -384 TO WORD COUNT.
1771 007050 012777 036430 172166 MOV      #RBUF,@TCBA  ;SET BUS ADDR TO RBUF.
1772 007056 112777 000005 172154 MOVB    #RDATA!DO,@TCCM ;READ DATA COMMAND.
1773 007064 000002          RTI              ;EXIT INTERRUPT.
1774          ;COMPLEMENT OBVERSE READ TEST. READ ONE BLOCK (256 WORDS) WITH RDATA IN REVERSE.
1775          ;ALL DATA SHOULD COMPLEMENT OBVERSE CORRECTLY. NO CONTROL ERRORS SHOULD OCCUR.
1776          .SBTTL T0037
1777          ;*****
1778 007066 000004          T0037: SCOPE
1779 007070 012706 001000    MOV      #1000,SP      ;SETUP THE STACK POINTER
1780 007074 004737 011766    JSR     PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1781 007100 000037          00037          ;HERE LIES THE NUMBER OF THIS TEST
1782 007102 004737 012470    R0037: JSR     PC,CLRBUF   ;CLEAR READ BUFFER
1783 007106 004737 012174    JSR     PC,STTCV     ;SET INTERRUPT VECTOR TO B1D
1784 007112 007170          C0037
1785 007114 005077 172120    CLR     @TCCM
1786 007120 012777 024103 172112 MOV      #MAINT!UO!REV!IE!RNUM!DO,@TCCM
1787 007126          MTCOD    MTK7,267.
1788 007126 004537 012674    JSR     R5,LMTCOD    ;CALL LOAD MT CODES SUB.
1789 007132 033306          MTK7          ;ADDRESS OF MARK TRACK CODES.
1790 007134 000413          267.         ;MARK TRACK CODE COUNT.
1791 007136 005777 172076    TST     @TCCM        ;ERROR BIT SET?
1792 007142 100002          BPL     A0037        ;BR IF NO ERROR.
1793 007144 104041          ERROR 41          ;ERROR BIT SET. EXAMINE TCST.
1794 007146 000405          BR       B0037
1795 007150 004537 013166    A0037: JSR     R5,CKDAT   ;COMPARE 256 WORDS STARTING AT RBUF.
1796 007154 001302          SBDAT2        ;REPORT ANY ERRORS.
1797 007156 036430          RBUF
1798 007160 000400          256.
1799 007162          B0037:
1800 007162 012706 001000    MOV      #1000,SP      ;RESTORE THE STACK POINTER
1801 007166 000421          BR       T0040        ;GO ON TO THE NEXT TEST
1802 007170 005777 172044    C0037: TST     @TCCM        ;HERE WHEN RNUM INTERRUPTS. ERROR.
1803 007174 100004          BPL     F0037        ;BR IF NO ERROR.
1804 007176 104041          ERROR 41          ;ERROR BIT SET. EXAMINE TCST.
1805
1806 007200 012706 001000    MOV      #1000,SP      ;RESTORE THE STACK POINTER
1807 007204 000412          BR       T0040        ;GO ON TO THE NEXT TEST
1808 007206 012777 177400 172026 F0037: MOV      #-256.,@TCWC ; -256 TO AND COUNT.
1809 007214 012777 036430 172022 MOV      #RBUF,@TCBA  ;ADDR OF RBUF TO BUS ADDRESS.
1810 007222 112777 000005 172010 MOVB    #RDATA!DO,@TCCM ;READ DATA COMMAND.
1811 007230 000002          RTI              ;EXIT INTERRUPT
1812          ;CHECK FOR CORRECT OPERATION OF BLOCK MISS ERROR.
1813          .SBTTL T0040
1814          ;*****
1815 007232 000004          T0040: SCOPE
1816 007234 012706 001000    MOV      #1000,SP      ;SETUP THE STACK POINTER
1817 007240 004737 011766    JSR     PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1818 007244 000040          00040        ;HERE LIES THE NUMBER OF THIS TEST
1819 007246 005077 171766    R0040: CLR     @TCCM
1820 007252 012777 177776 171762 MOV      #-2,@TCWC    ; -2 TO WORD COUNT.
```

1821	007260	012777	036430	171756		MOV	#RBUF,@TCBA	:RBUF ADDR TO TCBA.
1822	007266	012777	020003	171744		MOV	#MAINT!UO!FWD!RNUM!DO,@TCCM	
1823	007274					MTCOD	MTK7,5	
1824	007274	004537	012674			JSR	R5,LMTCOD	:CALL LOAD MT CODES SUB.
1825	007300	033306				MTK7		:ADDRESS OF MARK TRACK CODES.
1826	007302	000005				5		:MARK TRACK CODE COUNT.
1827	007304	005777	171730			TST	@TCCM	:ERROR BIT SET?
1828	007310	100002				BPL	A0040	:BR IF NO ERROR.
1829	007312	104041			R0040A:	ERROR 41		:ERROR BIT SET EXAMINE TCST.
1830	007314	000506				BR	F0040	
1831	007316	112777	000005	171714	A0040:	MOVB	#RDATA!DO,@TCCM	:ISSUE RDATA COMMAND.
1832	007324					MTCOD	MTK7A,2	
1833	007324	004537	012674			JSR	R5,LMTCOD	:CALL LOAD MT CODES SUB.
1834	007330	033344				MTK7A		:ADDRESS OF MARK TRACK CODES.
1835	007332	000002				2		:MARK TRACK CODE COUNT.
1836	007334	032777	002000	171674		BIT	#BIT10,@TCST	:BLOCK MISS ERROR SET?
1837	007342	001405				BEQ	B0040	:BR IF NO BLOCK MISS. OK.
1838	007344	017737	171672	001162		MOV	@TCWC,\$REGO	:MAKE WORD COUNT INFO PRINTABLE
1839	007352	104055				ERROR 55		:BLOCK MISS SET WHEN RDATA ISSUED JUST
1840	007354	000466				BR	F0040	:BEFORE REV CHECK MARK. SHOULDN'T HAVE.
1841	007356	005077	171656		B0040:	CLR	@TCCM	
1842	007362	012777	177776	171652		MOV	#-2,@TCWC	:-2 TO WORD COUNT.
1843	007370	012777	036430	171646		MOV	#RBUF,@TCBA	:RBUF ADDR TO TCBA.
1844	007376	012777	020003	171634		MOV	#MAINT!UO!FWD!RNUM!DO,@TCCM	
1845	007404					MTCOD	MTK7,6	
1846	007404	004537	012674			JSR	R5,LMTCOD	:CALL LOAD MT CODES SUB.
1847	007410	033306				MTK7		:ADDRESS OF MARK TRACK CODES.
1848	007412	000006				6		:MARK TRACK CODE COUNT.
1849	007414	005777	171620			TST	@TCCM	:ERROR BIT SET?
1850	007420	100734				BMI	R0040A	:BR IF ERROR BIT SET?
1851	007422	112777	000005	171610		MOVB	#RDATA!DO,@TCCM	:ISSUE RDATA COMMAND.
1852	007430					MTCOD	MTK7B,2	
1853	007430	004537	012674			JSR	R5,LMTCOD	:CALL LOAD MT CODES SUB.
1854	007434	033352				MTK7B		:ADDRESS OF MARK TRACK CODES.
1855	007436	000002				2		:MARK TRACK CODE COUNT.
1856	007440	032777	002000	171570		BIT	#BIT10,@TCST	:BLOCK MISS ERROR SET?
1857	007446	001005				BNE	C0040	:BR IF BLOCK MISS.
1858	007450	017737	171566	001162		MOV	@TCWC,\$REGO	:MAKE WORD COUNT INFO PRINTABLE
1859	007456	104056				ERROR 56		:BLOCK MISS FAILED TO SET WHEN RDATA ISSUED
1860	007460	000424				BR	F0040	:RIGHT AFTER REV CHECK MARK. IT SHOULD HAVE.
1861	007462	005777	171552		C0040:	TST	@TCCM	:ERROR BIT SET?
1862	007466	100405				BMI	D0040	:BR IF ERROR BIT SET.
1863	007470	017737	171546	001162		MOV	@TCWC,\$REGO	:MAKE WORD COUNT INFO PRINTABLE
1864	007476	104057				ERROR 57		:BLOCK MISS FAILED TO SET ERROR BIT.
1865	007500	000414				BR	F0040	
1866	007502	005077	171532		D0040:	CLR	@TCCM	:0 TO ERROR BIT.
1867	007506	032777	002000	171524		BIT	#BIT10,@TCCM	:BLOCK MISS CLEARED?
1868	007514	001406				BEQ	F0040	:BR IF BLOCK MISS CLEARED.
1869	007516	017737	171520	001162		MOV	@TCWC,\$REGO	:MAKE WORD COUNT INFO PRINTABLE
1870	007524	104060				ERROR 60		:0 TO ERROR FAILED TO CLEAR BLOCK MISS.
1871	007526	004737	012220			JSR	PC,SRCTT	
1872	007532				F0040:			
1873	007532	012706	001000			MOV	#1000,SP	:RESTORE THE STACK POINTER
1874	007536	000400				BR	T0041	:GO ON TO THE NEXT TEST
1875								
1876								

:READ ALL TEST (RALL)

:AFTER BLOCK IS FOUND, SWITCH TO RALL. READ 258 WORDS. 1ST WORD READ SHOULD BE

```
1877 :THE REVERSE CHECKSUM (SHOULD BE 0). LAST WORD READ SHOULD BE THE FORWARD
1878 :CHECKSUM (SHOULD BE 770000). ALL OTHER WORDS SHOULD BE DATA.
1879 :SBTTL T0041
1880 :*****
1881 007540 000004 T0041: SCOPE
1882 007542 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
1883 007546 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1884 007552 000041 00041 ;HERE LIES THE NUMBER OF THIS TEST
1885 007554 004737 012470 R0041: JSR PC,CLRBUF ;CLEAR READ BUFFER.
1886 007560 004737 012174 JSR PC,STTCV ;SET INTERRUPT VECTOR TO EIF.
1887 007564 007754 F0041
1888 007566 005077 171446 CLR @TCCM
1889 007572 012777 020103 171440 MOV #MAINT!UC!FWD!IE!RNUM!DO,@TCCM
1890 007600 MTCOD MTK7,267.
1891 007600 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
1892 007604 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.
1893 007606 000413 267. ;MARK TRACK CODE COUNT.
1894 007610 005777 171424 R0041A: TST @TCCM ;ERROR BIT SET?
1895 007614 100002 BPL R0041B ;BR IF NO ERROR.
1896 007616 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
1897 007620 000461 BR G0041
1898 007622 017724 171420 R0041B: MOV @TCDT,(4)+ ;SAVE DATA IN READ BUFFER.
1899 007626 005337 001274 DEC CTRA ;258 WORDS READ?
1900 007632 001401 BEQ A0041 ;BR IF 258 WORDS READ.
1901 007634 000002 RTI ;NOT DONE YET. EXIT INTERRUPT.
1902 007636 005737 036430 A0041: TST RBUF ;1ST WORD IN RBUF EQUAL 0?
1903 007642 001416 BEQ D0041 ;BR IF 1ST WORD IS 0.
1904 007644 022737 055555 036430 CMP #55555,RBUF ;1ST WORD EQUAL 55555?
1905 007652 001002 BNE B0041 ;BR IF NOT 55555.
1906 007654 104043 ERROR 43 ;55555. 1ST WORD READ WITH RALL WAS
1907 007656 000442 BR G0041 ;REV GUARD INSTEAD OF REV CHECKSUM.
1908 007660 022737 066666 036430 B0041: CMP #66666,RBUF ;1ST WORD EQUAL 66666?
1909 007666 001002 BNE C0041 ;BR IF NOT 66666.
1910 007670 104043 ERROR 43 ;66666. 1ST WORD READ WITH RALL WAS
1911 007672 000434 BR G0041 ;REV LOCK INSTEAD OF REV CHECKSUM.
1912 007674 104043 C0041: ERROR 43 ;1ST WORD READ WITH RALL WAS NOT
1913 007676 000432 BR G0041 ;REV CHECKSUM. EXAMINE RBUF (1ST WORD).
1914 007700 004537 013166 D0041: JSR R5,CKDAT
1915 007704 001276 SBDAT1
1916 007706 036432 RBUF+2
1917 007710 000400 256.
1918 007712 022737 170000 037432 CMP #170000,RBUF+514.;FWD CHKSUM EQUAL 170000? 1ST WORD READ.
1919 007720 001402 BEQ D0041A
1920 007722 104061 ERROR 61 ;LAST WORD READ SHOULD HAVE BEEN THE FWD CHECKSUM.
1921 007724 000417 BR G0041 ;IN CORE IT SHOULD BE 170000.
1922 007726 005777 171310 D0041A: TST @TCWC ;WORD COUNT STILL 0?
1923 007732 001402 BEQ D0041B
1924 007734 104062 ERROR 62 ;TCWC (WORD COUNT) WAS MODIFIED DURING
1925 007736 000412 BR G0041 ;RALL. SHOULDN'T HAVE.
1926 007740 022777 036430 171276 D0041B: CMP #RBUF,@TCBA ;BUS ADDRESS STILL EQUAL #RBUF?
1927 007746 001406 BEQ G0041
1928 007750 104063 ERROR 63 ;TCBA (BUS ADDRESS) MODIFIED DURING
1929 007752 000404 BR G0041 ;RALL. SHOULDN'T HAVE.
1930 007754 005777 171260 F0041: TST @TCCM ;HERE WHEN RNUM INTERRUPTS. ERROR.
1931 007760 100004 BPL I0041 ;BR IF NO ERROR.
1932 007762 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
```



```
1933 007764          G0041:
1934 007764 012706 001000      MOV    #1000,SP      ;RESTORE THE STACK POINTER
1935 007770 000421          BR     T0042        ;GO ON TO THE NEXT TEST
1936 007772 012737 000402 001274 10041: MOV    #258.,CTRA   ;NUMBER OF WORDS TO READ TO CTRA.
1937 010000 012704 036430          MOV    #RBUF,R4     ;ADDR TO STORE DATA TO R4.
1938 010004 005077 171232          CLR    @TCWC        ;ZERO WORD COUNT.
1939 010010 012777 036430 171226  MOV    #RBUF,@TCBA  ;SET BUS ADDRESS TO RBUF.
1940 010016 004737 012174          JSR   PC,STTCV     ;SET INTERRUPT VECTOR TO E1AA.
1941 010022 007610          R0041A
1942 010024 112777 000107 171206  MOVB   #RALL!IE!DO,@TCCM ;RALL COMMAND.
1943 010032 000002          RTI                ;EXIT INTERRUPT.
1944
1945 ;DATA MISS TEST. TEST THAT DATA MISS ERROR SETS WHEN DATA REGISTER (TCDT) IS
1946 ;NOT REFERENCED UNDER RALL COMMAND, BEFORE THE NEXT DATA WORD IS LOADED INTO
1947 ;THE DATA REGISTER. (READY BIT IS CLEARED WHEN IN RALL BY REFERENCING
1948 ;THE DATA REGISTER (TCDT).
1949 ;SBTTL T0042
1950 ;*****
1950 010034 000004          T0042: SCOPE
1951 010036 012706 001000      MOV    #1000,SP      ;SETUP THE STACK POINTER
1952 010042 004737 011766      JSR   PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
1953 010046 000042          00042              ;HERE LIES THE NUMBER OF THIS TEST
1954 010050 004737 012174      R0042: JSR   PC,STTCV     ;SET INTERRUPT VECTOR TO F1E.
1955 010054 010156          D0042
1956 010056 005077 171156      CLR    @TCCM
1957 010062 012777 020103 171150  MOV    #MAINT!UO!FWD!IE!RNUM!DO,@TCCM
1958 010070          MTCOD  MTK7,7
1959 010070 004537 012674      JSR   R5,LMTCOD     ;CALL LOAD MT CODES SUB.
1960 010074 033306          MTK7
1961 010076 000007          7                  ;ADDRESS OF MARK TRACK CODES.
1962 010100 032777 001000 171130  BIT    #BIT9,@TCST  ;MARK TRACK CODE COUNT.
1963 010106 001002          BNE   A0042        ;DATA MISS ERROR SET?
1964 010110 104064          ERROR 64          ;BR IF DATA MISS IS SET.
1965 010112 000416          BR    C0042        ;DATA MISS FAILED TO SET.
1966 010114 005777 171120      A0042: TST   @TCCM        ;ERROR BIT SET?
1967 010120 100402          BMI   B0042        ;BR IF ERROR BIT SET.
1968 010122 104065          ERROR 65          ;DATA MISS FAILED TO SET ERROR BIT.
1969 010124 000411          BR    C0042
1970 010126 005077 171106      B0042: CLR    @TCCM        ;0 TO ERROR BIT.
1971 010132 032777 001000 171076  BIT    #BIT9,@TCST  ;DATA MISS CLEARED?
1972 010140 001403          BEQ   C0042        ;BR IF DATA MISS IS CLEAR.
1973 010142 104066          ERROR 66          ;0 TO ERROR FAILED TO CLEAR DATA MISS.
1974 010144 004737 012220      JSR   PC,SRSETT
1975 010150          C0042:
1976 010150 012706 001000      MOV    #1000,SP      ;RESTORE THE STACK POINTER
1977 010154 000422          BR     T0043        ;GO ON TO THE NEXT TEST
1978 010156 005777 171056      D0042: TST   @TCCM        ;HERE WHEN RNUM INTERRUPTS. ERROR?
1979 010162 100004          BPL   G0042        ;BR IF NO ERROR.
1980 010164 104041          ERROR 41          ;ERROR BIT SET. EXAMINE TCST.
1981
1982 010166 012706 001000      MOV    #1000,SP      ;RESTORE THE STACK POINTER
1983 010172 000413          BR     T0043        ;GO ON TO THE NEXT TEST
1984 010174 004737 012174      G0042: JSR   PC,STTCV     ;SET INTERRUPT VECTOR TO F1H.
1985 010200 010212          H0042
1986 010202 112777 000107 171030  MOVB   #RALL!IE!DO,@TCCM ;ISSUE RALL. IE SET.
1987 010210 000002          RTI                ;EXIT INTERRUPT.
1988 010212 112777 000007 171020  H0042: MOVB   #RALL!DO,@TCCM ;HERE WHEN RALL INTERRUPTS. DISABLE INTERRUPTS.
```


1989 010220 000002
1990
1991
1992
1993
1994
1995
1996
1997
1998 010222 000004
1999 010224 012706 001000
2000 010230 004737 011766
2001 010234 000043
2002 010236 004737 013052
2003 010242 005077 170772
2004 010246 012777 020003 170764
2005 010254
2006 010254 004537 012674
2007 010260 033306
2008 010262 000005
2009 010264 005777 170750
2010 010270 100002
2011 010272 104041
2012 010274 000457
2013 010276 012777 177400 170736
2014 010304 012777 036430 170732
2015 010312 012704 037430
2016 010316 112777 000015 170714
2017 010324
2018 010324 004537 013044
2019 010330 010442
2020 010332 033344
2021 010334 000406
2022 010336 005777 170676
2023 010342 100002
2024 010344 104041
2025 010346 000432
2026 010350 105777 170664
2027 010354 100402
2028 010356 104067
2029 010360 000425
2030 010362 005777 170654
2031 010366 001407
2032 010370 017737 170646 001172
2033 010376 005077 170560
2034 010402 104044
2035 010404 000413
2036 010406 022777 037430 170630
2037 010414 001402
2038 010416 104045
2039 010420 000405
2040 010422 004537 013166
2041 010426 001276
2042 010430 037430
2043 010432 000400
2044 010434

```
RTI ;DO NOT READ TCDT, EXIT INTERRUPT.
;WRITE DATA TEST. AFTER BLOCK NUMBER IS 'FOUND', ISSUE WDATA COMMAND
;UNDER MAINTENANCE MODE, WORD COUNT = -256, TCBA = RBUF. AFTER
;EACH MARK TRACK CODE IS PASSED, THE DATA IN THE DATA REGISTER IS SAVED.
;WHEN THE OPERATION IS COMPLETED, A COMPARE OF WRITE DATA AND THE DATA
;REGISTER DATA SAVED IS MADE TO SEE IF THEY MATCH. WORD COUNT AND TCBA
;ARE ALSO CHECKED FOR CORRECT CONTENTS.
.SBTTL T0043
;*****
T0043: SCOPE
MOV #1000,SP ;SETUP THE STACK POINTER
JSR PC,ORDER ;MAKE SURE TESTS ARE IN PROPER SEQUENCE
00043 ;HERE LIES THE NUMBER OF THIS TEST
R0043: JSR PC,LBDAT1 ;SET UP WRITE DATA (256 WORDS).
CLR @TCCM
MOV #MAINT!UO!FWD!RNUM!DO,@TCCM
MTCOD MTK7,5
JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
MTK7 ;ADDRESS OF MARK TRACK CODES.
5 ;MARK TRACK CODE COUNT.
TST @TCCM ;ERROR BIT SET?
BPL A0043 ;BR IF NO ERROR.
ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
BR G0043
A0043: MOV #-256,@TCWC ;-256 TO WORD COUNT.
MOV #RBUF,@TCBA ;ADDR OF RBUF TO TCBA.
MOV #RBUF+512,R4 ;ADDR TO SAVE TCDT DATA TO R4.
MOVB #WDATA!DO,@TCCM ;ISSUE WDATA COMMAND.
MTCOE H0043,MTK7A,262.
JSR R5,LMTCOE ;CALL LOAD MT CODES SUBROUTINE.
H0043 ;ADDR TO GO AFTER EACH CODE PASSED.
MTK7A ;ADDRESS OF MARK TRACK CODES.
262. ;MARK TRACK CODE COUNT.
TST @TCCM ;ERROR BIT SET?
BPL B0043 ;BR IF NO ERROR.
ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
BR G0043
B0043: TSTB @TCCM ;READY BIT SET?
BMI C0043 ;BR IF READY IS SET.
ERROR 67 ;READY BIT FAILED TO SET.
BR G0043
C0043: TST @TCWC ;WORD COUNT 0?
BEQ D0043 ;BR IF WORD COUNT IS 0.
MOV @TCWC,$REG4 ;PREPARE ERONIOUS WORD COUNT FOR PRINTOUT
CLR @$REG0 ;PREPARE GOOD WORD COUNT INFO FOR PRINTOUT
ERROR 44 ;WORD COUNT NOT 0.
BR G0043
D0043: CMP #RBUF+512,@TCBA ;TCBA CORRECT?
BEQ F0043 ;BR IF TCBA CORRECT.
ERROR 45 ;TCBA INCORRECT. SHOULD BE RBUF+512.
BR G0043
F0043: JSR R5,CKDAT ;COMPARE WRITE DATA AGAINST TCDT SAVED
SBDAT1 ;DATA. REPORT ERRORS.
RBUF+512.
256.
G0043:
```

```
2045 010434 012706 001000      MOV      #1000,SP      ;RESTORE THE STACK POINTER
2046 010440 000403              BR          T0044      ;GO ON TO THE NEXT TEST
2047 010442 017724 170600      H0043: MOV      @TCDT,(4)+ ;HERE AFTER EACH MARK CODE IS PASSED.
2048 010446 000002              RTI                   ;SAVE TCDT DATA AND EXIT IOT TRAP.
2049                               ;WRITE DATA COMPLEMENT OBVERSE TEST.
2050                               .SBTTL T0044
2051                               ;*****
2052 010450 000004      T0044: SCOPE
2053 010452 012706 001000      MOV      #1000,SP      ;SETUP THE STACK POINTER
2054 010456 004737 011766      JSR      PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2055 010462 000044              00044                ;HERE LIES THE NUMBER OF THIS TEST
2056 010464 004737 013052      R0044: JSR      PC,LBDAT1  ;SET UP WRITE DATA (256 WORDS).
2057 010470 005077 170544      CLR      @TCCM
2058 010474 012777 024003 170536  MOV      #MAINT!UO!REV.RNUM!DO,@TCCM
2059 010502              MTCOD      MTK7,5
2060 010502 004537 012674      JSR      R5,LMTCOD     ;CALL LOAD MT CODES SUB.
2061 010506 033306              MTK7
2062 010510 000005              5                    ;ADDRESS OF MARK TRACK CODES.
2063 010512 005777 170522      TST      @TCCM         ;MARK TRACK CODE COUNT.
2064 010516 100002              BPL      A0044         ;ERROR BIT SET?
2065 010520 104041              ERROR 41            ;BR IF NO ERROR.
2066 010522 000432              BR          C0044     ;ERROR BIT SET EXAMINE TCST.
2067 010524 012777 177400 170510  A0044: MOV      #-256,@TCWC   ;-256 TO WORD COUNT.
2068 010532 012777 036430 170504  MOV      #RBUF,@TCBA   ;ADDR OF RBUF TO TCBA.
2069 010540 012704 037430      MOV      #RBUF+512,R4  ;ADDR TO SAVE TCDT DATA TO R4.
2070 010544 112777 000015 170466  MOVB    #WDATA!DO,@TCCM ;ISSUE WDATA COMMAND.
2071 010552              MTCOE      D0044,MTK7A,262.
2072 010552 004537 013044      JSR      R5,LMTCOE     ;CALL LOAD MT CODES SUBROUTINE.
2073 010556 010616              D0044
2074 010560 033344              MTK7A
2075 010562 000406              262.                ;ADDR TO GO AFTER EACH CODE PASSED.
2076 010564 005777 170450      TST      @TCCM         ;ADDRESS OF MARK TRACK CODES.
2077 010570 100002              BPL      B0044         ;MARK TRACK CODE COUNT.
2078 010572 104041              ERROR 41            ;ERROR BIT SET?
2079 010574 000405              BR          C0044     ;BR IF NO ERROR.
2080 010576 004537 013166      B0044: JSR      R5,CKDAT  ;ERROR BIT SET. EXAMINE TCST.
2081 010602 001306              SBDAT3
2082 010604 037430              RBUF+512.
2083 010606 000400              256.
2084 010610              C0044:
2085 010610 012706 001000      MOV      #1000,SP      ;RESTORE THE STACK POINTER
2086 010614 000403              BR          T0045     ;GO ON TO THE NEXT TEST
2087 010616 017724 170424      D0044: MOV      @TCDT,(4)+ ;HERE AFTER EACH MARK CODE IS PASSED.
2088 010622 000002              RTI                   ;SAVE TCDT DATA AND EXIT IOT TRAP.
2089                               ;WRITE ALL TEST.
2090                               .SBTTL T0045
2091                               ;*****
2092 010624 000004      T0045: SCOPE
2093 010626 012706 001000      MOV      #1000,SP      ;SETUP THE STACK POINTER
2094 010632 004737 011766      JSR      PC,TORDER     ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2095 010636 000045              00045                ;HERE LIES THE NUMBER OF THIS TEST
2096 010640 004737 013052      R0045: JSR      PC,LBDAT1  ;SET UP WRITE DATA.
2097 010644 005077 170370      CLR      @TCCM
2098 010650 012777 020003 170362  MOV      #MAINT!UO!FWD!RNUM!DO,@TCCM
2099 010656              MTCOD      MTK7,4
2100 010656 004537 012674      JSR      R5,LMTCOD     ;CALL LOAD MT CODES SUB.
```

```
2101 010662 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.
2102 010664 000004 4 ;MARK TRACK CODE COUNT.
2103 010666 005777 170346 TST @TCM ;ERROR BIT SET?
2104 010672 100002 BPL A0045 ;BR IF NO ERROR.
2105 010674 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
2106 010676 000470 BR G0045
2107 010700 005077 170336 A0045: CLR @TCWC ;0 TO WORD COUNT.
2108 010704 012777 036430 170332 MOV #RBUF,@TCBA ;ADDR OF RBUF TO TCBA.
2109 010712 012703 036426 MOV #RBUF-2,R3
2110 010716 012704 037430 MOV #RBUF+512.,R4
2111 010722 012737 000402 001274 MOV #258.,CTRA ;# OF WORDS TO WRITE TO CTRA.
2112 010730 004737 012174 JSR PC,STICV ;SET INTERRUPT VECTOR TO IIC.
2113 010734 010756 B0045
2114 010736 112777 000117 170274 MOVB #WALL!IE!DO,@TCM;ISSUE WRITE ALL COMMAND. INTERRUPT ENABLED.
2115 010744 MTCOE I0045,MTK5,260.
2116 010744 004537 013044 JSR R5,LMTCOE ;CALL LOAD MT CODES SUBROUTINE.
2117 010750 011066 I0045 ;ADDR TO GO AFTER EACH CODE PASSED.
2118 010752 033336 MTK5 ;ADDRESS OF MARK TRACK CODES.
2119 010754 000404 260. ;MARK TRACK CODE COUNT.
2120 010756 005777 170256 B0045: TST @TCM ;ERROR BIT SET?
2121 010762 100002 BPL B0045A ;BR IF NO ERROR.
2122 010764 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
2123 010766 000434 BR G0045
2124 010770 012377 170252 B0045A: MOV (3)+,@TCDT ;WRITE DATA TO TCDT.
2125 010774 005337 001274 DEC CTRA ;WROTE 257 WORDS?
2126 011000 001401 BEQ C0045 ;BR IF 257 WORDS WRITTEN.
2127 011002 000002 RTI ;NOT DONE. EXIT INTERRUPT.
2128 011004 005737 037430 C0045: TST @#RBUF+512. ;1ST WORD WRITTEN EQUAL 0?
2129 011010 001404 BEQ D0045 ;BR IF FIRST WORD 0.
2130 011012 005037 001162 CLR $REGO
2131 011016 104070 ERROR 70 ;1ST WORD WRITTEN NOT 0. (REV CHECKSUM).
2132 011020 000417 BR G0045
2133 011022 004537 013166 D0045: JSR R5,CKDAT ;CHECK THAT SAVED TCDT DATA MATCHES
2134 011026 001276 SBDAT1 ;WRITE DATA.
2135 011030 037432 RBUF+514.
2136 011032 000400 256.
2137 011034 005777 170202 TST @TCWC ;WORD COUNT STILL 0?
2138 011040 001402 BEQ F0045 ;BR IF WORD COUNT IS 0.
2139 011042 104071 ERROR 71 ;WORD COUNT MODIFIED DURING WRITE ALL.
2140 011044 000405 BR G0045
2141 011046 022777 036430 170170 F0045: CMP #RBUF,@TCBA ;TCBA STILL EQUAL RBUF?
2142 011054 001401 BEQ G0045 ;BR IF TCBA STILL SAME.
2143 011056 104072 ERROR 72 ;TCBA MODIFIED DURING WRITE ALL.
2144 011060 G0045:
2145 011060 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
2146 011064 000403 BR T0046 ;GO ON TO THE NEXT TEST
2147 011066 017724 170154 I0045: MOV @TCDT,(4)+ ;HERE AFTER EACH MARK CODE IS PASSED.
2148 011072 000002 RTI ;SAVE TCDT DATA AND EXIT IOT TRAP.
2149
2150 .SBTTL T0046
2151 :*****
2152 011074 000004 T0046: SCOPE
2153 011076 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
2154 011102 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2155 011106 000046 00046 ;HERE LIES THE NUMBER OF THIS TEST
2156 011110 004537 013314 R0046: JSR R5,CKSELE ;SST TO U1.
```

```
2157 011114 000400          U1
2158 011116 104073          ERROR 73          ;SST TO U1 DID NOT CAUSE SFLECT ERROR.
2159
2160 011120 012706 001000    MOV      #1000,SP    ;RESTORE THE STACK POINTER
2161 011124 000400          BR        T0047      ;GO ON TO THE NEXT TEST
2162          .SBTTL T0047
2163          :*****
2164 011126 000004          T0047: SCOPE
2165 011130 012706 001000    MOV      #1000,SP    ;SETUP THE STACK POINTER
2166
2167 011134 004737 011766    JSR      PC,TORDER  ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2168 011140 000047          00047          ;HERE LIES THE NUMBER OF THIS TEST
2169 011142 004537 013314    R0047: JSR      R5,CKSELE ;SST TO U2.
2170 011146 001000          U2
2171 011150 104073          ERROR 73          ;SST TO U2 DID NOT CAUSE SELECT ERROR.
2172
2173 011152 012706 001000    MOV      #1000,SP    ;RESTORE THE STACK POINTER
2174          .SBTTL T0050
2175          :*****
2176 011156 000004          T0050: SCOPE
2177 011160 012706 001000    MOV      #1000,SP    ;SETUP THE STACK POINTER
2178 011164 004737 011766    JSR      PC,TORDER  ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2179 011170 000050          00050          ;HERE LIES THE NUMBER OF THIS TEST
2180 011172 004537 013314    R0050: JSR      R5,CKSELE ;SST TO U1.
2181 011176 001400          U3
2182 011200 104073          ERROR 73          ;SST TO U3 DID NOT CAUSE SELECT ERROR.
2183
2184 011202 012706 001000    MOV      #1000,SP    ;RESTORE THE STACK POINTER
2185          .SBTTL T0051
2186          :*****
2187 011206 000004          T0051: SCOPE
2188 011210 012706 001000    MOV      #1000,SP    ;SETUP THE STACK POINTER
2189 011214 004737 011766    JSR      PC,TORDER  ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2190 011220 000051          00051          ;HERE LIES THE NUMBER OF THIS TEST
2191 011222 004537 013314    R0051: JSR      R5,CKSELE ;ISSUE A SST COMMAND
2192 011226 002000          U4
2193 011230 104073          ERROR 73          ;SST TO U4 DID NOT CAUSE SELECT ERROR.
2194
2195 011232 012706 001000    MOV      #1000,SP    ;RESTORE THE STACK POINTER
2196          .SBTTL T0052
2197          :*****
2198 011236 000004          T0052: SCOPE
2199 011240 012706 001000    MOV      #1000,SP    ;SETUP THE STACK POINTER
2200 011244 004737 011766    JSR      PC,TORDER  ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2201 011250 000052          00052          ;HERE LIES THE NUMBER OF THIS TEST
2202 011252 004537 013314    R0052: JSR      R5,CKSELE ;ISSUE A SST COMMAND
2203 011256 002400          U5
2204 011260 104073          ERROR 73          ;SST TO U5 DID NOT CAUSE SELECT ERROR.
2205
2206 011262 012706 001000    MOV      #1000,SP    ;RESTORE THE STACK POINTER
2207          .SBTTL T0053
2208          :*****
2209 011266 000004          T0053: SCOPE
2210 011270 012706 001000    MOV      #1000,SP    ;SETUP THE STACK POINTER
2211 011274 004737 011766    JSR      PC,TORDER  ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2212 011300 000053          00053          ;HERE LIES THE NUMBER OF THIS TEST
```

```
2213 011302 004537 013314 R0053: JSR R5,CKSELE ;ISSUE A SST COMMAND
2214 011306 003000 U6
2215 011310 104073 ERROR 73 ;SST TO U6 DID NOT CAUSE SELECT ERROR.
2216 011312 000240 NOP
2217 011314 000240 NOP
2218 011316 000240 NOP
2219
2220 011320 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
2221 .SBTTL T0054
2222 *****
2223 011324 000004 T0054: SCOPE
2224 011326 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
2225 011332 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2226 011336 000054 00054 ;HERE LIES THE NUMBER OF THIS TEST
2227 011340 004537 013314 R0054: JSR R5,CKSELE ;ISSUE A SST COMMAND
2228 011344 003400 U7
2229 011346 104073 ERROR 73 ;SST TO U7 DID NOT CAUSE SELECT ERROR.
2230
2231 011350 012706 001000 MOV #1000,SP ;RESTORE THE STACK POINTER
2232 .SBTTL T0055
2233 *****
2234 011354 000004 T0055: SCOPE
2235 011356 012706 001000 MOV #1000,SP ;SETUP THE STACK POINTER
2236 011362 004737 011766 JSR PC,TORDER ;MAKE SURE TESTS ARE IN PRPOER SEQUENCE
2237 011366 000055 00055 ;HERE LIES THE NUMBER OF THIS TEST
2238 011370 004737 013104 JSR PC,LBBIND ;LOAD BUFFER WITH BINARY DATA.
2239 011374 005077 167640 CLR @TCCM
2240 011400 012777 020003 167632 MOV #MAINT!UO!FWD!RNUM!DO,@TCCM
2241 011406 MTCOD MTK7,5
2242 011406 004537 012674 JSR R5,LMTCOD ;CALL LOAD MT CODES SUB.
2243 011412 033306 MTK7 ;ADDRESS OF MARK TRACK CODES.
2244 011414 000005 5 ;MARK TRACK CODE COUNT.
2245 011416 005777 167616 TST @TCCM ;ERROR BIT SET?
2246 011422 100002 BPL A0055 ;BR IF NO ERROR.
2247 011424 104041 ERROR 41 ;ERROR BIT SET. EXAMINE TCST.
2248 011426 000441 BR D0055
2249 011430 012777 177400 167604 A0055: MOV #-256,@TCWC ;-256 TO WORD COUNT.
2250 011436 012777 036430 167600 MOV #RBUF,@TCBA ;RBUF ADDR TO TCBA.
2251 011444 012704 037430 MOV #RBUF+512.,R4 ;ADDR TO SAVE TCDT DATA TO R4.
2252 011450 112777 000015 167562 MOVB #WDATA!DO,@TCCM ;ISSUE WRITE DATA COMMAND.
2253 011456 MTCOE F0055,MTK7A,262.
2254 011456 004537 013044 JSR R5,LMTCOE ;CALL LOAD MT CODES SUBROUTINE.
2255 011462 011540 F0055 ;ADDR TO GO AFTER EACH CODE PASSED.
2256 011464 033344 MTK7A ;ADDRESS OF MARK TRACK CODES.
2257 011466 000406 262. ;MARK TRACK CODE COUNT.
2258 011470 005777 167544 TST @TCCM ;ERROR BIT SET?
2259 011474 100002 BPL B0055 ;BR IF NO ERROR.
2260 011476 104044 ERROR 44 ;ERROR BIT SET. EXAMINE TCST.
2261 011500 000414 BR D0055
2262 011502 012701 037430 B0055: MOV #RBUF+512.,R1 ;ADDR OF DATA TO CHECK TO R1.
2263 011506 012703 036430 MOV #RBUF,R3 ;ADDR OF EXPECTED DATA TO R3.
2264 011512 012702 000400 MOV #256.,R2 ;# OF WORDS TO CHECK TO R2.
2265 011516 005037 013312 CLR WDCNT
2266 011522 004737 013240 C0055: JSR PC,CDTCK ;CHECK DATA WORD.
2267 011526 005302 DEC R2 ;ALL WORDS CHECKED?
2268 011530 001374 BNE C0055 ;BR IF NOT DONE YET.
```

```
2269 011532 D0055:
2270 011532 012706 001000      MOV    #1000,SP      ;RESTORE THE STACK POINTER
2271 011536 000403              BR     T0056        ;GO ON TO THE NEXT TEST
2272 011540 017724 167502      F0055: MOV    @TCDT,(4)+ ;HERE AFTER EACH MARK CODE IS PASSED.
2273 011544 000002              RTI                ;SAVE TCDT DATA AND EXIT.
2274
2275      .SBTTL T0056
2276 011546 000004      T0056: SCOPE
2277 011550      .SEOP STARTX,,PASCNT
2278      .SBTTL END OF PASS ROUTINE
2279
2280 011550      STARS
2281      ;*****
2282      ;*INCREMENT THE PASS NUMBER ($PASS)
2283      ;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
2284      ;*IF THERES A MONITOR GO TO IT
2285      ;*IF THERE ISN'T JUMP TO STARTX
2286
2287 011550      $EOP:
2288 011550 000004      SCOPE
2289      .LIST
2290 011552 005037 001102      CLR    $STNM        ;;ZERO THE TEST NUMBER
2291 011556 005037 001222      CLR    $TIMES       ;;ZERO THE NUMBER OF ITERATIONS
2292 011562 005237 001100      INC    $PASS        ;;INCREMENT THE PASS NUMBER
2293 011566 042737 100000 001100 BIC    #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
2294 011574 005327              DEC    (PC)+        ;;LOOP?
2295 011576 000001      $EOPCT: .WORD 1
2296 011600 003022      BGT    $DOAGN       ;;YES
2297 011602 012737      MOV    (PC)+,@(PC)+ ;;RESTORE COUNTER
2298 011604 000001      $ENDCT: .WORD 1
2299 011606 011576      $EOPCT
2300 011610 104401 011655      TYPE   , $ENDMG     ;;TYPE 'END PASS #'
2301 011614      TYPDEC $PASS
2302 011614 013746 001100      MOV    $PASS,-(SP) ;;SAVE $PASS FOR TYPEOUT
2303 011620 104405      TYPDS              ;;GO TYPE--DECIMAL ASCII WITH SIGN
2304 011622 104401 011652      TYPE   , $ENULL    ;;TYPE A NULL CHARACTER
2305 011626 013700 000042      $GET42: MOV    @#42,R0 ;;GET MONITOR ADDRESS
2306 011632 001405      BEQ    $DOAGN      ;;BRANCH IF NO MONITOR
2307      .LIST
2308 011634 000005      RESET
2309 011636 004710      $ENDAD: JSR    PC,(R0) ;;CLEAR THE WORLD
2310 011640 000240      NOP                    ;;GO TO MONITOR
2311 011642 000240      NOP                    ;;SAVE ROOM
2312 011644 000240      NOP                    ;;FOR
2313 011646      NOP                    ;;ACT11
2314 011646 000137      $DOAGN: JMP    @(PC)+        ;;RETURN
2315 011650 002632      $RTNAD: .WORD  STARTX
2316 011652 377 377 000      $ENULL: .BYTE  -1,-1,0 ;;NULL CHARACTER STRING
2317 011655 015 042412 042116 $ENDMG: .ASCIZ <15><12>/END PASS #/
2318 011662 050040 051501 020123
2319 011670 000043
2320      ;* THIS ROUTINE HANDLES FATAL TRAP ERRORS
2321 011672 012737 016041 011730 TRAP10: MOV    #TRPM10,TMESAD ;ADDRESS OF TRAP TO 10 MESSAGE TO THE MESSAGE PO
2322 011700 000403              BR     TRAPX        ;ENTER THE FATAL TRAP ERROR REPORT ROUTINE
2323 011702 012737 015717 011730 TRAP4:  MOV    #TRPM4S,TMESAD
2324 011710 011600      TRAPX: MOV    (SP),R0 ;SAVE PC OF WHERE TRAP OCCURRED
```

```
2325 011712 162700 000002 SUB #2,R0 ;MAKE IT POINT EXACTLY AT THE OFFENDING WORD
2326 011716 012706 001000 MOV #1000,SP ;MAKE SURE THAT THE STACK GIVES NO PROBLEMS
2327 011722 005046 CLR -(SP) ;FAKEOUT THE PRINTOUT ROUTINE
2328 011724 004737 014266 JSR PC,$TYPE ;PRINT THE TRAP MESSAGE
2329 011730 000000 TMESAD: 000000 ;ADDRESS OF THE TRAP MESSAGE GOES HERE
2330 011732 010016 MOV RO,(SP) ;PUT ERROR PC BACK ONTO THE STACK
2331 011734 104402 TYP0C
2332 011736 104401 001233 TRYAGN: TYPE , $CRLF
2333 011742 104401 015767 TYPE ,TRPMES ;THEN PRINTOUT THE TRAP MESSAGE
2334 011746 000240 NOP ;PATCHING SPACE, THOUGHTFULLY PROVIDED
2335 011750 000240 NOP ;PATCHING SPACE, THOUGHTFULLY PROVIDED
2336 011752 000240 NOP ;PATCHING SPACE, THOUGHTFULLY PROVIDED
2337 011754 000240 NOP ;PATCHING SPACE, THOUGHTFULLY PROVIDED
2338 011756 000240 NOP ;PATCHING SPACE, THOUGHTFULLY PROVIDED
2339 011760 000005 RESET
2340 011762 000137 002314 JMP @#START
2341 ;SUBROUTINE TO CHECK TO MAKE SURE THAT EACH TEST IN
2342 ;RUNNING WHEN IT SHOULD
2343 011766 011637 001162 TORDER: MOV (SP),$REG0 ;GET ADDRESS OF TEST #
2344 011772 062716 000002 ADD #2,(SP) ;BUMP PRETURN ADDRESS TO HOP OVER THE IN LINE TEST #
2345 011776 123777 001102 167156 CMPB $TSTNM,@$REG0 ;FIND OUT IF THE TEST #'S MATCH
2346 012004 001410 BEQ TORDER ;IF THEY DO HOP OVER THE ERROR SIGNAL STUFF
2347 012006 013737 001162 001164 MOV $REG0,$REG1 ;SAVE ADDRESS OF TEST
2348 012014 017737 167142 001162 MOV @$REG0,$REG0 ;GET TEST# WAS DATA READY FOR PRINTOUT
2349 012022 104026 ERROR 26 ;PRINTOUT 'OUT OF ORDER' MESSAGE
2350 012024 000744 BR TRYAGN ;GO TRY TO START OVER
2351 012026 000207 TORDER: RTS PC ;RETURN
2352 ;SAVE REGS 0 TO 4 SUBROUTINE.
2353 012030 012666 177764 SV04: MOV (6)+,-12.(6) ;MOVE PC UPSTACK.
2354 012034 012737 000207 012106 MOV #RTSPC,SV05C
2355 012042 000412 BR SV05B
2356 ;SUB TO SAVE REGS 0 TO 5 AND PLACE JSR PC IN R5.
2357 012044 012737 000240 012106 SV05S: MOV #NOP,SV05C
2358 012052 000403 BR SV05A
2359 ;SUB TO SAVE REGS 0 TO 5.
2360 012054 012737 000207 012106 SV05: MOV #RTSPC,SV05C
2361 012062 012666 177762 SV05A: MOV (6)+,-14.(6) ;MOVE PC UPSTACK.
2362 012066 010546 SV05B: MOV R5,-(6)
2363 012070 010446 MOV R4,-(6)
2364 012072 010346 MOV R3,-(6)
2365 012074 010246 MOV R2,-(6)
2366 012076 010146 MOV R1,-(6)
2367 012100 010046 MOV R0,-(6)
2368 012102 162706 000002 SUB #2,SP
2369 012106 000207 SV05C: RTS PC ;RTS PC OR NOP.
2370 012110 016605 000016 MOV 14.(6),R5 ;JSR PC TO R5.
2371 012114 000207 RTS PC ;EXIT.
2372 ;RESTORE REGS 0 TO 4 SUBROUTINE.
2373 012116 062706 000002 RS04: ADD #2,SP
2374 012122 012600 MOV (6)+,R0 ;RESTORE REGS 0 TO 4.
2375 012124 012601 MOV (6)+,R1
2376 012126 012602 MOV (6)+,R2
2377 012130 012603 MOV (6)+,R3
2378 012132 012604 MOV (6)+,R4
2379 012134 016646 177764 MOV -12.(6),-(6) ;MOVE PC DOWN STACK.
2380 012140 000207 RTS PC ;EXIT
```


2381									
2382	012142	010566	000016						
2383									
2384	012146	062706	000002						
2385	012152	012600							
2386	012154	012601							
2387	012156	012602							
2388	012160	012603							
2389	012162	012604							
2390	012164	012605							
2391	012166	016646	177762						
2392	012172	000207							
2393									
2394	012174	0C4737	012044						
2395	012200	013701	001250						
2396	012204	012521							
2397	012206	013721	001252						
2398	012212	004737	012142						
2399	012216	000207							
2400									
2401	012220	010046							
2402	012222	012700	052525						
2403	012226	005100							
2404	012230	010037	012224						
2405	012234	000005							
2406	012236	012600							
2407	012240	000207							
2408	012242	004537	012646						
2409	012246	033250							
2410	012250	033360							
2411	012252	000006							
2412	012254	004537	012646						
2413	012260	033300							
2414	012262	033336							
2415	012264	000006							
2416	012266	004537	012646						
2417	012272	036411							
2418	012274	036352							
2419	012276	000006							
2420	012300	000207							
2421									
2422	012302	004737	012044						
2423	012306	010500							
2424	012310	005740							
2425	012312	000000							
2426	012314	004737	012142						
2427	012320	000207							
2428									
2429	012322	013700	012370						
2430	012326	006100							
2431	012330	006100							
2432	012332	063700	012372						
2433	012336	010037	012370						
2434	012342	006100							
2435	012344	006100							
2436	012346	063700	012372						

;SUB TO SET R5 IN EMT PC AND RESTORE REGS 0 TO 5.
RS05S: MOV R5,14.(6) ;SET EMT PC TO R5 CONTENTS.

;SUB TO RESTORE REGS 0 TO 5.
RS05: ADD #2,SP

MOV (6)+,R0

MOV (6)+,R1

MOV (6)+,R2

MOV (6)+,R3

MOV (6)+,R4

MOV (6)+,R5

MOV -14.(6),-(6) ;MOVE PC DOWNSTACK.

RTS PC ;EXIT

;ROUTINE TO SET TC11 INTERRUPT VECTOR AND PRIORITY

STTCV: JSR PC,SV05S

MOV TCVTR,R1 ;VECTOR TO R1.

MOV (5)+,(1)+ ;SET DESIRED VECTOR.

MOV TCLVL,(1)+ ;SET TC11 PRIORITY.

JSR PC,RS05S

RTS PC

;ROUTINE TO ISSUE RESET.

SRSETT: MOV R0,-(6) ;PUSH R0.

MOV #52525,R0 ;DATA TO R0.

COM R0 ;COMPLEMENT (R0).

MOV R0,SRSETT+4 ;(R0) TO SRSETT+4.

RESET ;ISSUE RESET. (R0) IS

MOV (6)+,R0 ;RESTORE R0.

RTS PC ;EXIT

RSTMTK: JSR R5,BMOVE ;RESTORE MTKVAR MARK CODE.

MTKC10 ;AND GOOD CHECKSUM.

MTKVAR

6 JSR R5,BMOVE

MTK5P

MTK5

6 JSR R5,BMOVE

GCKSM

FCKSM

6

RTS PC ;EXIT.

;COMMON HALT ROUTINE

CHLT: JSR PC,SV05S

MOV R5,R0 ;DEVELOP ADDR OF CALLER.

TST -(0)

HALT ;HALT CALL ADDR IN DATA LIGTHS.

JSR PC,RS05S

RTS PC ;EXIT.

;RANDOM NUMBER GENERATOR. ROUTINE EXITS WITH NUMBER IN REGISTER 0.

RNGEN: MOV RP1,R0

ROL R0

ROL R0

ADD RP2,R0

MOV R0,RP1

ROL R0

ROL R0

ADD RP2,R0

2437	012352	006100			ROL	R0	
2438	012354	006100			ROL	R0	
2439	012356	010037	012372		MOV	R0,RP2	
2440	012362	013700	012370		MOV	RP1,R0	
2441	012366	000207			RTS	PC	;EXIT. NUMBER IN R0
2442	012370	001233			RP1:	1233	
2443	012372	007622			RP2:	7622	
2444					;SUBROUTINE TO DELAY A SPECIFIED NUMBER OF MILLISECONDS		
2445	012374	004737	012044		DLY:	JSR	PC,SV05S
2446	012400	012500				MOV	(5)+,R0
2447	012402	005037	177776			CLR	PSW
2448	012406	012701	000226		DLYA:	MOV	#226,R1
2449	012412	005301			DLYB:	DEC	R1
2450	012414	001376				BNE	DLYB
2451	012416	005300				DEC	R0
2452	012420	001372				BNE	DLYA
2453	012422	004737	012142			JSR	PC,RS05S
2454	012426	000207				RTS	PC
2455					;SUBROUTINE TO STALL A RANDOM NUMBER OF MILLISECONDS. MAXIMUM STALL		
2456					;DETERMINED BY CONTENTS OF LOC STLMSK.		
2457	012430	004737	012044		STAL:	JSR	PC,SV05S
2458	012434	004737	012322			JSR	PC,RNGEN
2459	012440	043700	012466			BIC	STLMSK,R0
2460	012444	001407				BEQ	STALB
2461	012446	010037	012456			MOV	R0,STALA
2462	012452	004737	012374			JSR	PC,DLY
2463	012456	000000					
2464	012460	004737	012142		STALA:	OPEN	
2465	012464	000207				JSR	PC,RS05S
2466	012466	000000			STALB:	RTS	PC
2467					STLMSK:	OPEN	
2468	012470	005037	036430		;SUBROUTINE TO CLEAR DECTAPE READ BUFFER.		
2469	012474	004537	012646		CLRBUF:	CLR	RBUF
2470	012500	036430				JSR	R5,BMOVE
2471	012502	036431				RBUF	
2472	012504	001777				RBUF+1	
2473	012506	000207				1023.	
2474						RTS	PC
2475	012510	012737	177777	012532	;SUBROUTINE TO INITIALIZE BINARY COUNT PATTERNS		
2476	012516	004537	012646		INBIN:	MOV	#-1,RIND
2477	012522	012532				JSR	R5,BMOVE
2478	012524	012533				RIND	
2479	012526	000013				RIND+1	
2480	012530	000207				11.	
2481	012532	000000				RTS	PC
2482	012534	000000			RIND:	OPEN	
2483	012536	000000			PT0:	OPEN	
2484	012540	000000			PT1:	OPEN	
2485	012542	000000			PIND:	OPEN	
2486	012544	000000			PTOP:	OPEN	
2487					PT1P:	OPEN	
2488	012546	013737	012534	012536	;SPECIAL BINARY COUNT PATTERN SUBROUTINE. EXITS WITH BIN CHAR IN R0		
2489	012554	005137	012536		GTBIN:	MOV	PT0,PT1
2490	012560	005137	012532			COM	PT1
2491	012564	001002				COM	RIND
2492	012566	005237	012536			BNE	+6
						INC	PT1

```
2493 012572 013737 012536 012534      MOV      PT1,PT0      ;SAVE BIN CHAR IN PTO
2494 012600 013700 012536      MOV      PT1,R0      ;BIN CHAR TO R0.
2495 012604 000207      RTS      PC          ;EXIT.
2496 012606 013737 012542 012544  GTBINP: MOV      PTOP,PT1P    ;PREVIOUS BIN CHAR TO PT1P
2497 012614 005137 012544      COM      PT1P
2498 012620 005137 012540      COM      PIND
2499 012624 001002      BNE      .+6
2500 012626 005237 012544      INC      PT1P
2501 012632 013737 012544 012542      MOV      PT1P,PTOP    ;SAVE BIN CHAR IN PTO.
2502 012640 013701 012544      MOV      PT1P,R1      ;BIN CHAR TO R1.
2503 012644 000207      RTS      PC          ;EXIT.
2504      ;SUBROUTINE TO MOVE A VARIABLE NUMBER OF BYTES.
2505 012646 004737 012030  BMOVE:  JSR      PC,SV04      ;SAVE REGS.
2506 012652 012501      MOV      (5)+,R1      ;GET 'FROM' ADDRESS
2507 012654 012502      MOV      (5)+,R2      ;GET 'TO' ADDRESS
2508 012656 012503      MOV      (5)+,R3      ;GET COUNT
2509 012660 112122      BMOVA:  MOVB     (1)+,(2)+  ;MOVE BYTE
2510 012662 005303      DEC      R3          ;DECREMENT COUNT
2511 012664 001375      BNE      BMOVA       ;BRANCH IF NOT DONE.
2512 012666 004737 012116      JSR      PC,RS04     ;RESTORE REGS.
2513 012672 000205      RTS      R5          ;DONE EXIT
2514      ;SUB TO PASS TIMING, MARK, AND DATA TO TC11 CONTROL UNDER MAINTENANCE MODE.
2515 012674 005037 001260  LMTCOD: CLR      CODCAL    ;DO NOT CALL CODE AFTER EACH MARK
2516 012700 012537 013034  LMTCAA: MOV      (5)+,MTKADR  ;GET MARK TRACK ADDRESS.
2517 012704 012537 013040      MOV      (5)+,CDCNT    ;GET NTH CODE COUNT.
2518 012710 052777 020000 166322  BIS      #BIT13,@TCCM  ;SET MAINTENANCE BIT.
2519 012716 013737 013040 J13042  MOV      CDCNT,CDCTR  ;CODE COUNT TO CODE COUNTER.
2520 012724 013701 001236      MOV      TCST,R1      ;ADDR CONTAINING TCST ADDR TO R1.
2521 012730 012702 000100      MOV      #10C,R2
2522 012734 013700 013034  LMTCA:  MOV      MTKADR,R0  ;MARK TRACK ADDR TO R0.
2523 012740 012737 000006 013036  LMTCB:  MOV      #6,BTCTR  ;6 TO BIT COUNTER.
2524 012746 111011      LMTCC:  MOVB     (0),(1)   ;SET MARK TRACK BIT AND DATA .
2525 012750 150210      BISB    R2,(0)
2526 012752 111011      MOVB    (0),(1)      ;TP1. LOADS MARK TRACK.
2527 012754 111011      MOVB    (0),(1)      ;RELOAD DATA.
2528 012756 140210      BICB    R2,(0)
2529 012760 112011      MOVB    (0)+,(1)     ;TPO. SHIFTS DATA IN RWB.
2530 012762 005337 013036      DEC      BTCTR       ;6TH BIT SET?
2531 012766 001413      BEQ     LMTCE       ;BR IF 6TH BIT SET.
2532 012770 022737 000002 013036  CMP      #2,BTCTR    ;NOT 6TH. 4TH BIT SET?
2533 012776 001363      BNE     LMTCC       ;BRANCH IF NOT.
2534 013000 005737 001260      TST     CODCAL      ;DO WE WANT TO CALL CODE
2535 013004 001760      BEQ     LMTCC       ;DO NOT IF CODE CALL SWITCH = 0
2536 013006 005046      CLR     -(6)        ;IF ITS NOT =0 FAKE A INTERRUPT
2537 013010 004777 166244      JSR     PC,@CODCAL  ;TO LOCATION SPECIFIED IN CODE CALL SWITCH
2538 013014 000754      BR      LMTCC
2539 013016 005337 013042  LMTCE:  DEC      CDCTR     ;NTH CODE SET?
2540 013022 001001      BNE     LMTCD       ;BRANCH IF NOT.
2541 013024 000205      RTS     R5          ;EXIT.
2542 013026 105710      LMTCD:  TSTB    @R0     ;LAST CODE?
2543 013030 100343      BPL     LMTCB       ;BRANCH IF NOT LAST CODE.
2544 013032 000740      BR      LMTCA       ;LAST CODE.
2545 013034 000000      MTKADR: OPEN
2546 013036 000000      BTCTR:  OPEN
2547 013040 000000      CDCNT:  OPEN
2548 013042 000000      CDCTR:  OPEN
```

2549	013044	012537	001260	LMTCOE:	MOV	(5)+,CODCAL	:SAVE ADDRESS TO GO TO AFTER EACH MARK	
2550	013050	000713			BR	LMTCAA		
2551	013052	004737	012470	LBDAT1:	JSR	PC,CLPSUF	:CLEAR BUFFER AREA.	
2552	013056	004537	012646		JSR	R5,BMOVE	:LOAD 256 WORD BUFFER WITH SBDAT1 DATA.	
2553	013062	001276			SBDAT1			
2554	013064	036430			RBUF			
2555	013066	000004			4			
2556	013070	004537	012646		JSR	R5,BMOVE		
2557	013074	036430			RBUF			
2558	013076	036434			RBUF+4			
2559	013100	000774			508.			
2560	013102	000207			RTS	PC	:EXIT.	
2561	013104	004737	012030	LBBIND:	JSR	PC,SV04		
2562	013110	004737	012470		JSR	PC,CLRBUF	:CLEAR BUFFER AREA AND FILL	
2563	013114	012704	036430		MOV	#RBUF,R4	:256 WORD BUFFER WITH BINARY DATA.	
2564	013120	012737	000400	013150	MOV	#256.,CTRLB		
2565	013126	004737	012546	LBINDA:	JSR	PC,GTBIN	:GET BINARY WORD.	
2566	013132	010024			MOV	R0,(4)+	:STORE PER R4.	
2567	013134	005337	013150		DEC	CTRLB	:256 WORDS STORED?	
2568	013140	001372			BNE	LBINDA	:BR IF NOT DONE YET.	
2569	013142	004737	012116		JSR	PC,RS04		
2570	013146	000207			RTS	PC	:DONE. EXIT.	
2571	013150	000000		CTRLB:	OPEN			
2572	013152	004537	012646	MBCKSM:	JSR	R5,BMOVE	:BAD CHECKSUM TO FCKSM.	
2573	013156	036417			BCKSM			
2574	013160	036352			FCKSM			
2575	013162	000006			6			
2576	013164	000207			RTS	PC	:EXIT.	
2577	013166	004737	012030	CKDAT:	JSR	PC,SV04	:SAVE REGS.	
2578	013172	012537	013306		MOV	(5)+,SBDAT	:ADDR OF S/B DATA TO SBDAT.	
2579	013176	012501			MOV	(5)+,R1	:ADDR OF DATA TO R1.	
2580	013200	012502			MOV	(5)+,R2	:WORD COUNT TO R2.	
2581	013202	005037	013312		CLR	WDCNT	:CLEAR # OF WORD BEING CHECKED.	
2582	013206	013703	013306	CKDTA:	MOV	SBDAT,R3	:ADDR OF S/B DATA TO R3.	
2583	013212	004737	013240		JSR	PC,CDTCK	:GO CHECK DATA.	
2584	013216	005302			DEC	R2	:LAST WORD CHECKED?	
2585	013220	001404			BEQ	CKDTB	:BR IF LAST WORD CHECKED.	
2586	013222	004737	013240		JSR	PC,CDTCK	:CHECK ANOTHER WORD.	
2587	013226	005302			DEC	R2	:LAST WORD CHECKED?	
2588	013230	001366			BNE	CKDTA	:BR IF NOT LAST WORD.	
2589	013232	004737	012116	CKDTB:	JSR	PC,RS04	:RESTORE REGS.	
2590	013236	000205			RTS	R5	:EXIT.	
2591	013240	010137	013310	CDTCK:	MOV	R1,DATADR	:ADDR OF DATA TO DATADR.	
2592	013244	012337	001270		MOV	(3)+,CRBUF	:S/B WORD TO CRBUF.	
2593	013250	012137	001272		MOV	(1)+,CRBUFA	:WAS WORD TO CRBUFA.	
2594	013254	005237	013312		INC	WDCNT	:INCREMENT WORD NUMBER.	
2595	013260	023737	001270	001272	CMP	CRBUF,CRBUFA	:COMPARE S/B AND WAS DATA.	
2596	013266	001001			BNE	CDTCKA	:BR IF DATA NOT SAME.	
2597	013270	000207			RTS	PC	:EXIT.	
2598	013272	016737	000004	013304	CDTCKA:	MOV	4(PC),MLPC	:MAKE MAIN LINE PC ACCESSABLE
2599								
2600	013300	104035			ERROR	35	:PRINTOUT ERROR MESSAGE	
2601	013302	000207			RTS	PC	:EXIT	
2602	013304	000000		MLPC:	000000			
2603	013306	000000		SBDAT:	OPEN			
2604	013310	000000		DATADR:	OPEN			

```
2605 013312 000000 WDCNT: OPEN
2606 013314 012537 015364 CKSELE: MOV (S)+,CKSELT ;UNIT # TO CKSELT.
2607 013320 052737 000011 015364 BIS #SST!DO,CKSELT
2608 013326 013777 015364 165704 MOV CKSELT,@TCCM ;ISSUE SST TO DESIRED UNIT.
2609 013334 105777 165700 TSTB @TCCM ;WAIT FOR READY.
2610 013340 100375 BPL .-4
2611 013342 032777 004000 165666 BIT #BIT11,@TCST ;SELECT ERROR SET?
2612 013350 001001 BNE .+4 ;BR IF SELECT ERROR SET.
2613 013352 000205 RTS R5 ;ERROR EXIT. SELECT ERROR SHOULD BE SET.
2614 013354 062705 000006 ADD #6,R5
2615 013360 000205 RTS R5 ;OK EXIT.
2616 013362 .SSCOPE 4,SCOMAC
2617 .SBTTL SCOPE HANDLER ROUTINE
2618
2619 013362 STARS
2620 ;:*****
2621 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
2622 ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
2623 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
2624 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2625 ;*SW14=1 LOOP ON TEST
2626 ;*SW11=1 INHIBIT ITERATIONS
2627 ;*SW09=1 LOOP ON ERROR
2628 .LIST
2629 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
2630 ;*CALL
2631 ;* SCOPE ;:SCOPE=IOT
2632
2633 013362 $SCOPE:
2634 .IRP NEWINS,<SCOMAC>
2635 NEWINS
2636 .ENDM
2637 013362 SCOMAC
2638 013362 005077 165652 CLR @TCCM
2639 013366 005066 000002 CLR 2(SP) ;PS TO =0 AFTER WE EXIT THE SCOPE ROUTINE
2640 013372 004737 012220 JSR PC,SRSETT
2641 013376 004737 012242 JSR PC,RSTMTK
2642 013402 032777 040000 165530 1$: BIT #BIT14,@SWR ;:LOOP ON PRESENT TEST?
2643 013410 001111 BNE $OVER ;:YES IF SW14=1
2644 ;:*****START OF CODE FOR THE XOR TESTER*****
2645 013412 000416 $XTSTR: BR 6$ ;:IF RUNNING ON THE 'XOR' TESTER CHANGE
2646 ;:THIS INSTRUCTION TO A 'NOP' (NOP=240)
2647 013414 013746 000004 MOV @#ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
2648 013420 012737 013440 000004 MOV #5,@#ERRVEC ;:SET FOR TIMEOUT
2649 013426 005737 177060 TST @#177060 ;:TIME OUT ON XOR?
2650 013432 012637 000004 MOV (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
2651 013436 000463 BR $SVLAD ;:GO TO THE NEXT TEST
2652 013440 022626 5$: CMP (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
2653 013442 012637 000004 MOV (SP)+,@#ERRVEC ;:RESTORE THE ERROR VECTOR
2654 013446 000423 BR 7$ ;:LOOP ON THE PRESENT TEST
2655 013450 6$:;*****END OF CODE FOR THE XOR TESTER*****
2656 013450 032777 000400 165462 BIT #BIT08,@SWR ;:LOOP ON SPEC. TEST?
2657 013456 001404 BEQ 2$ ;:BR IF NO
2658 013460 127737 165454 001102 CMPB @SWR,$STNM ;:ON THE RIGHT TEST? SWR<7:0>
2659 013466 001462 BEQ $OVER ;:BR IF YES
2660 013470 105737 001103 2$: TSTB $ERFLG ;:HAS AN ERROR OCCURRED?
```

```
2661 013474 001421 BEQ 3$ ;;BR IF NO
2662 013476 123737 001115 001103 CMPB $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
2663 013504 101015 BHI 3$ ;;BR IF NO
2664 013506 032777 001000 165424 EIT #BIT09,@SWR ;;LOOP ON ERROR?
2665 013514 001404 BEQ 4$ ;;BR IF NO
2666 013516 013737 001110 001106 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
2667 013524 000443 BR $OVER
2668 013526 105037 001103 4$: CLR $ERFLG ;;ZERO THE ERROR FLAG
2669 013532 005037 001222 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
2670 013536 000415 BR 1$ ;;ESCAPE TO THE NEXT TEST
2671 013540 032777 004000 165372 3$: BIT #BIT11,@SWR ;;INHIBIT ITERATIONS?
2672 013546 001011 BNE 1$ ;;BR IF YES
2673 013550 005737 001100 TST $PASS ;;IF FIRST PASS OF PROGRAM
2674 013554 001406 BEQ 1$ ;; INHIBIT ITERATIONS
2675 013556 005237 001104 INC $ICNT ;;INCREMENT ITERATION COUNT
2676 013562 023737 001222 001104 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
2677 013570 002021 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
2678 013572 012737 000001 001104 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
2679 013600 013737 013650 001222 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
2680 013606 105237 001102 $SVLAD: INCB $TSTNM ;;COUNT TEST NUMBERS
2681 013612 011637 001106 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
2682 013616 011637 001110 MOV (SP),$LPERR ;;SAVE ERROR LOOP ADDRESS
2683 013622 005037 001224 CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
2684 013626 112737 000001 001115 MOVB #1,$ERMAX ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2685 013634 013777 001102 165300 $OVER: MOV $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
2686 013642 013716 001106 MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
2687 013646 000002 RTI ;;FIXES PS
2688 013650 000004 $MXCNT: 4 ;;MAX. NUMBER OF ITERATIONS
2689 .MACRO SAVE
2690 MOV SP,$REG6
2691 SUB #4,$REG6
2692 MOV 2(SP),$REG7
2693 CLR $REG5
2694 MOVB $TSTNM,$REG5
2695 MOV @TCCM,$REG2
2696 MOV @TCST,$REG1
2697 MOV @TCBA,$REG3
2698 .ENDM SAVE
2699 013652 $ERROR $ERRTYP,SAVE
2700 .SBTTL ERROR HANDLER ROUTINE
2701
2702 013652 STARS
2703 ;:*****
2704 ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
2705 ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
2706 ;*AND GO TO $ERRTYP ON ERROR
2707 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2708 ;*SW15=1 HALT ON ERROR
2709 ;*SW13=1 INHIBIT ERROR TYPEOUTS
2710 ;*SW10=1 BELL ON ERROR
2711 ;*SW09=1 LOOP ON ERROR
2712 ;*CALL
2713 ;* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER
2714
2715 013652 $ERROR:
2716 .IRP NEWINS,<SAVE>
```

```
2717 NEWINS
2718 .ENDM
2719 013652 SAVE
2720 013652 010637 001176 MOV SP,$REG6
2721 013656 162737 000004 001176 SUB #4,$REG6
2722 013664 016637 000002 001200 MOV 2(SP),$REG7
2723 013672 005037 001174 CLR $REG5
2724 013676 113737 001102 001174 MOVB $STNM,$REG5
2725 013704 017737 165330 001166 MOV @TCCM,$REG2
2726 013712 017737 165320 001164 MOV @TCST,$REG1
2727 013720 017737 165320 001170 MOV @TCBA,$REG3
2728 013726 105237 001103 7$: INCB $ERFLG ;;SET THE ERROR FLAG
2729 013732 001775 BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
2730 013734 013777 001102 165200 MOV $STNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
2731 013742 032777 002000 165170 BIT #BIT10,@SWR ;;BELL ON ERROR?
2732 013750 001402 BEQ 1$ ;;NO - SKIP
2733 013752 104401 001226 TYPE $BELL ;;RING BELL
2734 013756 005237 001112 1$: INC $ERTL ;;COUNT THE NUMBER OF ERRORS
2735 013762 011637 001116 MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
2736 013766 162737 000002 001116 SUB #2,$ERRPC
2737 013774 117737 165116 001114 MOVB @SERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
2738 014002 032777 020000 165130 BIT #BIT13,@SWR ;;SKIP TYPEOUT IF SET
2739 014010 001004 BNE 20$ ;;SKIP TYPEOUTS
2740 014012 004737 015160 JSR PC,$ERRTY? ;;GO TO USER ERROR ROUTINE
2741 014016 104401 001233 TYPE $CRLF
2742 014022 20$:
2743 014022 005777 165112 2$: TST @SWR ;;HALT ON ERROR
2744 014026 100001 BPL 3$ ;;SKIP IF CONTINUE
2745 014030 000000 HALT ;;HALT ON ERROR!
2746 014032 032777 001000 165100 3$: BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?
2747 014040 001402 BEQ 4$ ;;BR IF NO
2748 014042 013716 001110 MOV $LPERR,(SP) ;;FUDGE RETURN FOR LOOPING
2749 014046 005737 001224 4$: TST $ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS
2750 014052 001402 BEQ 5$ ;;BR IF NONE
2751 014054 013716 001224 MOV $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE
2752 014060 5$:
2753 .LIST
2754 014060 022737 011636 000042 CMP #SENDAD,@#42 ;;ACT-11 AUTO-ACCEPT?
2755 014066 001001 BNE 6$ ;;BRANCH IF NO
2756 014070 000000 HALT ;;YES
2757 014072 6$:
2758 014072 000002 RTI ;;RETURN
2759 014074 .SPOWER <<POWPUS>,<POWPOP>,<POWMES>>
2760 .SBTTL POWER DOWN AND UP ROUTINES
2761
2762 014074 STARS
2763 *****
2764 :POWER DOWN ROUTINE
2765 014074 012737 014250 000024 $PWRDN: MOV #SILLUP,@PWRVEC ;;SET FOR FAST UP
2766 014102 012737 000340 000026 MOV #340,@PWRVEC+2 ;;PRIO:7
2767 014110 PUSH <R0,R1,R2,R3,R4,R5>
2768 .IRP B,<R0,R1,R2,R3,R4,R5>
2769 MOV B,-(SP) ;;PUSH B ON STACK
2770 .ENDM
2771 014110 010046 MOV R0,-(SP) ;;PUSH R0 ON STACK
2772 014112 010146 MOV R1,-(SP) ;;PUSH R1 ON STACK
```

```
2773 014114 010246      MOV     R2,-(SP)      ;;PUSH R2 ON STACK
2774 014116 010346      MOV     R3,-(SP)      ;;PUSH R3 ON STACK
2775 014120 010446      MOV     R4,-(SP)      ;;PUSH R4 ON STACK
2776 014122 010546      MOV     R5,-(SP)      ;;PUSH R5 ON STACK
2777 014124                PUSH    <<POWPUS>,<POWPOP>,<POWMES>>
2778                .IRP   B,<<POWPUS>,<POWPOP>,<POWMES>>
2779                MOV     B,-(SP)      ;;PUSH B ON STACK
2780                .ENDM
2781 014124 013746 001312      MOV     POWPUS,-(SP)  ;;PUSH POWPUS ON STACK
2782 014130 013746 001314      MOV     POWPOP,-(SP)  ;;PUSH POWPOP ON STACK
2783 014134 013746 016172      MOV     POWMES,-(SP)  ;;PUSH POWMES ON STACK
2784 014140                PUSH    @SWR
2785                .IRP   B,<@SWR>
2786                MOV     B,-(SP)      ;;PUSH B ON STACK
2787                .ENDM
2788 014140 017746 164774      MOV     @SWR,-(SP)    ;;PUSH @SWR ON STACK
2789 014144 010637 014254      MOV     SP,$SAVR6    ;;SAVE SP
2790 014150 012737 014162 000024      MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
2791 014156 000000                HALT
2792 014160 000776                BR     .-2           ;;HANG UP
2793
2794 014162                STARS
2795                ;*****
2796                ;POWER UP ROUTINE
2797 014162 012737 014250 000024      $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
2798 014170 013706 014254                MOV     $SAVR6,SP      ;;GET SP
2799 014174 005037 014254                CLR     $SAVR6        ;;WAIT LOOP FOR THE TTY
2800 014200 005237 014254      1$:   INC     $SAVR6        ;;WAIT FOR THE INC
2801 014204 001375                BNE    1$             ;;OF <POWPUS>,<POWPOP>,<POWMES> WORD
2802 014206                POP     @SWR
2803                .IRP   B,<@SWR>
2804                MOV     (SP)+,B      ;;POP STACK INTO B
2805                .ENDM
2806 014206 012677 164726      MOV     (SP)+,@SWR    ;;POP STACK INTO @SWR
2807 014212                POP     <R5,R4,R3,R2,R1,R0>
2808                .IRP   B,<R5,R4,R3,R2,R1,R0>
2809                MOV     (SP)+,B      ;;POP STACK INTO B
2810                .ENDM
2811                MOV     (SP)+,R5      ;;POP STACK INTO R5
2812                MOV     (SP)+,R4      ;;POP STACK INTO R4
2813                MOV     (SP)+,R3      ;;POP STACK INTO R3
2814                MOV     (SP)+,R2      ;;POP STACK INTO R2
2815                MOV     (SP)+,R1      ;;POP STACK INTO R1
2816                MOV     (SP)+,R0      ;;POP STACK INTO R0
2817 014226 012737 014074 000024      MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
2818 014234 012737 000340 000026      MOV     #340,@#PWRVEC+2 ;;PRIO:7
2819 014242 104401                TYPE    $POWER        ;;REPORT THE POWER FAILURE
2820 014244 014256      $PWRMG: .WORD    $POWER    ;;POWER FAIL MESSAGE POINTER
2821 014246 000002                RTI
2822 014250 000000      $ILLUP: HALT
2823 014252 000776                BR     .-2           ;;THE POWER UP SEQUENCE WAS STARTED
2824 014254 000000                $SAVR6: 0             ;;BEFORE THE POWER DOWN WAS COMPLETE
2825 014256 005015 047520 042527      $POWER: .ASCIZ  <15><12>'POWER' ;;PUT THE SP HERE
2826 014264 000122
2827
2828 014266                .$TYPE .EVEN
```

```
2829 .SBTTL TYPE ROUTINE
2830
2831 014266
2832 STARS
2833 ;*****
2834 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
2835 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
2836 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
2837 ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
2838 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
2839 ;*
2840 ;*CALL:
2841 ;*1) USING A TRAP INSTRUCTION
2842 ;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
2843 ;*OR
2844 ;* TYPE
2845 ;* MESADR
2846 ;*
2847 014266 105737 001157 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
2848 014272 100002 BPL 1$ ;; BR IF YES
2849 014274 000000 HALT ;; HALT HERE IF NO TERMINAL
2850 014276 000407 BR 3$ ;; LEAVE
2851 014300 010046 1$: MOV R0,-(SP) ;; SAVE R0
2852 014302 017600 000002 MOV @2(SP),R0 ;; GET ADDRESS OF ASCIZ STRING
2853 014306 112046 2$: MOVB (R0)+,-(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
2854 014310 001005 BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
2855 014312 005726 TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
2856 014314 012600 60$: MOV (SP)+,R0 ;; RESTORE R0
2857 014316 062716 3$: ADD #2,(SP) ;; ADJUST RETURN PC
2858 014322 000002 RTI ;; RETURN
2859 014324 122716 000011 4$: CMPB #HT,(SP) ;; BRANCH IF <HT>
2860 014330 001430 BEQ 8$
2861 014332 122716 000200 CMPB #CRLF,(SP) ;; BRANCH IF NOT <CRLF>
2862 014336 001006 BNE 5$
2863 014340 005726 TST (SP)+ ;; POP <CR><LF> EQUIV
2864 014342 104401 TYPE ;; TYPE A CR AND LF
2865 014344 001233 $CRLF
2866 014346 105037 014502 CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
2867 014352 000755 BR 2$ ;; GET NEXT CHARACTER
2868 014354 004737 014436 5$: JSR PC,$TYPEC ;; GO TYPE THIS CHARACTER
2869 014360 123726 001156 6$: CMPB $FILLC,(SP)+ ;; IS IT TIME FOR FILLER CHARS.?
2870 014364 001350 BNE 2$ ;; IF NO GO GET NEXT CHAR.
2871 014366 013746 001154 MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
2872 ;; AND THE NULL CHAR.
2873 014372 105366 000001 7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
2874 014376 002770 BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
2875 014400 004737 014436 JSR PC,$TYPEC ;; GO TYPE A NULL
2876 014404 105337 014502 DECB $CHARCNT ;; DO NOT COUNT AS A COUNT
2877 014410 000770 BR 7$ ;; LOOP
2878
2879 ;HORIZONTAL TAB PROCESSOR
2880
2881 014412 112716 000040 8$: MOVB #' ,(SP) ;; REPLACE TAB WITH SPACE
2882 014416 004737 014436 9$: JSR PC,$TYPEC ;; TYPE A SPACE
2883 014422 132737 000007 014502 BITB #7,$CHARCNT ;; BRANCH IF NOT AT
2884 014430 001372 BNE 9$ ;; TAB STOP
```



```
2885 014432 005726          TST      (SP)+          ;;POP SPACE OFF STACK
2886 014434 000724          BR       2$             ;;GET NEXT CHARACTER
2887 014436 105777 164506   $TYPEC: TSTB @STPS      ;;WAIT UNTIL PRINTER IS READY
2888 014442 100375          BPL     $TYPEC
2889 014444 116677 000002 164500  MOVB    2(SP),@STPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2890 014452 122766 000015 000002  CMPB    #CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
2891 014460 001003          BNE     1$             ;;BRANCH IF NO
2892 014462 105037 014502  CLR     $CHARCNT       ;;YES--CLEAR CHARACTER COUNT
2893 014466 000406          BR      $TYPEX        ;;EXIT
2894 014470 122766 000012 000002 1$:  CMPB    #LF,2(SP)      ;;IS CHARACTER A LINE FEED?
2895 014476 001402          BEQ     $TYPEX        ;;BRANCH IF YES
2896 014500 105227          INCB    (PC)+         ;;COUNT THE CHARACTER
2897 014502 000000          $CHARCNT: .WORD 0     ;;CHARACTER COUNT STORAGE
2898 014504 000207          $TYPEX: RTS          PC
2899
2900 014506          .STYPDEC
2901          .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
2902
2903 014506          STARS
2904          ;*****
2905          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
2906          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
2907          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
2908          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
2909          ;*REPLACED WITH SPACES.
2910          ;*CALL:
2911          ;*      MOV      NUM,-(SP)          ;;PUT THE BINARY NUMBER ON THE STACK
2912          ;*      TYPDS                    ;;GO TO THE ROUTINE
2913
2914 014506          $TYPDS: PUSH    <R0,R1,R2,R3,R5>
2915          .IRP     B,<R0,R1,R2,R3,R5>
2916          MOV     B,-(SP)                    ;;PUSH B ON STACK
2917          .ENDM
2918 014506 010046          MOV     R0,-(SP)          ;;PUSH R0 ON STACK
2919 014510 010146          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
2920 014512 010246          MOV     R2,-(SP)          ;;PUSH R2 ON STACK
2921 014514 010346          MOV     R3,-(SP)          ;;PUSH R3 ON STACK
2922 014516 010546          MOV     R5,-(SP)          ;;PUSH R5 ON STACK
2923 014520 012746 020200  MOV     #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
2924 014524 016605 000020  MOV     20(SP),R5        ;;GET THE INPUT NUMBER
2925 014530 100004          BPL     1$             ;;BR IF INPUT IS POS.
2926 014532 005405          NEG     R5             ;;MAKE THE BINARY NUMBER POS.
2927 014534 112766 000055 000001  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
2928 014542 005000          CLR     R0             ;;ZERO THE CONSTANTS INDEX
2929 014544 012703 014722  MOV     #$DBLK,R3        ;;SETUP THE OUTPUT POINTER
2930 014550 112723 000040  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
2931 014554 005002          CLR     R2             ;;CLEAR THE BCD NUMBER
2932 014556 016001 014712  MOV     $DTBL(R0),R1     ;;GET THE CONSTANT
2933 014562 160105          SUB     R1,R5          ;;FORM THIS BCD DIGIT
2934 014564 002402          BLT    4$             ;;BR IF DONE
2935 014566 005202          INC     R2             ;;INCREASE THE BCD DIGIT BY 1
2936 014570 000774          BR     3$
2937 014572 060105          4$:  ADD     R1,R5          ;;ADD BACK THE CONSTANT
2938 014574 005702          TST    R2             ;;CHECK IF BCD DIGIT=0
2939 014576 001002          BNE    5$             ;;FALL THROUGH IF 0
2940 014600 105716          TSTB   (SP)           ;;STILL DOING LEADING 0'S?
```

```
2941 014602 100407
2942 014604 106316
2943 014606 103003
2944 014610 116663 000001 177777
2945 014616 052702 000060
2946 014622 052702 000040
2947 014626 110223
2948 014630 005720
2949 014632 020027 000010
2950 014636 002746
2951 014640 003002
2952 014642 010502
2953 014644 000764
2954 014646 105726
2955 014650 100003
2956 014652 116663 177777 177776
2957 014660 105013
2958 014662
2959
2960
2961
2962 014662 012605
2963 014664 012603
2964 014666 012602
2965 014670 012601
2966 014672 012600
2967 014674 104401 014722
2968 014700 016666 000002 000004
2969 014706 012616
2970 014710 000002
2971 014712 023420
2972 014714 001750
2973 014716 000144
2974 014720 000012
2975 014722 000004
2976 014732
2977
2978
2979 014732
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
```

```

      BMI      7$          ;;BR IF YES
5$:   ASLB     (SP)        ;;MSD?
      BCC     6$          ;;BR IF NO
      MOVB    1(SP),-1(R3) ;;YES--SET THE SIGN
6$:   BIS     #'0,R2      ;;MAKE THE BCD DIGIT ASCII
7$:   BIS     #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
      MOVB    R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
      TST     (R0)+       ;;JUST INCREMENTING
      CMP     R0,#10      ;;CHECK THE TABLE INDEX
      BLT     2$          ;;GO DO THE NEXT DIGIT
      BGT     8$          ;;GO TO EXIT
      MOV     R5,R2       ;;GET THE LSD
      BR      6$          ;;GO CHANGE TO ASCII
8$:   TSTB    (SP)+       ;;WAS THE LSD THE FIRST NON-ZERO?
      BPL     9$          ;;BR IF NO
9$:   MOVB    -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
      CLRB    (R3)        ;;SET THE TERMINATOR
      POP     <R5,R3,R2,R1,R0>
      .IRP    B,<R5,R3,R2,R1,R0>
      MOV     (SP)+,B     ;;POP STACK INTO B
      .ENDM
      MOV     (SP)+,R5    ;;POP STACK INTO R5
      MOV     (SP)+,R3    ;;POP STACK INTO R3
      MOV     (SP)+,R2    ;;POP STACK INTO R2
      MOV     (SP)+,R1    ;;POP STACK INTO R1
      MOV     (SP)+,R0    ;;POP STACK INTO R0
      TYPE    $DBLK       ;;NOW TYPE THE NUMBER
      MOV     2(SP),4(SP) ;;ADJUST THE STACK
      MOV     (SP)+,(SP)
      RTI                ;;RETURN TO USER
$DTBL: 10000.
      1000.
      100.
      10.
$DBLK: .BLKW 4
.$TYPOCT
.$SBTTL BINARY TO OCTAL (ASCII) AND TYPE
STARS
;*****
;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;*OCTAL (ASCII) NUMBER AND TYPE IT.
;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
;*CALL:
;*      MOV     NUM,-(SP)    ;;NUMBER TO BE TYPED
;*      TYPOS    ;;CALL FOR TYPEOUT
;*      .BYTE   N           ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;*      .BYTE   M           ;;M=1 OR 0
;*
;*                               ;;1=TYPE LEADING ZEROS
;*                               ;;0=SUPPRESS LEADING ZEROS
;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
;*$TYPOS OR $TYPOC
;*CALL:
;*      MOV     NUM,-(SP)    ;;NUMBER TO BE TYPED
;*      TYPON   ;;CALL FOR TYPEOUT
```

```
2997  
2998  
2999  
3000  
3001  
3002  
3003 014732 017646 000000  
3004 014736 116637 000001 015155  
3005 014744 112637 015157  
3006 014750 062716 000002  
3007 014754 000406  
3008 014756 112737 000001 015155  
3009 014764 112737 000006 015157  
3010 014772 112737 000005 015154  
3011 015000 010346  
3012 015002 010446  
3013 015004 010546  
3014 015006 113704 015157  
3015 015012 005404  
3016 015014 062704 000006  
3017 015020 110437 015156  
3018 015024 113704 015155  
3019 015030 016605 000012  
3020 015034 005003  
3021 015036 006105 1$:  
3022 015040 000404 BR 3$  
3023 015042 006105 2$:  
3024 015044 006105 ROL R5  
3025 015046 006105 ROL R5  
3026 015050 010503 MOV R5,R3  
3027 015052 006103 3$:  
3028 015054 105337 015156 DECB $OMODE  
3029 015060 100016 BPL 7$  
3030 015062 042703 177770 BIC #177770,R3  
3031 015066 001002 BNE 4$  
3032 015070 005704 TST R4  
3033 015072 001403 BEQ 5$  
3034 015074 005204 4$:  
3035 015076 052703 000060 INC R4  
3036 015102 052703 000040 5$:  
3037 015106 110337 015152 BIS #'0,R3  
3038 015112 104401 015152 BIS #' ,R3  
3039 015116 105337 015154 MOV R3,8$  
3040 015122 003347 7$:  
3041 015124 002402 DECB $OCNT  
3042 015126 005204 BGT 2$  
3043 015130 000744 BLT 6$  
3044 015132 012605 6$:  
3045 015134 012604 INC R4  
3046 015136 012603 BR 2$  
3047 015140 016666 000002 000004 MOV (SP)+,R5  
3048 015146 012616 MOV (SP)+,R4  
3049 015150 000002 MOV (SP)+,R3  
3050 015152 000 RTI  
3051 015153 000 8$:  
3052 015154 000 .BYTE 0  
$OCNT: .BYTE 0
```

3053 015155 000
3054 015156 000000
3055 015160
3056
3057
3058 015160
3059
3060
3061
3062
3063
3064 015160
3065 015160 104401 001233
3066 015164 010046
3067 015166 005000
3068 015170 153700 001114
3069 015174 001004
3070
3071 015176
3072 015176 013746 001116
3073
3074 015202 104402
3075 015204 000426
3076 015206 005300
3077 015210 006300
3078 015212 006300
3079 015214 006300
3080 015216 062700 001324
3081 015222 012037 015232
3082 015226 001404
3083 015230 104401
3084 015232 000000
3085 015234 104401 001233
3086 015240 012037 015250
3087 015244 001404
3088 015246 104401
3089 015250 000000
3090 015252 104401 001233
3091 015256 011000
3092 015260 001004
3093 015262 012600
3094 015264 104401 001233
3095 015270 000207
3096 015272
3097 015272 013046
3098 015274 104402
3099 015276 005710
3100 015300 001770
3101 015302 104401 015310
3102 015306 000771
3103 015310 020040 000
3104 015314 015314
3105 015314
3106
3107
3108 015314

```
$OFILL: .BYTE 0          ;;ZERO FILL SWITCH
$OMODE: .WORD 0          ;;NUMBER OF DIGITS TO TYPE
.$ERRTYP
.$BTTL  ERROR MESSAGE TYPEOUT ROUTINE

STARS
:*****
:*THIS ROUTINE USES THE 'ITEM CONTROL BYTE' ($ITEMB) TO DETERMINE WHICH
:*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE 'ERROR TABLE' ($ERRTB),
:*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

$ERRTYP:
      TYPE      .$CRLF          ;;'CARRIAGE RETURN' & 'LINE FEED'
      MOV      RO,-(SP)        ;;SAVE RO
      CLR      RO              ;;PICKUP THE ITEM INDEX
      BISB     @#$ITEMB,RO
      BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                          ;;TYPE THE PC OF THE ERROR
      TYPOCT   $ERRPC,<ERROR ADDRESS>
      MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                          ;;ERROR ADDRESS
                          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
      BR      6$              ;;GET OUT
1$:   DEC      RO              ;;ADJUST THE INDEX SO THAT IT WILL
      ASL     RO              ;;      WORK FOR THE ERROR TABLE
      ASL     RO
      ASL     RO
      ADD     #$ERRTB,RO      ;;FORM TABLE POINTER
      MOV     (RO)+,2$        ;;PICKUP 'ERROR MESSAGE' POINTER
      BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
      TYPE    'ERROR MESSAGE'
                          ;;TYPE THE 'ERROR MESSAGE'
2$:   .WORD    0              ;;'ERROR MESSAGE' POINTER GOES HERE
      TYPE    .$CRLF          ;;'CARRIAGE RETURN' & 'LINE FEED'
3$:   MOV     (RO)+,4$        ;;PICKUP 'DATA HEADER' POINTER
      BEQ     5$              ;;SKIP TYPEOUT IF 0
      TYPE    'DATA HEADER'
                          ;;TYPE THE 'DATA HEADER'
4$:   .WORD    0              ;;'DATA HEADER' POINTER GOES HERE
      TYPE    .$CRLF          ;;'CARRIAGE RETURN' & 'LINE FEED'
5$:   MOV     (RO),RO         ;;PICKUP 'DATA TABLE' POINTER
      BNE     7$              ;;GO TYPE THE DATA
6$:   MOV     (SP)+,RO        ;;RESTORE RO
      TYPE    .$CRLF          ;;'CARRIAGE RETURN' & 'LINE FEED'
      RTS     PC              ;;RETURN
7$:   TYPOCT   @ (RO)+        ;;TYPE AN OCTAL NUMBER
      MOV     @ (RO)+,-(SP)   ;;SAVE @ (RO)+ FOR TYPEOUT
                          ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
      TST     (RO)           ;;IS THERE ANOTHER NUMBER?
      BEQ     6$              ;;BR IF NO
      TYPE    '  '           ;;TYPE TWO(2) SPACES
      BR      7$              ;;LOOP
8$:   .ASCIZ  ' / /'         ;;TWO(2) SPACES
      .EVEN

.$TRAP
.$BTTL  TRAP DECODER

STARS
```

```
3109 ::*****
3110 :*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
3111 :*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3112 :*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3113 :*GO TO THAT ROUTINE.
3114
3115 015314 010046 $TRAP: MOV R0,-(SP) ::SAVE R0
3116 015316 016600 000002 MOV 2(SP),R0 ::GET TRAP ADDRESS
3117 015322 005740 TST -(R0) ::BACKUP BY 2
3118 015324 111000 MOVB (R0),R0 ::GET RIGHT BYTE OF TRAP
3119 015326 006300 ASL R0 ::POSITION FOR INDEXING
3120 015330 016000 015350 MOV $TRPAD(R0),R0 ::INDEX TO TABLE
3121 015334 000200 RTS R0 ::GO TO ROUTINE
3122
3123
3124 ::THIS IS USE TO HANDLE THE 'GETPRI' MACRO
3125
3126 015336 011646 $TRAP2: MOV (SP),-(SP) ::MOVE THE PC DOWN
3127 015340 016666 000004 000002 MOV 4(SP),2(SP) ::MOVE THE PSW DOWN
3128 015346 000002 RTI ::RESTORE THE PSW
3129
3130 .MACRO SETTRAP A,B,MSG
3131 $$SET A,B,\<TRAP+$TRP>,\$TRP,<MSG>
3132
3133 .NLIST
3134 $TRP=$TRP+1
3135 .LIST
3136 .ENDM SETTRAP
3137 .MACRO $$SET A,B,C,D,COMNT
3138 .IF EQ $TRP-1
3139 .SBTTL TRAP TABLE
3140
3141 :*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3142 :*BY THE 'TRAP' INSTRUCTION.
3143
3144 : ROUTINE
3145 : -----
3146 $TRPAD: .WORD $TRAP2
3147 .ENDC
3148 .IIF NDF GNS,.NLIST
3149 A= C
3150 .IIF NDF GNS,.LIST
3151 B ;;CALL=A TRAP+D(C) COMNT
3152 .ENDM $$SET
3153 .MACRO TRMTRP
3154 $TERM=-$TRPAD
3155 .ENDM TRMTRP
3156 .LIST
3157 015350 SETTRAP TYPE,$TYPE,^/TTY TYPEOUT ROUTINE/
3158 015350 $$SET TYPE,$TYPE,\<TRAP+$TRP>,\$TRP,<TTY TYPEOUT ROUTINE>
3159 .SBTTL TRAP TABLE
3160
3161 :*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
3162 :*BY THE 'TRAP' INSTRUCTION.
3163
3164 : ROUTINE
3165 : -----
```

3165	015350	015336			\$TRPAD: .WORD \$TRAP2
3166					.LIST
3167	015352	014266			\$TYPE ;;CALL=TYPE TRAP+1(104401) TTY TIMEOUT ROUTINE
3168					.LIST
3169	015354				SETTRAP TYPOC,\$TYPOC,^/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
3170	015354				\$\$SET TYPOC,\$TYPOC,\<TRAP+\$TRP>,\\$TRP,<TYPE OCTAL NUMBER (WITH LEADING ZEROS)>
3171					.LIST
3172	015354	014756			\$TYPOC ;;CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
3173					.LIST
3174	015356				SETTRAP TYPOS,\$TYPOS,^/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
3175	015356				\$\$SET TYPOS,\$TYPOS,\<TRAP+\$TRP>,\\$TRP,<TYPE OCTAL NUMBER (NO LEADING ZEROS)>
3176					.LIST
3177	015356	014732			\$TYPOS ;;CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
3178					.LIST
3179	015360				SETTRAP TYPON,\$TYPON,^/TYPE OCTAL NUMBER (AS PER LAST CALL)/
3180	015360				\$\$SET TYPON,\$TYPON,\<TRAP+\$TRP>,\\$TRP,<TYPE OCTAL NUMBER (AS PER LAST CALL)>
3181					.LIST
3182	015360	014772			\$TYPON ;;CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
3183					.LIST
3184	015362				SETTRAP TYPDS,\$TYPDS,^/TYPE DECIMAL NUMBER (WITH SIGN)/
3185	015362				\$\$SET TYPDS,\$TYPDS,\<TRAP+\$TRP>,\\$TRP,<TYPE DECIMAL NUMBER (WITH SIGN)>
3186					.LIST
3187	015362	014506			\$TYPDS ;;CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
3188					.LIST
3189					
3190					
3191	015364	000000			CKSELT: OPEN
3192	015366	055103	041524	042502	STMES: .ASCII 'CZTCBEO TC2-TC11 TEST #2'<15><12>
3193	015374	020060	041524	026462	
3194	015402	041524	030461	052040	
3195	015410	051505	020124	031043	
3196	015416	005015			
3197	015420	042523	020124	047125	.ASCII 'SET UNITO TO REMOTE AND WRITE LOCK.'
3198	015426	052111	020060	047524	
3199	015434	051040	046505	052117	
3200	015442	020105	047101	020104	
3201	015450	051127	052111	020105	
3202	015456	047514	045503	056	
3203	015463	101	046114	047440	.ASCII 'ALL OTHER UNITS OFF.'<15><12>
3204	015470	044124	051105	052440	
3205	015476	044516	051524	047440	
3206	015504	043106	006456	012	
3207	015511	127	052122	020115	.ASCII 'WRTM SWITCH OFF, WALL SWITCH ON.'<15><12>
3208	015516	053523	052111	044103	
3209	015524	047440	043106	020054	
3210	015532	040527	046114	051440	
3211	015540	044527	041524	020110	
3212	015546	047117	006456	012	
3213	015553	116	052117	035105	.ASCIIZ 'NOTE: CACHE TURNED OFF (IF THERE) DURING TEST'<15><12>
3214	015560	041440	041501	042510	
3215	015566	052040	051125	042516	
3216	015574	020104	043117	020106	
3217	015602	044450	020106	044124	
3218	015610	051105	024505	042040	
3219	015616	051125	047111	020107	
3220	015624	042524	052123	005015	

3221	015632	000			
3222	015633	015	051412	052105	ASETSR: .ASCIZ <15><12>'SET SR OPTIONS. NORMAL SR = 0'
3223	015640	051440	020122	050117	
3224	015646	044524	047117	027123	
3225	015654	047040	051117	040515	
3226	015662	020114	051123	036440	
3227	015670	030040	000		
3228	015673	015	044412	053116	AINCRT: .ASCIZ <15><12>'INVALID TEST.'
3229	015700	046101	042111	052040	
3230	015706	051505	027124	000	
3231	015713	007			APGEND: .BYTE 007
3232	015714	025045	000		.ASCIZ '*'
3233	015717	106	052101	046101	TRPM4S: .ASCIZ 'FATAL ERROR TRAP TO LOCATION 4 FROM LOC''
3234	015724	042440	051122	051117	
3235	015732	052040	040522	020120	
3236	015740	047524	046040	041517	
3237	015746	052101	047511	020116	
3238	015754	020064	051106	046517	
3239	015762	046040	041517	000	
3240	015767	050	041077	042101	TRPMES: .ASCIZ ''(?BAD CPU?) ATTEMPTING TO RESTART PROGRAM''
3241	015774	041440	052520	024477	
3242	016002	040440	052124	046505	
3243	016010	052120	047111	020107	
3244	016016	047524	051040	051505	
3245	016024	040524	052122	050040	
3246	016032	047522	051107	046501	
3247	016040	000			
3248	016041	106	052101	046101	TRPM10: .ASCIZ 'FATAL ERROR TRAP TO LOCATION 10 FROM LOC ''
3249	016046	042440	051122	051117	
3250	016054	052040	040522	020120	
3251	016062	047524	046040	041517	
3252	016070	052101	047511	020116	
3253	016076	030061	043040	047522	
3254	016104	020115	047514	020103	
3255	016112	000			
3256	016113	124	051505	051524	RESTART: .ASCIZ 'TESTS ARE OUT OF SEQUENCE - - - RESTARTING....''
3257	016120	040440	042522	047440	
3258	016126	052125	047440	020106	
3259	016134	042523	052521	047105	
3260	016142	042503	026440	026440	
3261	016150	026440	051040	051505	
3262	016156	040524	052122	047111	
3263	016164	027107	027056	000056	
3264	016172	005015	042522	052123	POWMES: .ASCIZ <15> <12> 'RESTARTING AFTER A POWER FAILURE'<15> <12>
3265	016200	051101	044524	043516	
3266	016206	040440	052106	051105	
3267	016214	040440	050040	053517	
3268	016222	051105	043040	044501	
3269	016230	052514	042522	005015	
3270	016236	000			
3271	016237	123	052101	024040	EM1: .ASCIZ 'SAT (STOP ALL TRANSPORTS, COMMAND DID NOT CLEAR READY''
3272	016244	052123	050117	040440	
3273	016252	046114	052040	040522	
3274	016260	051516	047520	052122	
3275	016266	024523	041440	046517	
3276	016274	040515	042116	042040	

3445	017762	051503	000124							
3446					.EVEN					
3447	017766	001116	001176	001200	ET10:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1				
3448	017774	001174	001166	001164						
3449	020002	000000				000000				
3450										
3451										
3452	020004	050125	020123	044504	EM11:	.ASCIZ 'UPS DID NOT CLEAR WHEN LEAVING MAINTANENCE MODE''				
3453	020012	020104	047516	020124						
3454	020020	046103	040505	020122						
3455	020026	044127	047105	046040						
3456	020034	040505	044526	043516						
3457	020042	046440	044501	052116						
3458	020050	047101	047105	042503						
3459	020056	046440	042117	000105						
3460	020064	020040	041520	020040	EH11:	.ASCIZ '' PC SP PS TEST# TCCM TCST''				
3461	020072	020040	020040	050123						
3462	020100	020040	020040	020040						
3463	020106	051520	020040	020040						
3464	020114	052040	051505	021524						
3465	020122	020040	052040	041503						
3466	020130	020115	020040	052040						
3467	020136	051503	000124							
3468					.EVEN					
3469	020142	001116	001176	001200	ET11:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1				
3470	020150	001174	001166	001164						
3471	020156	000000				000000				
3472										
3473										
3474	020160	041524	052123	041040	EM12:	.ASCIZ 'TCST BIT 0 CAN BE SET WHILE IN MAINTANENCE MODF''				
3475	020166	052111	030040	041440						
3476	020174	047101	041040	020105						
3477	020202	042523	020124	044127						
3478	020210	046111	020105	047111						
3479	020216	046440	044501	052116						
3480	020224	047101	047105	042503						
3481	020232	046440	042117	000105						
3482	020240	020040	041520	020040	EH12:	.ASCIZ '' PC SP PS TEST# TCCM TCST''				
3483	020246	020040	020040	050123						
3484	020254	020040	020040	020040						
3485	020262	051520	020040	020040						
3486	020270	052040	051505	021524						
3487	020276	020040	052040	041503						
3488	020304	020115	020040	052040						
3489	020312	051503	000124							
3490					.EVEN					
3491	020316	001116	001176	001200	ET12:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1				
3492	020324	001174	001166	001164						
3493	020332	000000				000000				
3494										
3495										
3496	020334	041524	052123	041040	EM13:	.ASCIZ 'TCST BIT 1 CAN BE SET WHILE IN MAINTANENCE MODE''				
3497	020342	052111	030440	041440						
3498	020350	047101	041040	020105						
3499	020356	042523	020124	044127						
3500	020364	046111	020105	047111						

3557	021026	051503	000124		
3558					
3559	021032	001116	001176	001200	.EVEN
3560	021040	001174	001166	001164	ET15: \$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
3561	021046	000000			000000
3562					
3563					
3564	021050	046103	040505	044522	EM16: .ASCIZ 'CLEARING ILLEGAL OP FAILED TO CLEAR ILO ERROR'
3565	021056	043516	044440	046114	
3566	021064	043505	046101	047440	
3567	021072	020120	040506	046111	
3568	021100	042105	052040	020117	
3569	021106	046103	040505	020122	
3570	021114	046111	020117	051105	
3571	021122	047522	000122		
3572	021126	020040	041520	020040	EH16: .ASCIZ " PC SP PS TEST# TCCM TCST"
3573	021134	020040	020040	050123	
3574	021142	020040	020040	020040	
3575	021150	051520	020040	020040	
3576	021156	052040	051505	021524	
3577	021164	020040	052040	041503	
3578	021172	020115	020040	052040	
3579	021200	051503	000124		
3580					
3581	021204	001116	001176	001200	.EVEN
3582	021212	001174	001166	001164	ET16: \$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
3583	021220	000000			000000
3584					
3585					
3586	021222	046103	040505	044522	EM17: .ASCIZ 'CLEARING ILLEGAL OP FAILED TO CLEAR THE 'ERROR' BIT'
3587	021230	043516	044440	046114	
3588	021236	043505	046101	047440	
3589	021244	020120	040506	046111	
3590	021252	042105	052040	020117	
3591	021260	046103	040505	020122	
3592	021266	044124	020105	042447	
3593	021274	051122	051117	020047	
3594	021302	044502	000124		
3595	021306	020040	041520	020040	EH17: .ASCIZ " PC SP PS TEST# TCCM TCST"
3596	021314	020040	020040	050123	
3597	021322	020040	020040	020040	
3598	021330	051520	020040	020040	
3599	021336	052040	051505	021524	
3600	021344	020040	052040	041503	
3601	021352	020115	020040	052040	
3602	021360	051503	000124		
3603					
3604	021364	001116	001176	001200	.EVEN
3605	021372	001174	001166	001164	ET17: \$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
3606	021400	000000			000000
3607					
3608					
3609	021402	051127	046524	053440	EM20: .ASCIZ 'WRTM WITH WRTM SWITCH OFF DID NOT SET ILO ERROR BIT'
3610	021410	052111	020110	051127	
3611	021416	046524	051440	044527	
3612	021424	041524	020110	043117	

3613	021432	020106	044504	020104							
3614	021440	047516	020124	042523							
3615	021446	020124	046111	020117							
3616	021454	051105	047522	020122							
3617	021462	044502	000124								
3618	021466	020040	041520	020040	EH20:	.ASCIZ	" PC	SP	PS	TEST#	TCCM TCST"
3619	021474	020040	020040	050123							
3620	021502	020040	020040	020040							
3621	021510	051520	020040	020040							
3622	021516	052040	051505	021524							
3623	021524	020040	052040	041503							
3624	021532	020115	020040	052040							
3625	021540	051503	000124								
3626					.EVEN						
3627	021544	001116	001176	001200	ET20:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
3628	021552	001174	001166	001164							
3629	021560	000000				000000					
3630											
3631											
3632	021562	046111	020117	051105	EM21:	.ASCIZ	"ILO ERROR SETTING DID NOT CAUSE THE 'ERROR' BIT TO SET"				
3633	021570	047522	020122	042523							
3634	021576	052124	047111	020107							
3635	021604	044504	020104	047516							
3636	021612	020124	040503	051525							
3637	021620	020105	044124	020105							
3638	021626	042447	051122	051117							
3639	021634	020047	044502	020124							
3640	021642	047524	051440	052105							
3641	021650	000									
3642	021651	040	050040	020103	EH21:	.ASCIZ	" PC	SP	PS	TEST#	TCCM TCST"
3643	021656	020040	020040	051440							
3644	021664	020120	020040	020040							
3645	021672	050040	020123	020040							
3646	021700	020040	042524	052123							
3647	021706	020043	020040	041524							
3648	021714	046503	020040	020040							
3649	021722	041524	052123	000							
3650		021730			.EVEN						
3651	021730	001116	001176	001200	ET21:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1					
3652	021736	001174	001166	001164							
3653	021744	000000				000000					
3654											
3655											
3656	021746	046103	040505	044522	EM22:	.ASCIZ	"CLEARING ERROR BIT ALSO CLEARED ILO ERROR"				
3657	021754	043516	042440	051122							
3658	021762	051117	041040	052111							
3659	021770	040440	051514	020117							
3660	021776	046103	040505	042522							
3661	022004	020104	046111	020117							
3662	022012	051105	047522	000122							
3663	022020	020040	041520	020040	EH22:	.ASCIZ	" PC	SP	PS	TEST#	TCCM TCST"
3664	022026	020040	020040	050123							
3665	022034	020040	020040	020040							
3666	022042	051520	020040	020040							
3667	022050	052040	051505	021524							
3668	022056	020040	052040	041503							

3781	023064	042522	042101	020131	EM30:	.ASCIZ 'READY WAS NOT SET AFTER BLOCK MARK WAS SHIFTED INTO THE WINDOW REGISTER''									
3782	023072	040527	020123	047516											
3783	023100	020124	042523	020124											
3784	023106	043101	042524	020122											
3785	023114	046102	041517	020113											
3786	023122	040515	045522	053440											
3787	023130	051501	051440	044510											
3788	023136	052106	042105	044440											
3789	023144	052116	020117	044124											
3790	023152	020105	044527	042116											
3791	023160	053517	051040	043505											
3792	023166	051511	042524	000122											
3793	023174	020040	041520	020040	EH30:	.ASCIZ '' PC SP PS TEST# TCCM TCST''									
3794	023202	020040	020040	050123											
3795	023210	020040	020040	020040											
3796	023216	051520	020040	020040											
3797	023224	052040	051505	021524											
3798	023232	020040	052040	041503											
3799	023240	020115	020040	052040											
3800	023246	051503	000124												
3801					.EVEN										
3802	023252	001116	001176	001200	ET30:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1									
3803	023260	001174	001166	001164											
3804	023266	000000				000000									
3805															
3806															
3807	023270	047111	047503	051122	EM31:	.ASCIZ ''INCORRECT BLOCK # IN DATA REG AFTER BLOCK MARK WAS DETECTED''									
3808	023276	041505	020124	046102											
3809	023304	041517	020113	020043											
3810	023312	047111	042040	052101											
3811	023320	020101	042522	020107											
3812	023326	043101	042524	020122											
3813	023334	046102	041517	020113											
3814	023342	040515	045522	053440											
3815	023350	051501	042040	052105											
3816	023356	041505	042524	000104											
3817	023364	020040	041520	020040	EH31:	.ASCIZ '' PC SP PS TEST# TCCM TCST TCDT TCDT S/B''									
3818	023372	020040	020040	050123											
3819	023400	020040	020040	020040											
3820	023406	051520	020040	020040											
3821	023414	052040	051505	021524											
3822	023422	020040	052040	041503											
3823	023430	020115	020040	052040											
3824	023436	051503	020124	020040											
3825	023444	052040	042103	020124											
3826	023452	052040	042103	020124											
3827	023460	027523	000102												
3828					.EVEN										
3829	023464	001116	001176	001200	ET31:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,\$REG4,\$REG0									
3830	023472	001174	001166	001164											
3831	023500	001172	001162												
3832	023504	000000				000000									
3833															
3834															
3835	023506	052115	020105	040527	EM32:	.ASCIZ 'MTE WAS NOT SET BY AN ILLEGAL MARK TRACK CODE''									
3836	023514	020123	047516	020124											

4565	032352	001116	001176	001200	ET72:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1,\$REG3
4566	032360	001174	001166	001164		
4567	032366	001170				
4568	032370	000000				000000
4569						
4570						
4571	032372	051523	020124	044504	EM73:	.ASCIZ 'SST DID NOT CAUSE A SELECT ERROR'
4572	032400	020104	047516	020124		
4573	032406	040503	051525	020105		
4574	032414	020101	042523	042514		
4575	032422	052103	042440	051122		
4576	032430	051117	000			
4577	032433	040	050040	020103	EH73:	.ASCIZ '' PC SP PS TEST# TCCM TCST''
4578	032440	020040	020040	051440		
4579	032446	020120	020040	020040		
4580	032454	050040	020123	020040		
4581	032462	020040	042524	052123		
4582	032470	020043	020040	041524		
4583	032476	046503	020040	020040		
4584	032504	041524	052123	000		
4585		032512			.EVEN	
4586	032512	001116	001176	001200	ET73:	\$ERRPC,\$REG6,\$REG7,\$REG5,\$REG2,\$REG1
4587	032520	001174	001166	001164		
4588	032526	000000				000000
4589						
4590	032530	051124	050101	042520	EM74:	.ASCIZ 'TRAPPED TO LOC 4 ATTEMPTING TO ACCESS TCCM'
4591	032536	020104	047524	046040		
4592	032544	041517	032040	040440		
4593	032552	052124	046505	052120		
4594	032560	047111	020107	047524		
4595	032566	040440	041503	051505		
4596	032574	020123	041524	046503		
4597	032602	000				
4598	032603	040	050040	020103	EH74:	.ASCIZ '' PC SP PS TEST#''
4599	032610	020040	020040	051440		
4600	032616	020120	020040	020040		
4601	032624	050040	020123	020040		
4602	032632	020040	042524	052123		
4603	032640	000043				
4604					.EVEN	
4605	032642	001116	001176	001200	ET74:	\$ERRPC,\$REG6,\$REG7,\$REG5
4606	032650	001174				
4607	032652	000000				000000
4608	032654	051124	050101	042520	EM75:	.ASCIZ 'TRAPPED TO LOC 4 ATTEMPTING TO ACCESS TCST'
4609	032662	020104	047524	046040		
4610	032670	041517	032040	040440		
4611	032676	052124	046505	052120		
4612	032704	047111	020107	047524		
4613	032712	040440	041503	051505		
4614	032720	020123	041524	052123		
4615	032726	000				
4616	032727	040	050040	020103	EH75:	.ASCIZ '' PC SP PS TEST#''
4617	032734	020040	020040	051440		
4618	032742	020120	020040	020040		
4619	032750	050040	020123	020040		
4620	032756	020040	042524	052123		

4621	032764	000043							
4622					.EVEN				
4623	032766	001116	001176	001200	ET75:	\$ERRPC,\$REG6,\$REG7,\$REG5			
4624	032774	001174							
4625	032776	000000				000000			
4626	033000	051124	050101	042520	EM76:	.ASCIZ 'TRAPPED TO LOC 4 ATTEMPTING TO ACCESS TCWC'			
4627	033006	020104	047524	046040					
4628	033014	041517	032040	040440					
4629	033022	052124	046505	052120					
4630	033030	047111	020107	047524					
4631	033036	040440	041503	051505					
4632	033044	020123	041524	041527					
4633	033052	000							
4634	033053	040	050040	020103	EH76:	.ASCIZ '' PC SP PS TEST#''			
4635	033060	020040	020040	051440					
4636	033066	020120	020040	020040					
4637	033074	050040	020123	020040					
4638	033102	020040	042524	052123					
4639	033110	000043							
4640					.EVEN				
4641	033112	001116	001176	001200	ET76:	\$ERRPC,\$REG6,\$REG7,\$REG5			
4642	033120	001174							
4643	033122	000000				000000			
4644	033124	051124	050101	042520	EM77:	.ASCIZ 'TRAPPED TO LOC 4 ATTEMPTING TO ACCESS TCBA'			
4645	033132	020104	047524	046040					
4646	033140	041517	032040	040440					
4647	033146	052124	046505	052120					
4648	033154	047111	020107	047524					
4649	033162	040440	041503	051505					
4650	033170	020123	041524	040502					
4651	033176	000							
4652	033177	040	050040	020103	EH77:	.ASCIZ '' PC SP PS TEST#''			
4653	033204	020040	020040	051440					
4654	033212	020120	020040	020040					
4655	033220	050040	020123	020040					
4656	033226	020040	042524	052123					
4657	033234	000043							
4658					.EVEN				
4659	033236	001116	001176	001200	ET77:	\$ERRPC,\$REG6,\$REG7,\$REG5			
4660	033244	001174							
4661	033246	000000				000000			
4662	033250				MTKC10:				
4663	033250					C10 V7,V2,V7,V2,V7,V2			
4664	033250	034	010	074		.BYTE 0!V7,0!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 10.			
4665	033253	010	034	010					
4666	033256				MTKER:				
4667	033256					EMTE			
4668	033256	040	040	040		.BYTE I,I,I,0,0,I			
4669	033261	000	000	040					
4670	033264				MTKEND:				
4671	033264					C22			
4672	033264	000	040	000		.BYTE 0,I,0,0,I,0 ;MTK CODE 22. FWD END ZONE.			
4673	033267	000	040	000					
4674	033272				MTK55:				
4675	033272					C55			
4676	033272	040	000	040		.BYTE I,0,I,I,0,I ;MTK CODE 55. REV END ZONE MARK.			

4677	033275	040	000	040		
4678	033300				MTK5P:	
4679	033300					
4680	033300	000	030	070	C10	V0,V6,V6,V6,V6,V6
4681	033303	030	030	030	.BYTE	0!V0,0!V6,I!V6,0!V6,0!V6,0!V6 ;MTK CODE 10.
4682	033306				MTK7:	
4683	033306					
4684	033306	000	040	000	C25	
4685	033311	040	000	040	.BYTE	0,I,0,I,0,I ;MTK CODE 25. EXTENSION MARK.
4686	033314					
4687	033314	000	040	000	C25	
4688	033317	040	000	040	.BYTE	0,I,0,I,0,I ;MTK CODE 25. EXTENSION MARK.
4689	033322					
4690	033322	000	064	010	C26	V0,V5,V2,V5,V2,V5
4691	033325	064	050	024	.BYTE	0!V0,I!V5,0!V2,I!V5,I!V2,0!V5 ;FWD BLOCK MARK.
4692	033330					
4693	033330	000	064	064	C32	V0,V5,V5,V5,V5,V5
4694	033333	024	064	024	.BYTE	0!V0,I!V5,I!V5,0!V5,I!V5,0!V5 ;REV GUARD.
4695	033336				MTK5:	
4696	033336					
4697	033336	000	030	070	C10	V0,V6,V6,V6,V6,V6
4698	033341	030	030	030	.BYTE	0!V0,0!V6,I!V6,0!V6,0!V6,0!V6 ;MTK CODE 10.
4699	033344				MTK7A:	
4700	033344					
4701	033344	000	000	040	C10	V0,V0,V0,V0,V0,V0
4702	033347	000	000	000	.BYTE	0!V0,0!V0,I!V0,0!V0,0!V0,0!V0 ;MTK CODE 10.
4703	033352				MTK7B:	
4704	033352					
4705	033352	000	024	040	C10	V0,V5,V0,V5,V0,V5
4706	033355	024	000	024	.BYTE	0!V0,0!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 10.
4707	033360				MTKVAR:	
4708	033360					
4709	033360	034	010	074	C10	V7,V2,V7,V2,V7,V2
4710	033363	010	034	010	.BYTE	0!V7,0!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 10.
4711		000176				
4712					.REPT	126.
4713					C70	V0,V5,V0,V5,V0,V5
4714					C70	V7,V2,V7,V2,V7,V2
4715	033366				.ENDR	
4716	033366	040	064	040	C70	V0,V5,V0,V5,V0,V5
4717	033371	024	000	024	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4718	033374					
4719	033374	074	050	074	C70	V7,V2,V7,V2,V7,V2
4720	033377	010	034	010	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4721	033402					
4722	033402	040	064	040	C70	V0,V5,V0,V5,V0,V5
4723	033405	024	000	024	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4724	033410					
4725	033410	074	050	074	C70	V7,V2,V7,V2,V7,V2
4726	033413	010	034	010	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4727	033416					
4728	033416	040	064	040	C70	V0,V5,V0,V5,V0,V5
4729	033421	024	000	024	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4730	033424					
4731	033424	074	050	074	C70	V7,V2,V7,V2,V7,V2
4732	033427	010	034	010	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.

4733	033432				C70	V0,V5,V0,V5,V0,V5	
4734	033432	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4735	033435	024	000	024			
4736	033440				C70	V7,V2,V7,V2,V7,V2	
4737	033440	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4738	033443	010	034	010			
4739	033446				C70	V0,V5,V0,V5,V0,V5	
4740	033446	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4741	033451	024	000	024			
4742	033454				C70	V7,V2,V7,V2,V7,V2	
4743	033454	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4744	033457	010	034	010			
4745	033462				C70	V0,V5,V0,V5,V0,V5	
4746	033462	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4747	033465	024	000	024			
4748	033470				C70	V7,V2,V7,V2,V7,V2	
4749	033470	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4750	033473	010	034	010			
4751	033476				C70	V0,V5,V0,V5,V0,V5	
4752	033476	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4753	033501	024	000	024			
4754	033504				C70	V7,V2,V7,V2,V7,V2	
4755	033504	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4756	033507	010	034	010			
4757	033512				C70	V0,V5,V0,V5,V0,V5	
4758	033512	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4759	033515	024	000	024			
4760	033520				C70	V7,V2,V7,V2,V7,V2	
4761	033520	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4762	033523	010	034	010			
4763	033526				C70	V0,V5,V0,V5,V0,V5	
4764	033526	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4765	033531	024	000	024			
4766	033534				C70	V7,V2,V7,V2,V7,V2	
4767	033534	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4768	033537	010	034	010			
4769	033542				C70	V0,V5,V0,V5,V0,V5	
4770	033542	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4771	033545	024	000	024			
4772	033550				C70	V7,V2,V7,V2,V7,V2	
4773	033550	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4774	033553	010	034	010			
4775	033556				C70	V0,V5,V0,V5,V0,V5	
4776	033556	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4777	033561	024	000	024			
4778	033564				C70	V7,V2,V7,V2,V7,V2	
4779	033564	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4780	033567	010	034	010			
4781	033572				C70	V0,V5,V0,V5,V0,V5	
4782	033572	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4783	033575	024	000	024			
4784	033600				C70	V7,V2,V7,V2,V7,V2	
4785	033600	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4786	033603	010	034	010			
4787	033606				C70	V0,V5,V0,V5,V0,V5	
4788	033606	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.

4789	033611	024	000	024			
4790	033614				C70	V7,V2,V7,V2,V7,V2	
4791	033614	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4792	033617	010	034	010			
4793	033622				C70	V0,V5,V0,V5,V0,V5	
4794	033622	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4795	033625	024	000	024			
4796	033630				C70	V7,V2,V7,V2,V7,V2	
4797	033630	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4798	033633	010	034	010			
4799	033636				C70	V0,V5,V0,V5,V0,V5	
4800	033636	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4801	033641	024	000	024			
4802	033644				C70	V7,V2,V7,V2,V7,V2	
4803	033644	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4804	033647	010	034	010			
4805	033652				C70	V0,V5,V0,V5,V0,V5	
4806	033652	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4807	033655	024	000	024			
4808	033660				C70	V7,V2,V7,V2,V7,V2	
4809	033660	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4810	033663	010	034	010			
4811	033666				C70	V0,V5,V0,V5,V0,V5	
4812	033666	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4813	033671	024	000	024			
4814	033674				C70	V7,V2,V7,V2,V7,V2	
4815	033674	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4816	033677	010	034	010			
4817	033702				C70	V0,V5,V0,V5,V0,V5	
4818	033702	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4819	033705	024	000	024			
4820	033710				C70	V7,V2,V7,V2,V7,V2	
4821	033710	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4822	033713	010	034	010			
4823	033716				C70	V0,V5,V0,V5,V0,V5	
4824	033716	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4825	033721	024	000	024			
4826	033724				C70	V7,V2,V7,V2,V7,V2	
4827	033724	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4828	033727	010	034	010			
4829	033732				C70	V0,V5,V0,V5,V0,V5	
4830	033732	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4831	033735	024	000	024			
4832	033740				C70	V7,V2,V7,V2,V7,V2	
4833	033740	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4834	033743	010	034	010			
4835	033746				C70	V0,V5,V0,V5,V0,V5	
4836	033746	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4837	033751	024	000	024			
4838	033754				C70	V7,V2,V7,V2,V7,V2	
4839	033754	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4840	033757	010	034	010			
4841	033762				C70	V0,V5,V0,V5,V0,V5	
4842	033762	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4843	033765	024	000	024			
4844	033770				C70	V7,V2,V7,V2,V7,V2	

4845	033770	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4846	033773	010	034	010			
4847	033776				C70	V0,V5,V0,V5,V0,V5	
4848	033776	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4849	034001	024	000	024			
4850	034004				C70	V7,V2,V7,V2,V7,V2	
4851	034004	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4852	034007	010	034	010			
4853	034012				C70	V0,V5,V0,V5,V0,V5	
4854	034012	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4855	034015	024	000	024			
4856	034020				C70	V7,V2,V7,V2,V7,V2	
4857	034020	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4858	034023	010	034	010			
4859	034026				C70	V0,V5,V0,V5,V0,V5	
4860	034026	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4861	034031	024	000	024			
4862	034034				C70	V7,V2,V7,V2,V7,V2	
4863	034034	074	050	074	.BYTF	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4864	034037	010	034	010			
4865	034042				C70	V0,V5,V0,V5,V0,V5	
4866	034042	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4867	034045	024	000	024			
4868	034050				C70	V7,V2,V7,V2,V7,V2	
4869	034050	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4870	034053	010	034	010			
4871	034056				C70	V0,V5,V0,V5,V0,V5	
4872	034056	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4873	034061	024	000	024			
4874	034064				C70	V7,V2,V7,V2,V7,V2	
4875	034064	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4876	034067	010	034	010			
4877	034072				C70	V0,V5,V0,V5,V0,V5	
4878	034072	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4879	034075	024	000	024			
4880	034100				C70	V7,V2,V7,V2,V7,V2	
4881	034100	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4882	034103	010	034	010			
4883	034106				C70	V0,V5,V0,V5,V0,V5	
4884	034106	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4885	034111	024	000	024			
4886	034114				C70	V7,V2,V7,V2,V7,V2	
4887	034114	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4888	034117	010	034	010			
4889	034122				C70	V0,V5,V0,V5,V0,V5	
4890	034122	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4891	034125	024	000	024			
4892	034130				C70	V7,V2,V7,V2,V7,V2	
4893	034130	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4894	034133	010	034	010			
4895	034136				C70	V0,V5,V0,V5,V0,V5	
4896	034136	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4897	034141	024	000	024			
4898	034144				C70	V7,V2,V7,V2,V7,V2	
4899	034144	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4900	034147	010	034	010			

4901	034152				C70	V0,V5,V0,V5	
4902	034152	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4903	034155	024	000	024			
4904	034160				C70	V7,V2,V7,V2,V7,V2	
4905	034160	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4906	034163	010	034	010			
4907	034166				C70	V0,V5,V0,V5,V0,V5	
4908	034166	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4909	034171	024	000	024			
4910	034174				C70	V7,V2,V7,V2,V7,V2	
4911	034174	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4912	034177	010	034	010			
4913	034202				C70	V0,V5,V0,V5,V0,V5	
4914	034202	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4915	034205	024	000	024			
4916	034210				C70	V7,V2,V7,V2,V7,V2	
4917	034210	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4918	034213	010	034	010			
4919	034216				C70	V0,V5,V0,V5,V0,V5	
4920	034216	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4921	034221	024	000	024			
4922	034224				C70	V7,V2,V7,V2,V7,V2	
4923	034224	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4924	034227	010	034	010			
4925	034232				C70	V0,V5,V0,V5,V0,V5	
4926	034232	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4927	034235	024	000	024			
4928	034240				C70	V7,V2,V7,V2,V7,V2	
4929	034240	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4930	034243	010	034	010			
4931	034246				C70	V0,V5,V0,V5,V0,V5	
4932	034246	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4933	034251	024	000	024			
4934	034254				C70	V7,V2,V7,V2,V7,V2	
4935	034254	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4936	034257	010	034	010			
4937	034262				C70	V0,V5,V0,V5,V0,V5	
4938	034262	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4939	034265	024	000	024			
4940	034270				C70	V7,V2,V7,V2,V7,V2	
4941	034270	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4942	034273	010	034	010			
4943	034276				C70	V0,V5,V0,V5,V0,V5	
4944	034276	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4945	034301	024	000	024			
4946	034304				C70	V7,V2,V7,V2,V7,V2	
4947	034304	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4948	034307	010	034	010			
4949	034312				C70	V0,V5,V0,V5,V0,V5	
4950	034312	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
4951	034315	024	000	024			
4952	034320				C70	V7,V2,V7,V2,V7,V2	
4953	034320	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
4954	034323	010	034	010			
4955	034326				C70	V0,V5,V0,V5,V0,V5	
4956	034326	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.

4957	034331	024	000	024		
4958	034334				C70	V7,V2,V7,V2,V7,V2
4959	034334	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4960	034337	010	034	010		
4961	034342				C70	V0,V5,V0,V5,V0,V5
4962	034342	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4963	034345	024	000	024		
4964	034350				C70	V7,V2,V7,V2,V7,V2
4965	034350	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4966	034353	010	034	010		
4967	034356				C70	V0,V5,V0,V5,V0,V5
4968	034356	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4969	034361	024	000	024		
4970	034364				C70	V7,V2,V7,V2,V7,V2
4971	034364	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4972	034367	010	034	010		
4973	034372				C70	V0,V5,V0,V5,V0,V5
4974	034372	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4975	034375	024	000	024		
4976	034400				C70	V7,V2,V7,V2,V7,V2
4977	034400	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4978	034403	010	034	010		
4979	034406				C70	V0,V5,V0,V5,V0,V5
4980	034406	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4981	034411	024	000	024		
4982	034414				C70	V7,V2,V7,V2,V7,V2
4983	034414	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4984	034417	010	034	010		
4985	034422				C70	V0,V5,V0,V5,V0,V5
4986	034422	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4987	034425	024	000	024		
4988	034430				C70	V7,V2,V7,V2,V7,V2
4989	034430	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4990	034433	010	034	010		
4991	034436				C70	V0,V5,V0,V5,V0,V5
4992	034436	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4993	034441	024	000	024		
4994	034444				C70	V7,V2,V7,V2,V7,V2
4995	034444	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
4996	034447	010	034	010		
4997	034452				C70	V0,V5,V0,V5,V0,V5
4998	034452	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
4999	034455	024	000	024		
5000	034460				C70	V7,V2,V7,V2,V7,V2
5001	034460	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
5002	034463	010	034	010		
5003	034466				C70	V0,V5,V0,V5,V0,V5
5004	034466	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
5005	034471	024	000	024		
5006	034474				C70	V7,V2,V7,V2,V7,V2
5007	034474	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2 ;MTK CODE 70. DATA MARK.
5008	034477	010	034	010		
5009	034502				C70	V0,V5,V0,V5,V0,V5
5010	034502	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5 ;MTK CODE 70. DATA MARK.
5011	034505	024	000	024		
5012	034510				C70	V7,V2,V7,V2,V7,V2

5013	034510	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5014	034513	010	034	010			
5015	034516				C70	V0,V5,V0,V5,V0,V5	
5016	034516	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5017	034521	024	000	024			
5018	034524				C70	V7,V2,V7,V2,V7,V2	
5019	034524	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5020	034527	010	034	010			
5021	034532				C70	V0,V5,V0,V5,V0,V5	
5022	034532	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0 V5	;MTK CODE 70. DATA MARK.
5023	034535	024	000	024			
5024	034540				C70	V7,V2,V7,V2,V7,V2	
5025	034540	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5026	034543	010	034	010			
5027	034546				C70	V0,V5,V0,V5,V0,V5	
5028	034546	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5029	034551	024	000	024			
5030	034554				C70	V7,V2,V7,V2,V7,V2	
5031	034554	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5032	034557	010	034	010			
5033	034562				C70	V0,V5,V0,V5,V0,V5	
5034	034562	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5035	034565	024	000	024			
5036	034570				C70	V7,V2,V7,V2,V7,V2	
5037	034570	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5038	034573	010	034	010			
5039	034576				C70	V0,V5,V0,V5,V0,V5	
5040	034576	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5041	034601	024	000	024			
5042	034604				C70	V7,V2,V7,V2,V7,V2	
5043	034604	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5044	034607	010	034	010			
5045	034612				C70	V0,V5,V0,V5,V0,V5	
5046	034612	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5047	034615	024	000	024			
5048	034620				C70	V7,V2,V7,V2,V7,V2	
5049	034620	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5050	034623	010	034	010			
5051	034626				C70	V0,V5,V0,V5,V0,V5	
5052	034626	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5053	034631	024	000	024			
5054	034634				C70	V7,V2,V7,V2,V7,V2	
5055	034634	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5056	034637	010	034	010			
5057	034642				C70	V0,V5,V0,V5,V0,V5	
5058	034642	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5059	034645	024	000	024			
5060	034650				C70	V7,V2,V7,V2,V7,V2	
5061	034650	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5062	034653	010	034	010			
5063	034656				C70	V0,V5,V0,V5,V0,V5	
5064	034656	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5065	034661	024	000	024			
5066	034664				C70	V7,V2,V7,V2,V7,V2	
5067	034664	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5068	034667	010	034	010			

5069	034672				C70	V0,V5,V0,V5,V0,V5	
5070	034672	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5071	034675	024	000	024			
5072	034700				C70	V7,V2,V7,V2,V7,V2	
5073	034700	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5074	034703	010	034	010			
5075	034706				C70	V0,V5,V0,V5,V0,V5	
5076	034706	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5077	034711	024	000	024			
5078	034714				C70	V7,V2,V7,V2,V7,V2	
5079	034714	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5080	034717	010	034	010			
5081	034722				C70	V0,V5,V0,V5,V0,V5	
5082	034722	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5083	034725	024	000	024			
5084	034730				C70	V7,V2,V7,V2,V7,V2	
5085	034730	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5086	034733	010	034	010			
5087	034736				C70	V0,V5,V0,V5,V0,V5	
5088	034736	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5089	034741	024	000	024			
5090	034744				C70	V7,V2,V7,V2,V7,V2	
5091	034744	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5092	034747	010	034	010			
5093	034752				C70	V0,V5,V0,V5,V0,V5	
5094	034752	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5095	034755	024	000	024			
5096	034760				C70	V7,V2,V7,V2,V7,V2	
5097	034760	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5098	034763	010	034	010			
5099	034766				C70	V0,V5,V0,V5,V0,V5	
5100	034766	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5101	034771	024	000	024			
5102	034774				C70	V7,V2,V7,V2,V7,V2	
5103	034774	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5104	034777	010	034	010			
5105	035002				C70	V0,V5,V0,V5,V0,V5	
5106	035002	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5107	035005	024	000	024			
5108	035010				C70	V7,V2,V7,V2,V7,V2	
5109	035010	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5110	035013	010	034	010			
5111	035016				C70	V0,V5,V0,V5,V0,V5	
5112	035016	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5113	035021	024	000	024			
5114	035024				C70	V7,V2,V7,V2,V7,V2	
5115	035024	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5116	035027	010	034	010			
5117	035032				C70	V0,V5,V0,V5,V0,V5	
5118	035032	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5119	035035	024	000	024			
5120	035040				C70	V7,V2,V7,V2,V7,V2	
5121	035040	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5122	035043	010	034	010			
5123	035046				C70	V0,V5,V0,V5,V0,V5	
5124	035046	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.

5125	035051	024	000	024			
5126	035054				C70	V7,V2,V7,V2,V7,V2	
5127	035054	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0.V2	;MTK CODE 70. DATA MARK.
5128	035057	010	034	010			
5129	035062				C70	V0,V5,V0,V5,V0,V5	
5130	035062	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0.V5	;MTK CODE 70. DATA MARK.
5131	035065	024	000	024			
5132	035070				C70	V7,V2,V7,V2,V7,V2	
5133	035070	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5134	035073	010	034	010			
5135	035076				C70	V0,V5,V0,V5,V0,V5	
5136	035076	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5137	035101	024	000	024			
5138	035104				C70	V7,V2,V7,V2,V7,V2	
5139	035104	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5140	035107	010	034	010			
5141	035112				C70	V0,V5,V0,V5,V0,V5	
5142	035112	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5143	035115	024	000	024			
5144	035120				C70	V7,V2,V7,V2,V7,V2	
5145	035120	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5146	035123	010	034	010			
5147	035126				C70	V0,V5,V0,V5,V0,V5	
5148	035126	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5149	035131	024	000	024			
5150	035134				C70	V7,V2,V7,V2,V7,V2	
5151	035134	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5152	035137	010	034	010			
5153	035142				C70	V0,V5,V0,V5,V0,V5	
5154	035142	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5155	035145	024	000	024			
5156	035150				C70	V7,V2,V7,V2,V7,V2	
5157	035150	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5158	035153	010	034	010			
5159	035156				C70	V0,V5,V0,V5,V0,V5	
5160	035156	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5161	035161	024	000	024			
5162	035164				C70	V7,V2,V7,V2,V7,V2	
5163	035164	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5164	035167	010	034	010			
5165	035172				C70	V0,V5,V0,V5,V0,V5	
5166	035172	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5167	035175	024	000	024			
5168	035200				C70	V7,V2,V7,V2,V7,V2	
5169	035200	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5170	035203	010	034	010			
5171	035206				C70	V0,V5,V0,V5,V0,V5	
5172	035206	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5173	035211	024	000	024			
5174	035214				C70	V7,V2,V7,V2,V7,V2	
5175	035214	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5176	035217	010	034	010			
5177	035222				C70	V0,V5,V0,V5,V0,V5	
5178	035222	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5179	035225	024	000	024			
5180	035230				C70	V7,V2,V7,V2,V7,V2	

5181	035230	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5182	035233	010	034	010			
5183	035236				C70	V0,V5,V0,V5,V0,V5	
5184	035236	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5185	035241	024	000	024			
5186	035244				C70	V7,V2,V7,V2,V7,V2	
5187	035244	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5188	035247	010	034	010			
5189	035252				C70	V0,V5,V0,V5,V0,V5	
5190	035252	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5191	035255	024	000	024			
5192	035260				C70	V7,V2,V7,V2,V7,V2	
5193	035260	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5194	035263	010	034	010			
5195	035266				C70	V0,V5,V0,V5,V0,V5	
5196	035266	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5197	035271	024	000	024			
5198	035274				C70	V7,V2,V7,V2,V7,V2	
5199	035274	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5200	035277	010	034	010			
5201	035302				C70	V0,V5,V0,V5,V0,V5	
5202	035302	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5203	035305	024	000	024			
5204	035310				C70	V7,V2,V7,V2,V7,V2	
5205	035310	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5206	035313	010	034	010			
5207	035316				C70	V0,V5,V0,V5,V0,V5	
5208	035316	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5209	035321	024	000	024			
5210	035324				C70	V7,V2,V7,V2,V7,V2	
5211	035324	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5212	035327	010	034	010			
5213	035332				C70	V0,V5,V0,V5,V0,V5	
5214	035332	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5215	035335	024	000	024			
5216	035340				C70	V7,V2,V7,V2,V7,V2	
5217	035340	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5218	035343	010	034	010			
5219	035346				C70	V0,V5,V0,V5,V0,V5	
5220	035346	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5221	035351	024	000	024			
5222	035354				C70	V7,V2,V7,V2,V7,V2	
5223	035354	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5224	035357	010	034	010			
5225	035362				C70	V0,V5,V0,V5,V0,V5	
5226	035362	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5227	035365	024	000	024			
5228	035370				C70	V7,V2,V7,V2,V7,V2	
5229	035370	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5230	035373	010	034	010			
5231	035376				C70	V0,V5,V0,V5,V0,V5	
5232	035376	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5233	035401	024	000	024			
5234	035404				C70	V7,V2,V7,V2,V7,V2	
5235	035404	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5236	035407	010	034	010			

5237	035412				C70	V0,V5,V0,V5,V0,V5	
5238	035412	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5239	035415	024	000	024			
5240	035420				C70	V7,V2,V7,V2,V7,V2	
5241	035420	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5242	035423	010	034	010			
5243	035426				C70	V0,V5,V0,V5,V0,V5	
5244	035426	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5245	035431	024	000	024			
5246	035434				C70	V7,V2,V7,V2,V7,V2	
5247	035434	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5248	035437	010	034	010			
5249	035442				C70	V0,V5,V0,V5,V0,V5	
5250	035442	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5251	035445	024	000	024			
5252	035450				C70	V7,V2,V7,V2,V7,V2	
5253	035450	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5254	035453	010	034	010			
5255	035456				C70	V0,V5,V0,V5,V0,V5	
5256	035456	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5257	035461	024	000	024			
5258	035464				C70	V7,V2,V7,V2,V7,V2	
5259	035464	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5260	035467	010	034	010			
5261	035472				C70	V0,V5,V0,V5,V0,V5	
5262	035472	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5263	035475	024	000	024			
5264	035500				C70	V7,V2,V7,V2,V7,V2	
5265	035500	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5266	035503	010	034	010			
5267	035506				C70	V0,V5,V0,V5,V0,V5	
5268	035506	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5269	035511	024	000	024			
5270	035514				C70	V7,V2,V7,V2,V7,V2	
5271	035514	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5272	035517	010	034	010			
5273	035522				C70	V0,V5,V0,V5,V0,V5	
5274	035522	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5275	035525	024	000	024			
5276	035530				C70	V7,V2,V7,V2,V7,V2	
5277	035530	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5278	035533	010	034	010			
5279	035536				C70	V0,V5,V0,V5,V0,V5	
5280	035536	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5281	035541	024	000	024			
5282	035544				C70	V7,V2,V7,V2,V7,V2	
5283	035544	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5284	035547	010	034	010			
5285	035552				C70	V0,V5,V0,V5,V0,V5	
5286	035552	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5287	035555	024	000	024			
5288	035560				C70	V7,V2,V7,V2,V7,V2	
5289	035560	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5290	035563	010	034	010			
5291	035566				C70	V0,V5,V0,V5,V0,V5	
5292	035566	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.

5293	035571	024	000	024			
5294	035574				C70	V7,V2,V7,V2,V7,V2	
5295	035574	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5296	035577	010	034	010			
5297	035602				C70	V0,V5,V0,V5,V0,V5	
5298	035602	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5299	035605	024	000	024			
5300	035610				C70	V7,V2,V7,V2,V7,V2	
5301	035610	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5302	035613	010	034	010			
5303	035616				C70	V0,V5,V0,V5,V0,V5	
5304	035616	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5305	035621	024	000	024			
5306	035624				C70	V7,V2,V7,V2,V7,V2	
5307	035624	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5308	035627	010	034	010			
5309	035632				C70	V0,V5,V0,V5,V0,V5	
5310	035632	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5311	035635	024	000	024			
5312	035640				C70	V7,V2,V7,V2,V7,V2	
5313	035640	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5314	035643	010	034	010			
5315	035646				C70	V0,V5,V0,V5,V0,V5	
5316	035646	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5317	035651	024	000	024			
5318	035654				C70	V7,V2,V7,V2,V7,V2	
5319	035654	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5320	035657	010	034	010			
5321	035662				C70	V0,V5,V0,V5,V0,V5	
5322	035662	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5323	035665	024	000	024			
5324	035670				C70	V7,V2,V7,V2,V7,V2	
5325	035670	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5326	035673	010	034	010			
5327	035676				C70	V0,V5,V0,V5,V0,V5	
5328	035676	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5329	035701	024	000	024			
5330	035704				C70	V7,V2,V7,V2,V7,V2	
5331	035704	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5332	035707	010	034	010			
5333	035712				C70	V0,V5,V0,V5,V0,V5	
5334	035712	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5335	035715	024	000	024			
5336	035720				C70	V7,V2,V7,V2,V7,V2	
5337	035720	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5338	035723	010	034	010			
5339	035726				C70	V0,V5,V0,V5,V0,V5	
5340	035726	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5341	035731	024	000	024			
5342	035734				C70	V7,V2,V7,V2,V7,V2	
5343	035734	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5344	035737	010	034	010			
5345	035742				C70	V0,V5,V0,V5,V0,V5	
5346	035742	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5347	035745	024	000	024			
5348	035750				C70	V7,V2,V7,V2,V7,V2	

5349	035750	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5350	035753	010	034	010			
5351	035756				C70	V0,V5,V0,V5,V0,V5	
5352	035756	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5353	035761	024	000	024			
5354	035764				C70	V7,V2,V7,V2,V7,V2	
5355	035764	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5356	035767	010	034	010			
5357	035772				C70	V0,V5,V0,V5,V0,V5	
5358	035772	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5359	035775	024	000	024			
5360	036000				C70	V7,V2,V7,V2,V7,V2	
5361	036000	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5362	036003	010	034	010			
5363	036006				C70	V0,V5,V0,V5,V0,V5	
5364	036006	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5365	036011	024	000	024			
5366	036014				C70	V7,V2,V7,V2,V7,V2	
5367	036014	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5368	036017	010	034	010			
5369	036022				C70	V0,V5,V0,V5,V0,V5	
5370	036022	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5371	036025	024	000	024			
5372	036030				C70	V7,V2,V7,V2,V7,V2	
5373	036030	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5374	036033	010	034	010			
5375	036036				C70	V0,V5,V0,V5,V0,V5	
5376	036036	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5377	036041	024	000	024			
5378	036044				C70	V7,V2,V7,V2,V7,V2	
5379	036044	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5380	036047	010	034	010			
5381	036052				C70	V0,V5,V0,V5,V0,V5	
5382	036052	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5383	036055	024	000	024			
5384	036060				C70	V7,V2,V7,V2,V7,V2	
5385	036060	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5386	036063	010	034	010			
5387	036066				C70	V0,V5,V0,V5,V0,V5	
5388	036066	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5389	036071	024	000	024			
5390	036074				C70	V7,V2,V7,V2,V7,V2	
5391	036074	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5392	036077	010	034	010			
5393	036102				C70	V0,V5,V0,V5,V0,V5	
5394	036102	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5395	036105	024	000	024			
5396	036110				C70	V7,V2,V7,V2,V7,V2	
5397	036110	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5398	036113	010	034	010			
5399	036116				C70	V0,V5,V0,V5,V0,V5	
5400	036116	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5401	036121	024	000	024			
5402	036124				C70	V7,V2,V7,V2,V7,V2	
5403	036124	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5404	036127	010	034	010			

5405	036132				C70	V0,V5,V0,V5,V0,V5	
5406	036132	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5407	036135	024	000	024			
5408	036140				C70	V7,V2,V7,V2,V7,V2	
5409	036140	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5410	036143	010	034	010			
5411	036146				C70	V0,V5,V0,V5,V0,V5	
5412	036146	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5413	036151	024	000	024			
5414	036154				C70	V7,V2,V7,V2,V7,V2	
5415	036154	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5416	036157	010	034	010			
5417	036162				C70	V0,V5,V0,V5,V0,V5	
5418	036162	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5419	036165	024	000	024			
5420	036170				C70	V7,V2,V7,V2,V7,V2	
5421	036170	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5422	036173	010	034	010			
5423	036176				C70	V0,V5,V0,V5,V0,V5	
5424	036176	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5425	036201	024	000	024			
5426	036204				C70	V7,V2,V7,V2,V7,V2	
5427	036204	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5428	036207	010	034	010			
5429	036212				C70	V0,V5,V0,V5,V0,V5	
5430	036212	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5431	036215	024	000	024			
5432	036220				C70	V7,V2,V7,V2,V7,V2	
5433	036220	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5434	036223	010	034	010			
5435	036226				C70	V0,V5,V0,V5,V0,V5	
5436	036226	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0 V0,0!V5	;MTK CODE 70. DATA MARK.
5437	036231	024	000	024			
5438	036234				C70	V7,V2,V7,V2,V7,V2	
5439	036234	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5440	036237	010	034	010			
5441	036242				C70	V0,V5,V0,V5,V0,V5	
5442	036242	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5443	036245	024	000	024			
5444	036250				C70	V7,V2,V7,V2,V7,V2	
5445	036250	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5446	036253	010	034	010			
5447	036256				C70	V0,V5,V0,V5,V0,V5	
5448	036256	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5449	036261	024	000	024			
5450	036264				C70	V7,V2,V7,V2,V7,V2	
5451	036264	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5452	036267	010	034	010			
5453	036272				C70	V0,V5,V0,V5,V0,V5	
5454	036272	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5455	036275	024	000	024			
5456	036300				C70	V7,V2,V7,V2,V7,V2	
5457	036300	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5458	036303	010	034	010			
5459	036306				C70	V0,V5,V0,V5,V0,V5	
5460	036306	040	064	040	.BYTE	I.V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.

5461	036311	024	000	024			
5462	036314				C70	V7,V2,V7,V2,V7,V2	
5463	036314	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5464	036317	010	034	010			
5465	036322				C70	V0,V5,V0,V5,V0,V5	
5466	036322	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,0!V0,0!V5	;MTK CODE 70. DATA MARK.
5467	036325	024	000	024			
5468	036330				C70	V7,V2,V7,V2,V7,V2	
5469	036330	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,0!V7,0!V2	;MTK CODE 70. DATA MARK.
5470	036333	010	034	010			
5471	036336				C73	V0,V5,V0,V5,V0,V5	
5472	036336	040	064	040	.BYTE	I!V0,I!V5,I!V0,0!V5,I!V0,I!V5	;MTK CODE 73. DATA MARK.
5473	036341	024	040	064			
5474	036344				C73	V7,V2,V7,V2,V7,V2	
5475	036344	074	050	074	.BYTE	I!V7,I!V2,I!V7,0!V2,I!V7,I!V2	;MTK CODE 73. DATA MARK.
5476	036347	010	074	050			
5477	036352				FCKSM:		
5478	036352				C73	V0,V0,V0,V0,V0,V0	
5479	036352	040	040	040	.BYTE	I!V0,I!V0,I!V0,0!V0,I!V0,I!V0	;MTK CODE 73. DATA MARK.
5480	036355	000	040	040			
5481	036360				C73	V0,V0,V0,V0,V0,V0	
5482	036360	040	040	040	.BYTE	I!V0,I!V0,I!V0,0!V0,I!V0,I!V0	;MTK CODE 73. DATA MARK.
5483	036363	000	040	040			
5484	036366				C51	V0,V0,V0,V0,V0,V0	
5485	036366	040	000	040	.BYTE	I!V0,0!V0,I!V0,0!V0,0!V0,I!V0	;MTK CODE 51. FWD GUARD.
5486	036371	000	000	040			
5487	036374				C45	V0,V0,V0,V0,V0,V0	
5488	036374	040	000	000	.BYTE	I!V0,0!V0,0!V0,I!V0,0!V0,I!V0	;MTK CODE 45. REV BLOCK MARK.
5489	036377	040	000	040			
5490	036402				C25		
5491	036402	000	040	000	.BYTE	0,I,0,I,0,I	;MTK CODE 25. EXTENSION MARK.
5492	036405	040	000	040			
5493	036410				CEND		
5494	036410	377			.BYTE	-1	
5495	036411				GCKSM:		
5496	036411				C73	V7,V7,V0,V0,V0,V0	
5497	036411	074	074	040	.BYTE	I!V7,I!V7,I!V0,0!V0,I!V0,I!V0	;MTK CODE 73. DATA MARK.
5498	036414	000	040	040			
5499	036417				BCKSM:		
5500	036417				C73	V0,V0,V0,V0,V0,V0	
5501	036417	040	040	040	.BYTE	I!V0,I!V0,I!V0,0!V0,I!V0,I!V0	;MTK CODE 73. DATA MARK.
5502	036422	000	040	040			
5503		036426			.EVEN		
5504	036426	000000			OPEN		
5505		036430			RBUF=.		
5506		040430			. =RBUF+1024.		
5507		000001			.END		

A	= 100000	192#			
AINCRT	015673	3228#			
APGEND	015713	3231#			
ASETSR	015633	3222#			
A0001	002734	936	945#		
A0002	003004	950	959#		
A0003	003054	964	973#		
A0004	003124	978	987#		
A0005	003210	1005	1007#		
A0006	003252	1022	1024#		
A0007	003324	1036	1042#		
A0010	003376	1059#	1065		
A0011	003472	1075	1083#		
A0012	003552	1104#	1112	1115	
A0013	003642	1125	1128#		
A0014	003766	1152	1154#		
A0015	004042	1168	1170#		
A0016	004114	1187	1190#		
A0017	004230	1218	1221#		
A0020	004342	1248	1251#		
A0021	004442	1267	1274#		
A0022	004534	1296	1299#		
A0023	004634	1323	1326#		
A0024	004752	1353	1359#		
A0025	005040	1381	1384#		
A0026	005134	1409	1412#		
A0027	005234	1437	1440#		
A0030	005330	1463	1466#		
A0031	005422	1489	1492#		
A0032	005664	1545	1548#		
A0033	006142	1604	1608#		
A0034	006336	1649	1652#		
A0035	006542	1697	1700#		
A0036	006746	1745	1748#		
A0037	007150	1792	1795#		
A0040	007316	1828	1831#		
A0041	007636	1900	1902#		
A0042	010114	1963	1966#		
A0043	010276	2010	2013#		
A0044	010524	2064	2067#		
A0045	010700	2104	2107#		
A0055	011430	2246	2249#		
B	= 040000	193#			
BCKSM	036417	2573	5499#		
BELL	= 000007	186#			
BIT0	= 000001	147#	225	1150	1151
BIT00	= 000001	137#	147		
BIT01	= 000002	136#	146		
BIT02	= 000004	135#	145		
BIT03	= 000010	134#	144		
BIT04	= 000020	133#	143		
BIT05	= 000040	132#	142		
BIT06	= 000100	131#	141		
BIT07	= 000200	130#	140		
BIT08	= 000400	129#	139	2656	
BIT09	= 001000	128#	138	2664	2746

BIT1	=	000002	146#	218	220	222	224	1166	1167					
BIT10	=	002000	127#	212	213	214	215	1836	1856	1867	2731			
BIT11	=	004000	126#	205	2611	2671								
BIT12	=	010000	125#	204	1186	1195	1217	1226	1252					
BIT13	=	020000	124#	194	203	1128	1133	1137	1149	1165	1380	2518	2738	
BIT14	=	040000	123#	193	1544	1603	2642							
BIT15	=	100000	122#	181	192	1251								
BIT2	=	000004	145#	219	220	223	224							
BIT3	=	000010	144#	221	222	223	224							
BIT4	=	000020	143#											
BIT5	=	000040	142#											
BIT6	=	000100	141#	216	1039	1057	1079	1101	1270					
BIT7	=	000200	140#	1124	1129	1134								
BIT8	=	000400	139#	209	211	213	215							
BIT9	=	001000	138#	210	211	214	215	1962	1971					
BMOVE		012660	2509#	2511										
BMOVE		012646	1371	1399	1426	2408	2412	2416	2469	2476	2505#	2552	2556	2572
BPTVEC	=	000014	154#											
BTCTR		013036	2523*	2530*	2532	2546#								
B0010		003410	1054	1063#										
B0011		003474	1082	1084#										
B0012		003564	1098	1108#										
B0013		003664	1130	1133#										
B0016		004126	1191	1194#										
B0017		004242	1222	1225#										
B0020		004362	1250	1253	1255#									
B0021		004456	1275	1277#										
B0022		004552	1298	1301	1303#									
B0023		004644	1325	1327	1333#									
B0025		005050	1383	1385	1387#									
B0026		005144	1411	1413	1415#									
B0027		005244	1439	1441	1443#									
B0031		005434	1493	1496#										
B0032		005670	1543	1550#										
B0033		006162	1609	1613#										
B0034		006362	1653	1658#										
B0035		006566	1701	1706#										
B0036		006772	1749	1754#										
B0037		007162	1794	1799#										
B0040		007356	1837	1841#										
B0041		007660	1905	1908#										
B0042		010126	1967	1970#										
B0043		010350	2023	2026#										
B0044		010576	2077	2080#										
B0045		010756	2113	2120#										
B0045A		010770	2121	2124#										
B0055		011502	2259	2262#										
C	=	020000	194#											
CCR		001322	475#	917*										
CDCNT		013040	2517*	2519	2547#									
CDCTR		013042	2519*	2539*	2548#									
CDTCK		013240	2266	2583	2586	2591#								
CDTCKA		013272	2596	2598#										
CHGE1		002574	915#											
CHLT		012302	2422#											
CKDAT		013166	1662	1710	1757	1795	1914	2040	2080	2133	2577#			

CZTCBEO TC2-TC11 TEST #2
CZTCBE.P11 10-APR-80 15:17

MACY11 30A(1052) 10-APR-80 15:18 F 9 PAGE 106
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0109

D0040	007502	1862	1866#	
D0041	007700	1903	1914#	
D0041A	007726	1919	1922#	
D0041B	007740	1923	1926#	
D0042	010156	1955	1978#	
D0043	010406	2031	2036#	
D0044	010616	2073	2087#	
D0045	011022	2129	2133#	
D0055	011532	2248	2261	2269#
EH1	016325	494	3281#	
EH10	017710	536	3438#	
EH11	020064	542	3460#	
EH12	020240	548	3482#	
EH13	020414	554	3504#	
EH14	020607	560	3529#	
EH15	020754	566	3550#	
EH16	021126	572	3572#	
EH17	021306	578	3595#	
EH2	016504	500	3304#	
EH20	021466	584	3618#	
EH21	021651	590	3642#	
EH22	022020	596	3663#	
EH23	022150	602	3682#	
EH24	022323	608	3704#	
EH25	022504	614	3727#	
EH26	022637	620	3747#	
EH27	022770	626	3767#	
EH3	016645	506	3325#	
EH30	023174	632	3793#	
EH31	023364	638	3817#	
EH32	023564	644	3843#	
EH33	023732	650	3864#	
EH34	024103	656	662	3886#
EH35	024233	3905#		
EH36	024433	666	3929#	
EH37	024536	672	3944#	
EH4	017051	512	3351#	
EH40	024660	678	3962#	
EH41	025020	684	3982#	
EH42	025152	690	4001#	
EH43	025312	696	4021#	
EH44	025500	702	4045#	
EH45	025660	708	4068#	
EH46	026021	714	4089#	
EH47	026174	720	4111#	
EH5	017177	518	3370#	
EH50	026322	726	4130#	
EH51	026475	732	4153#	
EH52	026657	738	4177#	
EH53	027037	744	4201#	
EH54	027206	750	4224#	
EH55	027353	756	4246#	
EH56	027540	762	4271#	
EH57	027725	768	4295#	
EH6	017371	524	3395#	
EH60	030112	774	4320#	

EH61	030303	780	4345#	
EH62	030462	786	4368#	
EH63	030626	792	4389#	
EH64	030776	798	4411#	
EH65	031173	804	4437#	
EH66	031367	810	4462#	
EH67	031545	816	4485#	
EH7	017542	530	3417#	
EH70	031717	822	4507#	
EH71	032116	828	4533#	
EH72	032264	834	4555#	
EH73	032433	840	4577#	
EH74	032603	844	4598#	
EH75	032727	849	4616#	
EH76	033053	854	4634#	
EH77	033177	859	4652#	
EMTVEC=	000030	157#	879*	880*
EMTX =	000000	226#		
EM1	016237	493	3271#	
EM10	017636	535	3431#	
EM11	020004	541	3452#	
EM12	020160	547	3474#	
EM13	020334	553	3496#	
EM14	020510	559	3518#	
EM15	020704	565	3543#	
EM16	021050	571	3564#	
EM17	021222	577	3586#	
EM2	016422	499	3295#	
EM20	021402	583	3609#	
EM21	021562	589	3632#	
EM22	021746	595	3656#	
EM23	022114	601	3677#	
EM24	022244	607	3696#	
EM25	022420	613	3718#	
EM26	022600	619	3741#	
EM27	022726	625	3761#	
EM3	016600	505	3318#	
EM30	023064	631	3781#	
EM31	023270	637	3807#	
EM32	023506	643	3835#	
EM33	023660	649	3857#	
EM34	024026	655	661	3878#
EM35	024200	3900#		
EM36	024356	665	3921#	
EM37	024530	671	3943#	
EM4	016742	511	3339#	
EM40	024632	677	3958#	
EM41	024754	683	3976#	
EM42	025114	689	3996#	
EM43	025246	695	4015#	
EM44	025436	701	4039#	
EM45	025624	707	4063#	
EM46	026004	713	4086#	
EM47	026116	719	4103#	
EM5	017146	517	3365#	
EM50	026270	725	4125#	

G0042	010174	1979	1984#															
G0043	010434	2012	2025	2029	2035	2039	2044#											
G0045	011060	2106	2123	2132	2140	2142	2144#											
HT	= 000011	60#	2859	2900														
H0032	005774	1547	1549	1553	1558	1562	1568	1570	1572#									
H0034	006434	1670	1675#															
H0035	006640	1718	1723#															
H0036	007042	1765	1770#															
H0042	010212	1985	1988#															
H0043	010442	2019	2047#															
I	= 000040	190#	4664	4668	4672	4676	4680	4684	4687	4690	4693	4697	4701	4705				
		4709	4716	4719	4722	4725	4728	4731	4734	4737	4740	4743	4746	4749				
		4752	4755	4758	4761	4764	4767	4770	4773	4776	4779	4782	4785	4788				
		4791	4794	4797	4800	4803	4806	4809	4812	4815	4818	4821	4824	4827				
		4830	4833	4836	4839	4842	4845	4848	4851	4854	4857	4860	4863	4866				
		4869	4872	4875	4878	4881	4884	4887	4890	4893	4896	4899	4902	4905				
		4908	4911	4914	4917	4920	4923	4926	4929	4932	4935	4938	4941	4944				
		4947	4950	4953	4956	4959	4962	4965	4968	4971	4974	4977	4980	4983				
		4986	4989	4992	4995	4998	5001	5004	5007	5010	5013	5016	5019	5022				
		5025	5028	5031	5034	5037	5040	5043	5046	5049	5052	5055	5058	5061				
		5064	5067	5070	5073	5076	5079	5082	5085	5088	5091	5094	5097	5100				
		5103	5106	5109	5112	5115	5118	5121	5124	5127	5130	5133	5136	5139				
		5142	5145	5148	5151	5154	5157	5160	5163	5166	5169	5172	5175	5178				
		5181	5184	5187	5190	5193	5196	5199	5202	5205	5208	5211	5214	5217				
		5220	5223	5226	5229	5232	5235	5238	5241	5244	5247	5250	5253	5256				
		5259	5262	5265	5268	5271	5274	5277	5280	5283	5286	5289	5292	5295				
		5298	5301	5304	5307	5310	5313	5316	5319	5322	5325	5328	5331	5334				
		5337	5340	5343	5346	5349	5352	5355	5358	5361	5364	5367	5370	5373				
		5376	5379	5382	5385	5388	5391	5394	5397	5400	5403	5406	5409	5412				
		5415	5418	5421	5424	5427	5430	5433	5436	5439	5442	5445	5448	5451				
		5454	5457	5460	5463	5466	5469	5472	5475	5479	5482	5485	5488	5491				
		5497	5501															
IE	= 000100	216#	1457	1483	1537	1598	1643	1691	1739	1786	1889	1942	1957	1986				
		2114																
INBIN	012510	2475#																
IOTVEC	= 000020	155#	877*	878*														
I0031	005546	1515	1520#															
I0032	006002	1535	1575#															
I0041	007772	1931	1936#															
I0045	011066	2117	2147#															
K0032	006020	1576	1581#															
LBBIND	013104	2238	2561#															
LBDAT1	013052	2002	2056	2096	2551#													
LBINDA	013126	2565#	2568															
LF	= 000012	61#	2894	2900														
LMTCA	012734	2522#	2544															
LMTCAA	012700	2516#	2550															
LMTCB	012740	2523#	2543															
LMTCC	012746	2524#	2533	2535	2538													
LMTCD	013026	2540	2542#															
LMTCE	013016	2531	2539#															
LMTCOD	012674	1319	1349	1377	1405	1433	1459	1485	1539	1600	1645	1693	1741	1788				
		1824	1833	1846	1853	1891	1959	2006	2060	2100	2242	2515#						
LMTCOE	013044	2018	2072	2116	2254	2549#												
MAINT	- 020000	203#	1184	1215	1245	1277	1299	1317	1347	1375	1403	1431	1457	1483				
		1537	1598	1643	1691	1739	1786	1822	1844	1889	1957	2004	2058	2098				

ADITAG	227#	448													
CEND	292#	5493													
COMMEN	1#	162#													
C10	274#	4663	4679	4696	4700	4704	4708								
C22	289#	4671													
C25	265#	4683	4686	5490											
C26	268#	4689													
C32	271#	4692													
C45	286#	5487													
C51	283#	5484													
C55	262#	4675													
C70	277#	4715	4718	4721	4724	4727	4730	4733	4736	4739	4742	4745	4748	4751	4754
	4757	4760	4763	4766	4769	4772	4775	4778	4781	4784	4787	4790	4793	4796	4799
	4802	4805	4808	4811	4814	4817	4820	4823	4826	4829	4832	4835	4838	4841	4844
	4847	4850	4853	4856	4859	4862	4865	4868	4871	4874	4877	4880	4883	4886	4889
	4892	4895	4898	4901	4904	4907	4910	4913	4916	4919	4922	4925	4928	4931	4934
	4937	4940	4943	4946	4949	4952	4955	4958	4961	4964	4967	4970	4973	4976	4979
	4982	4985	4988	4991	4994	4997	5000	5003	5006	5009	5012	5015	5018	5021	5024
	5027	5030	5033	5036	5039	5042	5045	5048	5051	5054	5057	5060	5063	5066	5069
	5072	5075	5078	5081	5084	5087	5090	5093	5096	5099	5102	5105	5108	5111	5114
	5117	5120	5123	5126	5129	5132	5135	5138	5141	5144	5147	5150	5153	5156	5159
	5162	5165	5168	5171	5174	5177	5180	5183	5186	5189	5192	5195	5198	5201	5204
	5207	5210	5213	5216	5219	5222	5225	5228	5231	5234	5237	5240	5243	5246	5249
	5252	5255	5258	5261	5264	5267	5270	5273	5276	5279	5282	5285	5288	5291	5294
	5297	5300	5303	5306	5309	5312	5315	5318	5321	5324	5327	5330	5333	5336	5339
	5342	5345	5348	5351	5354	5357	5360	5363	5366	5369	5372	5375	5378	5381	5384
	5387	5390	5393	5396	5399	5402	5405	5408	5411	5414	5417	5420	5423	5426	5429
	5432	5435	5438	5441	5444	5447	5450	5453	5456	5459	5462	5465	5468		
C73	280#	5471	5474	5478	5481	5496	5500								
EMTDEF	309#														
EMTE	295#	4667													
ENDCOM	1#	162#													
ERROR	56#	945	959	973	987	1006	1023	1041	1063	1081	1103	1114	1126	1131	1136
	1153	1169	1188	1192	1197	1201	1219	1223	1228	1232	1249	1254	1272	1279	1297
	1302	1324	1328	1358	1382	1386	1410	1414	1438	1442	1464	1466	1490	1494	1499
	1505	1510	1516	1546	1548	1552	1557	1561	1567	1571	1577	1606	1611	1617	1623
	1650	1656	1660	1671	1698	1704	1708	1719	1746	1752	1756	1766	1793	1804	1829
	1839	1859	1864	1870	1896	1906	1910	1912	1920	1924	1928	1932	1964	1968	1973
	1980	2011	2024	2028	2034	2038	2065	2078	2105	2122	2131	2139	2143	2158	2171
	2182	2193	2204	2215	2229	2247	2260	2349	2600						
ESCAPE	1#	162#													
GETPRI	1#	162#													
GETSWR	1#	162#													
MTCOD	298#	1318	1348	1376	1404	1432	1458	1484	1538	1599	1644	1692	1740	1787	1823
	1832	1845	1852	1890	1958	2005	2059	2099	2241						
MTCOE	303#	2017	2071	2115	2253										
MULT	1#	162#													
NEWTST	1#	162#													
POP	1#	162#	2802	2807	2958										
PUSH	1#	162#	2767	2777	2784	2914									
REPORT	1#	162#													
SAVE	2689#	2719													
SCOMAC	316#	2637													
SCOPE	57#	935	949	963	977	993	1014	1031	1049	1070	1093	1120	1145	1161	1180
	1211	1241	1262	1291	1312	1342	1367	1395	1422	1450	1475	1529	1589	1635	1683
	1731	1778	1815	1881	1950	1998	2052	2092	2152	2164	2176	2187	2198	2209	2223

CZTCBE0 TC2-TC11 TEST #2
CZTCBE.P11 10-APR-80 15:17

I 10
MACY11 30A(1052) 10-APR-80 15:18 PAGE 123
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0125

.STRAP	1#	11#	3105
.STYPB	1#		
.STYPD	1#	11#	2900
.STYPE	1#	12#	2828
.STYPO	1#	11#	2976
.S4OCA	1#		
.1170	1#		

. ABS. 040430 000

ERRORS DETECTED: 0

CZTCBE.BIN,CZTCBE.LST/CRF/SOL/NL:TOC=CZTCBE.SML,CZTCBE.P11
RUN-TIME: 42 60 6 SECONDS
RUN-TIME RATIO: 234/109=2.1
CORE USED: 34K (67 PAGES)