IDENTIFICATION

PRODUCT CODE:	MAINDEC-Ø8-DHTDA-A-D REPLACES: MAINDEC-8E-D3AB-D
PRODUCT NAME:	TD8E DECTAPE DIAGNOSTIC
DATE CREATED:	NOVEMBER 1, 1972
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	E. STEINBERGER/B. HANSEN

1. ABSTRACT

TD8E DECTAPE DIAGNOSTIC IS A PROGRAM WHICH HAS BEEN WRITTEN TO CHECKOUT AND TEST TD8E DECTAPE CONTROLS WITH TU56 DECTAPE TRANSPORTS. THE PROGRAM TESTS THE BASIC FUNCTIONS OF THE CONTROL (IOT SKIPS, DATA TRANSFERS, ETC) AS WELL AS CHECKING THE ABILITY TO READ AND WRITE ON DECTAPE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-8E
TD8E DECTAPE CONTROL
TU56 DECTAPE TRANSPORT (AT LEAST ONE)
ALL NECESSARY CABLES AND MODULES

2.2 STORAGE

THE PROGRAM OCCUPPIES MEMORY FROM LOCATION 20 TO LOCATION 7177 AND USES LOCATIONS 7200 TO 7577 AS DATA BUFFER AREA.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

3.1 METHOD

THE PROGRAM IS LOADED USING THE STANDARD BINARY LOADER TECHNIQUE.

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

THE FOLLOWING IS A TABLE OF CONTROL SWITCH SETTINGS AND THEIR ACTION UPON THE PROGRAM:

SR	STATE	ACTION
0	1	LOOP ON CURRENT SUBTEST
	0	DON'T LOOP
1	1	LOOP ON CURRENT TEST
	0	DON'T LOOP
2	1	LOOP ON CONTROL TESTS
	0	DON'T LOOP
3	1	DON'T PRINT ERRORS
	0	PRINT ERRORS
4	1	DON'T HALT ON ERRORS
	0	HALT ON ERROR
5	1	
	0	
6	1	
	0	
7	1	
	0	
8	1	
	0	
9	1	
	0	
10	1	
	0	
11	1	SINGLE UNIT TRANSPORT
	0	DUAL UNIT TRANSPORT

4.2 STARTING ADDRESSES

0200	OPERATOR INTERVENTION TESTS
0201	CONTROL AND DATA TRANSFER TESTS
2100	SEARCH AND FIND ALL BLOCK NUMBERS
2200	DISPLAY BLOCK NUMBERS IN AC
2237	ROUTINE TO ROCK DECTAPE 0 (TIME DEPENDENT ON SWITCH REGISTER)
2400	READ AND CHECK THE MARK TRACK FROM ENDZONE TO ENDZONE
7200	IOT MODIFICATION PROGRAM

4,3 PROGRAM AND/OR OPERATOR ACTION

4,3,1 TO TEST "SELECT ERROR" AND "WRITE LOCK OUT"

4,3,1,1 DUAL TRANSPORTS

A) SET SWITCH REGISTER TO 0200

B) ON THE TRANSPORTS, SET ONE TRANSPORT TO UNIT 0, ON-LINE, WRITE LOCK; SET THE OTHER TRANSPORT TO UNIT 1, OFF-LINE.

C) DEPRESS "LOAD ADDRESS", THEN "CLEAR", THEN "CONTINUE", THE PROGRAM SHOULD TYPE "OK"

D) REVERSE THE ROLES OF THE TWO TRANSPORTS AND REPEAT STEP C,

E) SET BOTH TRANSPORTS TO UNIT 1, ON-LINE, DEPRESS "LOAD ADDRESS", THEN "CLEAR", THEN "CONTINUE", THE PROGRAM SHOULD INDICATE NO UNIT 0 SELECTED

F) PROCEED TO 4,3,2

4,3,1,2 SINGLE TRANSPORT

A) SET SWITCH REGISTER TO 0200

B) ON THE TRANSPORT, SET TO UNIT 0, ON-LINE, WRITE LOCK

C) DEPRESS "LOAD", THEN "CLEAR", THEN "CONTINUE", THE PROGRAM SHOULD TYPE "OK"

D) PROCEED TO 4,3,2

4,3,2 TO TEST CONTROL AND ABILITY TO PERFORM DATA TRANSFERS

A) SET SWITCH REGISTER TO 0201, DEPRESS "LOAD ADDRESS"

B) SET SWITCH REGISTER PER 4,1, SET SR11 IF ONLY ONE TRANSPORT EXISTS OR ONLY ONE TRANSPORT IS TO BE TESTED,

C) MOUNT A STANDARD PDP-8 DECTAPE (2702 BLOCKS, 201 WORDS PER BLOCK) ON EACH TRANSPORT TO BE TESTED WITH THE TAPES WRAPPED AT LEAST 2 TURNS ON EACH TAKE UP REEL, RESPECTIVELY,

D) SET A TRANSPORT TO UNIT 0, ON-LINE, WRITE ENABLE; SET THE OTHER TRANSPORT (IF IT EXISTS OR IS TO BE TESTED) TO UNIT 1, ON -LINE, WRITE ENABLE,

E) DEPRESS "CLEAR", THEN "CONTINUE", THE PROGRAM WILL PERFORM THE BASIC CONTROL TESTS ON THE TD8E, AND, IF SR2 IS A 0, PROCEED TO MOVE TAPE AND PERFORM DATA TRANSFERS TO AND FROM TAPE, CHECKING THE RESULTS,

4,3,3 TO MODIFY THE TD8E IOT SET TO HANDLE A CONTROL FOR UNITS OTHER THAN 0 AND 1,

A) SET SWITCH REGISTER TO 7200, DEPRESS "LOAD ADDRESS"

B) SET SWITCH REGISTER BITS 6, 7 AND 8 TO DEVICE SELECTOR BITS 6, 7, AND 8 OF THE CONTROL TO BE TESTED (4, 5, 6 OR 7)

C) DEPRESS "CLEAR", THEN "CONTINUE", THE PROGRAM WILL MODIFY ALL TD8E IOT'S TO HANDLE THE SELECTED CONTROL,

D) PERFORM ALL TESTS INDICATED IN 4,3,1 AND 4,3,2 ABOVE FOR THE SELECTED CONTROL SUBSTITUTING UNIT 2, 4 OR 6 FOR UNIT 0 AND UNIT 3, 5 OR 7 FOR UNIT 1 ABOVE,

E) CAUTION- THE CODE TO CHANGE THE IOT'S IS IN THE DATA BUFFER AREA FOR THE DATA TRANSFER TESTS AND WILL BE DESTROYED WHEN THAT PORTION OF THE PROGRAM IS RUN, AN OVERLAY TAPE IS PROVIDED TO ALLOW THIS CODE TO BE READ BACK INTO MEMORY FOR RE-EXECUTION,

5, OPERATING PROCEDURE

5,1 OPERATIONAL SWITCH SETTINGS

SEE 4,1

5,2 SUBROUTINE ABSTRACTS

NONE

5,3 PROGRAM AND/OR OPERATOR ACTION

SEE 4,3

5,3,1 IF PROBLEMS ARE SUSPECTED IN THE CONTROL WHEN READING THE TIMING TRACK OFF OF DECTAPE INCLUDING SINGLE LINE FLAG AND QUAD LINE FLAG, A ROCK TAPE ROUTINE HAS BEEN PROVIDED AT LOCATION 2237 TO ALLOW SCOPING OF SINGLE LINE FLAG, QUAD LINE FLAG, UP-TO-SPEED, ETC. (2237)

A) SET SWITCH REGISTER TO 2237, DEPRESS "LOAD ADDRESS"

B) SET SWITCH REGISTER TO 0070, DEPRESS "CLEAR" THEN "CONTINUE", THE DECTAPE ON UNIT 0 SHOULD START ROCKING,

C) MODIFY SWITCH REGISTER SETTING TO INCREASE OR DECREASE "ROCK" PERIOD,

D) CAUTION-IF THE NUMBER IN THE SWITCH REGISTER IS TOO SMALL, THE DECTAPE TRANSPORT WILL NOT GET UP TO SPEED BEFORE IT TURNS AROUND,

5,3,2 A ROUTINE HAS BEEN PROVIDED AT LOCATION 2100 TO ALLOW A DECTAPE TO BE RUN FROM ENDZONE TO ENDZONE COMPARING ALL BLOCK NUMBERS, TO RUN THIS ROUTINE, START THE COMPUTER AT LOCATION 2100, THE SR HAS NO AFFECT UPON THE ROUTINE, TO RUN UNIT 1 CHANGE THE CONTENTS OF UNIT (LOCATION 2234) TO 4000, A HALT WILL OCCUR AT LOCATION 2150 IF AN ERROR OCCURS, THE CONTENTS OF THE AC EQUALS THE BLOCK THAT WAS BEING SEARCHED FOR, PRESS "CONT" AND THE PROGRAM WILL HALT AT LOCATION 2153 WITH THE AC EQUAL TO THE BLOCK THAT WAS FOUND, PRESS "CONT" AGAIN, THE PROGRAM WILL RECYCLE UNTIL ANOTHER ERROR IS FOUND. (2100)

5,3,3 A ROUTINE HAS BEEN PROVIDED AT LOCATION 2200 TO ALLOW A DECTAPE TO BE RUN FROM ENDZONE TO ENDZONE WITH THE BLOCK NUMBERS DISPLAYED IN THE AC, TO RUN THIS ROUTINE, START THE COMPUTER AT LOCATION 2200, THE SR HAS NO AFFECT UPON THE ROUTINE, TO RUN UNIT 1, CHANGE THE CONTENTS OF UNIT (LOCATION 2234) TO 4000, NO ERRORS ARE DETECTED, (2200)

5,3,4 A ROUTINE HAS BEEN PROVIDED AT LOCATION 2400 TO ALLOW A DECTAPE TO BE RUN FROM ENDZONE TO ENDZONE WITH THE MARK TRACK FORMAT BEING CHECKED, AN ERROR HALT WILL OCCUR IF AN ERROR IS DETECTED, TO RUN THIS ROUTINE (ONLY ON UNIT 0), START THE COMPUTER AT LOCATION 2400, THE SR HAS NO AFFECT UPON THE ROUTINE. (2400)

6. ERRORS

6.1 ERROR HALTS AND DESCRIPTION

MOST ERROR HALTS ARE PRECEDED BY AN ERROR TYPEOUT (UNLESS SR3 IS A 1). IF NO ERROR TYPEOUT OCCURS, CONSULT THE LISTING FOR THE CAUSE OF THE ERROR.

6.2 ERROR RECOVERY

MOST ERRORS (EXCEPT DATA ERRORS) CAN BE /SCOPED BY SETTING SR0 TO 1 AND DEPRESSING "CONTINUE"

DATA ERRORS CANNOT BE /SCOPED, BUT DATA TRANSFERS CAN BE CONTINUED BY DEPRESSING "CONTINUE".

6.3 IF TAPE RUNS OFF THE END

NORMALLY, TAPE WILL NOT RUN OFF THE END OF THE REEL UNLESS THE PROGRAM IS IN A /SCOPE LOOP OR A SELECT ERROR OCCURS WHEN A TAPE IS MOVING (THE OPERATOR SETTING BOTH TAPE UNITS TO THE SAME NUMBER DURING THE DATA TRANSFER TESTS).

IF TAPE RUNS OFF THE END AND THE PROGRAM HANGS AROUND LOCATION 0621, CHECK THE ABILITY TO READ THE TIMING TRACK INTO THE TDBE CONTROL AND THE CIRCUITS RELATING TO THE TIMING PULSE GENERATOR.

IF TAPE RUNS OFF THE END AND THE PROGRAM HANGS AROUND LOCATION 1466, CHECK THE ABILITY TO READ THE MARK TRACK INTO THE TDBE CONTROL AND THE CIRCUITS RELATING TO THE MARK TRACK REGISTER.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

NONE IF PARAGRAPH 4.3 IS PROPERLY FOLLOWED.

7.2 OPERATING RESTRICTIONS

NONE IF PARAGRAPH 4.3 IS PROPERLY FOLLOWED.

8. MISCELLANEOUS

8.1 EXECUTION TIME — ≈ 20 MIN / Transport SA=201

THE EXECUTION TIME OF THE CONTROL TESTS IS NORMALLY LESS THAN 1 MINUTE, DEPENDING UPON THE POSITION OF TAPE ON UNIT 0.

THE EXECUTION TIME OF THE DATA TESTS DEPENDS ON WHETHER ONE OR TWO TRANSPORTS IS BEING EXERCISED. PASS "N" COMPLETE WILL BE PRINTED ON THE TELEPRINTER AFTER ALL DATA PATTERNS HAVE BEEN EXERCISED ONCE. (NORMALLY LESS THAN 1 HOUR PER PASS).

≈ 20 MIN / Transport.

9. PROGRAM DESCRIPTION

9.1 DATA REGISTER TEST (SA=0201)

IN THIS TEST THE DATA REGISTER IS CHECKED FOR ITS ABILITY TO BE LOADED AND READ. FIRST THE COMPLEMENT OF THE DATA TO BE CHECKED IS LOADED INTO THE DATA REGISTER, THEN THE DATA ITSELF IS LOADED INTO THE REGISTER. THIS IS DONE TO CHECK THAT ALL BITS CAN BE LOADED TO A 1 FROM A 0 AND TO A 0 FROM A 1. THE DATA IS THEN READ INTO THE AC AND CHECKED FOR ERRORS. AN INCREMENT PATTERN IS USED.

9.2 COMMAND REGISTER TEST (SA=0236)

IN THIS TEST THE COMMAND REGISTER IS CHECKED FOR ITS ABILITY TO BE LOADED AND READ. DATA IS LOADED INTO THE COMMAND REGISTER THEN READ IN THE AC AND CHECKED FOR ERRORS. A 400 INCREMENT PATTERN IS USED, THE STOP/GO BIT IS MASKED OUT.

9.3 INITIALIZE TEST (SA=0305)

THIS TESTS CHECKS THAT "CAF" CLEARS THE COMMAND REGISTER. THE C(R) IS LOADED WITH 6400, THEN "CAF" IS ISSUED. THE C(R) IS THEN READ AND CHECKED TO CONTAIN 0.

9.4 CHECK SDLC, SDLD, SDRC, AND SDRD AND AC CLEAR (SA=0400)

THIS TEST CHECKS THE AC CLEAR FUNCTION OF THE SDLC, SDLD, SDRC AND SDRD INSTRUCTION. THIS IS DONE BY SETTING THE AC TO 7777, THEN ISSUING THE APPROPRIATE IOT (ONE AT A TIME) AND CHECK TO SEE IF THE AC DID OR DID NOT CLEAR (SDLD DOES NOT CLEAR THE AC, THE OTHERS DO).

9,5 CHECK SINGLE LINE SKIP INSTRUCTION AND LOGIC TEST (SA=0600)

THIS TEST CHECKS THE SINGLE LINE FLAG LOGIC AND SKIP INSTRUCTION, IN PARTICULAR IT TESTS; SINGLE LINE FLAG CLEAR AFTER A "CAF"; SINGLE LINE FLAG SETS; SDSS DOES NOT CLEAR SINGLE LINE FLAG; CAF CLEARS SINGLE LINE FLAG; SDLD CLEARS SINGLE LINE FLAG; SDRC CLEARS SINGLE LINE FLAG; SDRD CLEARS SINGLE LINE FLAG; SDST, SDSQ, AND SDLC DOES NOT CLEAR SINGLE LINE FLAG.

9,6 CHECK QUAD LINE SKIP INSTRUCTION AND LOGIC TEST (SA=1024)

THIS TEST CHECKS THE QUAD LINE FLAG LOGIC AND SKIP INSTRUCTION; IN PARTICULAR IT TESTS; QUAD LINE FLAG CLEAR AFTER A "CAF"; QUAD LINE FLAG SETS AT PROPER TIME; SDSQ DOES NOT CLEAR QUAD LINE FLAG; CAF CLEARS QUAD LINE FLAG; SDLD CLEARS QUAD LINE FLAG; SDRC CLEARS QUAD LINE FLAG; SDRD CLEARS QUAD LINE FLAG; SDST, SDSQ, AND SDLC DOES NOT CLEAR QUAD LINE FLAG; ALL QUAD LINE FLAG COUNTER FLIP/FLOPS GET CLEARED (BY SDLD);

9,7 CHECK TIMING ERROR SKIP INSTRUCTION AND LOGIC TEST (SA=1315)

THIS TEST CHECKS THE TIMING ERROR LOGIC AND SKIP INSTRUCTION; IN PARTICULAR IT TEST; TIMING ERROR CLEAR AFTER A "CAF"; TIMING ERROR SETS IN READ MODE (SDSQ SKIPS); SDST DOES NOT CLEAR TIMING ERROR; CAF CLEARS TIMING ERROR; TIMING ERROR STATUS BIT CAN BE READ INTO AC BY SDRC; SDLC CLEARS TIMING ERROR; TIMING ERROR SETS IN WRITE MODE (PERFORMED AT REVERSE ENDZONE AT BEGINNING OF TAPE); TIMING ERROR STATUS CLEARS "WRITE"; SDRC SDRD SDLD ISSUED AT THE WRONG TIME SETS TIMING ERROR.

9,8 CHECK UP TO SPEED CIRCUITRY TEST (SA=1600)

THIS TEST CHECKS THE UP-TO-SPEED CIRCUITRY TO FUNCTION PROPERLY WHEN CERTAIN COMMANDS ARE GIVEN TO THE DECTAPE CONTROL. THE CHECK IS PERFORMED VIA THE FEATURE OF THE UP-TO-SPEED CIRCUITRY CLEARING THE MARK TRACK REGISTER WHEN THE UP-TO-SPEED DELAY STARTS TIMING OUT. THE COMMANDS ISSUED ARE; STOP TO GO; GO TO STOP; REVERSE TO FORWARD; FORWARD TO REVERSE; UNIT 0 TO UNIT 1; UNIT 1 TO UNIT 0 (ONLY IF UNIT 1 EXISTS - SR11 SET TO 1)

9,9 ROUTINE TO SEARCH AND FIND ALL BLOCK NUMBERS (SA=2100)

THIS ROUTINE RUNS TAPE FROM ENDZONE TO ENDZONE COMPARING ALL BLOCK NUMBERS,

9,10 DISPLAY BLOCK NUMBER ROUTINE (SA=2200)

THIS ROUTINE RUNS TAPE FROM ENDZONE TO ENDZONE DISPLAYING THE CURRENT BLOCK NUMBER IN THE AC,

9,11 ROUTINE TO ROCK DECTAPE 0 (SA=2237)

THIS ROUTINE ROCKS DECTAPE 0 FOR A DISTANCE DETERMINED BY THE CONTENTS OF THE SWITCH REGISTER, THIS ROUTINE CAN BE USED TO CHECK "UP TO SPEED", SINGLE LINE FLAG, AND QUAD LINE FLAG LOGIC,

9,12 ROUTINE TO RUN DECTAPE FROM ENDZONE TO ENDZONE AND CHECK THE MARK TRACK IN BLOCKS (SA=2400)

THIS ROUTINE RUNS DECTAPE 0 FROM ENDZONE TO ENDZONE AND CHECKS THE CONTENTS OF THE MARK TRACK ON TAPE IN THE BLOCKS ON TAPE,

9,13 CHECK SELECT ERROR CIRCUITRY TEST (SA=2200, 2600)

THIS TEST CHECKS THE "SELECT ERROR" CIRCUITRY OF THE TD8E CONTROL UNIT 0 IS "ON-LINE", UNIT 1 IS "OFF-LINE" OR NO-EXISTANT, FUNCTIONS CHECKED ARE; "SELECT ERROR" STATUS FROM UNIT 1; "SELECT ERROR" PREVENTING "WRITE" FROM SETTING; NO "SELECT ERROR" FROM UNIT 0,

9,14 CHECK WRITE LOCK OUT CIRCUITRY TEST (SA=2673)

THIS TEST CHECKS THE "WRITE LOCK OUT" CIRCUITRY OF THE TD8E CONTROL, UNIT 0 IS "ON-LINE" AND "WRITE LOCKED", FUNCTIONS CHECKED ARE; "WRITE-LOCK" STATUS FROM UNIT 0; WRITE LOCK STATUS PREVENTING "WRITE FROM SETTING,

"OK" IS PRINTED ON THE TELEPRINTER AFTER THE TWO TESTS DESCRIBED IN 9,14 AND 9,15 ABOVE ARE COMPLETED,

9,15 DATA TRANSFER TEST (SA=3000)

DATA TRANSFER TEST$ IS A SERIES OF ROUTINES WHICH CHECK THE READ * WRITE * SEARCH CAPABILITIES OF THE TD8E CONTROL. EIGHT BASIC DATA PATTERNS ARE USED FOR DATA TRANSFER, THESE ARE; A BUFFER FULL OF 0'S; A BUFFER FULL OF *1'S; A BUFFER FULL OF 2525'S; A BUFFER FULL OF THE DATA PATTERN 2225, 5522, 2555, REPEATED; A BUFFER FULL OF INCREMENT BY 1 DATA PATTERN; A BUFFER FULL OF DECREMENT BY 1 DATA PATTERN; A BUFFER FULL OF 6161'S; A BUFFER FULL OF 3434'S.
DATA TRANSFERS ARE PERFORMED IN BOTH THE FORWARD AND REVERSE DIRECTION, DATA IS WRITTEN IN THE FORWARD DIRECTION, FIRST INTO BLOCK 0, THE SEQUENCE OF OPERATIONS IS; WRITE DATA IN THE FORWARD DIRECTION; READ DATA IN THE FORWARD DIRECTION, CHECK CHECKSUM AND DATA; READ DATA IN THE REVERSE DIRECTION, CHECK CHECKSUM ONLY, THIS SEQUENCE IS REPEATED EVERY 100 BLOCKS (BLOCK 0, 100, 200, 300, ETC) UP TO AND INCLUDING BLOCK 2700, IF IT IS DESIRED TO GO A DIFFERENT NUMBER OF BLOCKS FORWARD CHANGE LOCATION 3154 TO THE DESIRED NUMBER OF BLOCKS.

DATA IS THEN WRITTEN IN THE REVERSE DIRECTION, FIRST INTO BLOCK 2701, THE SEQUENCE OF OPERATIONS IS; WRITE DATA IN THE REVERSE DIRECTION; READ DATA IN REVERSE DIRECTION, CHECK CHECKSUM AND DATA; READ DATA IN THE FORWARD DIRECTION, CHECK CHECKSUM ONLY, THIS SEQUENCE IS REPEATED EVERY 100 BLOCKS (BLOCK 2701, 2601, 2501, 2401, ETC) DOWN TO AND INCLUDING BLOCK 1, IF IT IS DESIRED TO GO A DIFFERENT NUMBER OF BLOCKS REVERSE CHANGE LOCATION 3146 TO THE 2'S COMPLEMENT OF THE DESIRED NUMBER OF BLOCKS.

AFTER UNIT 0 HAS BEEN COMPLETELY TRAVERSED ONCE (FORWARD AND BACKWARD), UNIT 1 WILL BE RUN, IF AVAILABLE, THE PROGRAM WILL THEN PROCEED TO THE NEXT DATA PATTERN AND UNIT 0 AGAIN, AFTER ALL 8 DATA PATTERNS HAVE BEEN EXERCISED ON BOTH UNITS, THE PROGRAM WILL PRINT "PASS 'N' COMPLETE" ON THE TELEPRINTER, THEN PROCEED BACK TO THE FIRST DATA PATTERN.

```
                    /TD8E DIAGNOSTIC
                    /
                    /COPYRIGHT 1971
                    /DIGITAL EQUIP. CORP.
                    /MAYNARD, MASS.

                    /DECTAPE COMMANDS

            6771    SDSS=6771                       /SKIP ON SINGLE LINE FLAG
            6772    SDST=6772                       /SKIP ON TIMING ERROR
            6773    SDSQ=6773                       /SKIP ON QUADRUPLE LINE FLAG
            6774    SDLC=6774                       /LOAD COMMAND REGISTER
            6775    SDLD=6775                       /LOAD DATA REGISTER, CLEAR FLAGS
            6776    SDRC=6776                       /READ COMMAND REGISTER AND MARK TRACK, CLEAR FLAGS
            6777    SDRD=6777                       /READ DATA REGISTER, CLEAR FLAGS

                    /NEW PDP-8E INSTRUCTIONS
            6007    CAF=6007                        /CLEAR ALL FLAGS (GENERATE INITIALIZE)

            0017    *17

0017  0000  AUTO,   0
0020  0000  OUT,    0
0021  0000  IN,     0
0022  0000  CNTR1,  0
0023  0000  CNTR2,  0
0024  0000  GOOD,   0
0025  0000  HEAD1,  0
0026  0000  HEAD2,  0
0027  0000  BLK,    0
0030  0000  FILPNT, 0
                    /SWITCH OPTIONS:
                    /SR0(1)!        LOOP ON CURRENT SUBTEST
                    /SR1(1)!        LOOP ON CURRENT TEST
                    /SR2(1)!        LOOP ON CONTROL TESTS
                    /SR3(1)!        DON'T PRINT ERRORS
                    /SR4(1)!        DON'T HALT ON ERRORS
                    /SR11(1)!       SINGLE UNIT TRANSPORT (0=ONLY)

0031  0000  TYPE,   0
0032  6046          TLS
0033  6041          TSF
0034  5033          JMP     .-1
0035  6042          TCF
0036  7200          CLA
0037  5431          JMP I   TYPE
0040  0000  CRLF,   0
0041  1177          TAD     (215
0042  4031          JMS     TYPE
0043  1176          TAD     (212
0044  4031          JMS     TYPE
0045  5440          JMP I   CRLF
0046  0000  LOOP1,  0
0047  7604          LAS
```



```
0050  7004          RAL
0051  7700          SMA CLA
0052  2046          ISZ     LOOP1
0053  5446          JMP I   LOOP1


0054  7770  M10,    -10
0055  0000  BLKTRY, 0
0056  0000  DISDL,  0
0057  0000  DISDA,  0
0060  0000  BLKCN,  0

0061  0000  BLKREV, 0
0062  1175          TAD     (3000
0063  1774'         TAD     UNIT
0064  6774  IOT172, SDLC
0065  4773'         JMS     RDQUAD
0066  4773'         JMS     RDQUAD
0067  5461          JMP  I  BLKREV

0070  0000  BLKEND, 0
0071  6771  IOT173, SDSS
0072  5071          JMP     .-1
0073  6776  IOT174, SDRC
0074  0172          AND     (77
0075  1171          TAD     (-22
0076  7640          SZA     CLA
0077  5071          JMP     .+6
0100  5470          JMP  I  BLKEND

0101  0000  BLKSER, 0
0102  6771  IOT175, SDSS
0103  5102          JMP     .-1
0104  6777  IOT176, SDRD
0105  3057          DCA     DISDA
0106  6776  IOT177, SDRC
0107  0172          AND     (77
0110  1170          TAD     (-26
0111  7640          SZA     CLA
0112  5102          JMP     .+10
0113  5501          JMP  I  BLKSER

      0200    PAGE
0200  5777'         JMP     SELECT          /GO TO OPERATOR INTERVENTION TESTS FIRST
                    /ROUTINE TO CHECK THE LOADING AND READING OF THE DATA REGISTER

0201  7300  DATREG, CLA CLL
0202  3020          DCA     OUT             /START WITH 0
0203  1376          TAD     (MESS1
0204  3025          DCA     HEAD1
0205  1020          TAD     OUT
0206  7040          CMA
0207  6775  IOT1,   SDLD                    /LOAD  DATA REGISTER WITH
0210  7200          CLA                     /COMPLEMENT OF DATA
0211  1020          TAD     OUT
```

```
0212  6775        IOT2,   SDLD                    /LOAD DATA REGISTER WITH DATA
0213  7200                CLA
0214  6777        IOT3,   SORD                    /READ DATA REGISTER
0215  3021                DCA     IN
0216  7604                LAS
0217  7710                SPA CLA                 /LOOP?
0220  5218                JMP     DATREG+7        /YES
0221  1021                TAD     IN              /COMPARE DATA IN
0222  7041                CIA
0223  1020                TAD     OUT             /WITH DATA SENT OUT
0224  7650                SNA CLA                 /EQUAL?
0225  5232                JMP     DATLUP          /YES
0226  4775'               JMS     ERROR1
0227  7604                LAS
0230  7710                SPA CLA                 /LOOP?
0231  5218                JMP     DATREG+7        /YES
0232  2020        DATLUP, ISZ     OUT             /INCREMENT NUMBER TO BE SENT
0233  5205                JMP     DATREG+4        /GO BACK TO DO NEXT NUMBER
0234  4046                JMS     LOOP1
0235  5201                JMP     DATREG

                        /ROUTINE TO CHECK THE LOADING AND READING OF THE COMMAND REGISTER

0236  7300        COMREG, CLA CLL
0237  3020                DCA     OUT             /START WITH 0
0240  1374                TAD     (MESS2
0241  3025                DCA     HEAD1
0242  1020                TAD     OUT
0243  0373                AND     (6400
0244  6774        IOT4,   SDLC                    /LOAD COMMAND REGISTER WITH DATA
0245  7200                CLA
0246  6776        IOT5,   SORC                    /READ COMMAND REGISTER
0247  0372                AND     (7400           /MASK TO CLR, BITS
0250  3021                DCA     IN              /AND STORE
0251  7604                LAS
0252  7710                SPA CLA                 /LOOP
0253  5242                JMP     COMREG+4        /YES

0254  1020                TAD     OUT             /GET GOOD WORD
0255  0373                AND     (6400           /MASK OUT GO BIT
0256  7041                CIA
0257  1021                TAD     IN              /COMPARE IT WITH  WORD IN
0260  7650                SNA CLA                 /BITS OK?
0261  5266                JMP     CLOOP           /YES
0262  4775'               JMS     ERROR1
0263  7604                LAS
0264  7710                SPA CLA
0265  5242                JMP     COMREG+4
0266  1020        CLOOP,  TAD     OUT
0267  1371                TAD     (400
0270  7450                SNA
0271  5303                JMP     INITST+2
0272  3020                DCA     OUT
0273  7604                LAS
0274  7010                RAR                     /MOVE SINGLE UNIT BIT INTO LINK
```



```
0275  7420                SNL CLA                 /SINGLE UNIT
0276  5242                JMP     COMREG+4        /NO
0277  7010                RAR
0300  1020                TAD     OUT             /YES, WORKING
0301  7040                SZA CLA         /ON 2ND UNIT?
0302  5242                JMP     COMREG+4        /NO
0303  4046                JMS     LOOP1
0304  5236                JMP     COMREG
0305  7300        INITST, CLA CLL                 /TEST INIT TO CLEAR CR
0306  1370                TAD     (MESS3
0307  3025                DCA     HEAD1
0310  1367                TAD     (MESS4
0311  3026                DCA     HEAD2
0312  1373                TAD     (6400
0313  6774        IOT6,   SDLC                    /LOAD CR WITH 74
0314  6007                CAF                     /CLEAR CR
0315  7604                LAS
0316  7710                SPA CLA                 /LOOP?
0317  5305                JMP     INITST          /YES
0320  6776        IOT7,   SORC                    /READ CR
0321  0372                AND     (7400
0322  7650                SNA CLA                 /CR BITS 0?
0323  5330                JMP     .+5             /YES, OK
0324  4766'               JMS     ERROR2          /NO, ERROR, INIT (CAF) DID NOT CLEAR CR
0325  7604                LAS
0326  7710                SPA CLA                 /LOOP?
0327  5305                JMP     INITST          /YES
0330  4046                JMS     LOOP1
0331  5305                JMP     INITST
0332  5771'               JMP     CHKCLA
0366  0537
0367  5054
0370  5044
0371  0400
0372  7400
0373  6400
0374  5021
0375  0476
0376  5000
0377  2600
      0400  PAGE

                        /CHECK SDLC, SDLD, SORC, SORD TO CLEAR AC AT PROPER TIME (OR NOT AT ALL)

0400  7300        CHKCLA, CLA CLL
0401  1377                TAD     (MESS5
0402  3025                DCA     HEAD1
0403  1376                TAD     (MESS6
0404  3026                DCA     HEAD2
0405  1167        CSDLC,  TAD     (6777           /SET AC TO 6777
0406  6774        IOT8,   SDLC
0407  7650                SNA CLA                 /DID SDLC CLEAR AC (AC SHOULD CLEAR)?
0410  5215                JMP     .+5             /YES
```

```
0411  7604          LAS                     /NO,ERROR
0412  7710          SPA CLA                 /LOOP?
0413  5205          JMP     CSDLC           /NO
0414  4337          JMS     ERROR2          /ERROR
0415  7604          LAS
0416  7710          SPA CLA                 /LOOP?
0417  5205          JMP     CSDLC           /YES
0420  1375          TAD     (MESS7
0421  3026          DCA     HEAD2
0422  7300  CSDRC,  CLA CLL
0423  6774  IOT9,   SDLC                    /LOAD COMMAND REGISTER WITH 0
0424  7240          CLA CMA                 /SET AC TO ALL 1'S
0425  6776  IOT10,  SDRC                    /READ COMMAND REGISTER
0426  7650          SNA CLA                 /ALL ZEROS (AC SHOULD CLEAR BEFORE READING)?
0427  5234          JMP     .+5             /YES
0430  7604          LAS                     /NO,ERROR
0431  7710          SPA CLA                 /LOOP?
0432  5222          JMP     CSDRC           /YES
0433  4337          JMS     ERROR2          /ERROR
0434  7604          LAS
0435  7710          SPA CLA                 /LOOP?
0436  5222          JMP     CSDRC           /YES
0437  1374          TAD     (MESS8
0440  3026          DCA     HEAD2
0441  7240  CSDLD,  CLA CMA                 /SET AC TO ALL 1'S
0442  6775  IOT11,  SDLD                    /LOAD DATA REGISTER
0443  7640          SZA CLA                 /DID SDLD CLEAR AC (AC SHOULD NOT CLEAR)?
0444  5251          JMP     .+5             /NO, ALL OK
0445  7604          LAS                     /YES,ERROR
0446  7710          SPA CLA                 /LOOP?
0447  5241          JMP     CSDLD           /YES
0450  4337          JMS     ERROR2          /ERROR
0451  7604          LAS
0452  7710          SPA CLA                 /LOOP?
0453  5241          JMP     CSDLD           /YES
0454  1373          TAD     (MESS9
0455  3026          DCA     HEAD2

0456  7300  CSDRD,  CLA CLL
0457  6775  IOT12,  SDLD                    /LOAD REGISTER WITH 0
0460  7240          CLA CMA                 /SET AC TO ALL 1'S
0461  6777  IOT13,  SDRD                    /READ DATA REGISTER
0462  7650          SNA CLA                 /ALL ZEROS(AC SHOULD CLEAR BEFORE READING)?
0463  5270          JMP     .+5             /YES
0464  7604          LAS                     /NO,ERROR
0465  7710          SPA CLA                 /LOOP?
0466  5256          JMP     CSDRD           /YES
0467  4337          JMS     ERROR2          /ERROR
0470  7604          LAS
0471  7710          SPA CLA                 /LOOP?
0472  5256          JMP     CSDRD           /YES
0473  4046          JMS     LOOP1
0474  5200          JMP     CHKCLA
0475  5772'         JMP     SINGLE
```



```
                    /ERROR HANDLER ROUTINE-DATA WORD TYPEOUTS

0476  0000  ERROR1, 0
0477  7604          LAS                     /GET SR
0500  0371          AND     (400            /MASK TO TYPEOUT BIT
0501  7640          SZA CLA                 /TYPE OUT ERROR?
0502  5325          JMP     ERR1HT          /NO
0503  4040          JMS     CRLF            /YES
0504  1025          TAD     HEAD1
0505  7450          SNA                     /TYPE OUT HEADER?
0506  5315          JMP     .+7             /NO
0507  4770'         JMS     MESAGE          /YES, PRINT HEADER
0510  3025          DCA     HEAD1
0511  4040          JMS     CRLF            /CR-LF
0512  1367          TAD     (FORMT1         /PRINT REST OF FORMAT
0513  4770'         JMS     MESAGE
0514  4040          JMS     CRLF            /CR-LF
0515  1020          TAD     OUT             /PRINT GOOD DATA
0516  0366          AND     (6400
0517  4765'         JMS     OPRINT
0520  1364          TAD     (240            /SPACE
0521  4031          JMS     TYPE
0522  1021          TAD     IN              /PRINT BAD DATA
0523  4765'         JMS     OPRINT
0524  4040          JMS     CRLF            /CRLF
0525  7604  ERR1HT, LAS                     /GET SR
0526  0363          AND     (200            /MASK TO HALT BIT
0527  7650          SNA CLA                 /STOP?
0530  7402  E1HLT,  HLT                     /NO
0531  5676          JMP I   ERROR1          /EXIT

0532  0717  FORMT1, TEXT    "GOOD BAD"
0533  1704
0534  4002
0535  0104
0536  0000

                    /ERROR HANDLER - NO DATA WORD TYPEOUTS

0537  0000  ERROR2, 0
0540  7604          LAS                     /GET SR
0541  0371          AND     (400            /MASK TO TYPEOUT BIT
0542  7640          SZA CLA                 /TYPE OUT ERROR?
0543  5356          JMP     ERR2HT          /NO
0544  4040          JMS     CRLF            /YES
0545  1025          TAD     HEAD1
0546  7450          SNA                     /TYPE OUT HEADER
0547  5353          JMP     .+4             /NO
0550  4770'         JMS     MESAGE          /YES
0551  3025          DCA     HEAD1
0552  4040          JMS     CRLF
0553  1026          TAD     HEAD2           /TYPE OUT ERROR MESSAGE
0554  4770'         JMS     MESAGE
0555  4040          JMS     CRLF
0556  7604  ERR2HT, LAS                     /GET SR
```

```
8557  0363          AND     (288            /MASK TO HALT BIT
8560  7650          SNA CLA                 /STOP?
8561  7402          HLT                     /YES
8562  5737          JMP I   ERROR2

8563  0200
8564  0240
8565  2316
8566  6466
8567  0532
8570  2264
8571  0400
8572  0666
8573  5161
8574  5151
8575  5136
8576  5123
8577  5076
      0600          PAGE

                    /CHECK SINGLE LINE SKIP INSTRUCTION AND LOGIC

8600  7300    SINGLE, CLA CLL
8601  1377          TAD     (MESS10
8602  3025          DCA     HEAD1
8603  1376          TAD     (MESS11
8604  3026          DCA     HEAD2
8605  6774          SDLC                    /CLEAR ALL FLAGS INITIALLY
8606  6771    IOT14,  SDSS                  /SKIP ON SINGLE LINE
8607  7410          SKP
8610  4775'         JMS     ERROR2          /ERROR, SDSS SHOULD NOT HAVE SKIPPED
8611  1374          TAD     (1000
8612  6774    IOT15,  SDLC                  /LOAD COMMAND REGISTER WITH UB,FWD,GO,READ
8613  1373          TAD     (MESS12
8614  3026          DCA     HEAD2

8615  7200    SINGL8, CLA
8616  1372          TAD     (-1000          /SET UP FOR
8617  3022          DCA     CNTR1           /A DELAY
8620  3023          DCA     CNTR2           /OF ABOUT 1 SECOND
8621  6771    IOT16,  SDSS                  /SINGLE LINE FLAG UP YET?
8622  7410          SKP                     /NO
8623  5234          JMP     SING1+2         /YES
8624  2023          ISZ     CNTR2           /NO, COUNT
8625  5221          JMP     .-4
8626  2022          ISZ     CNTR1           /DELAY OVER?
8627  5221          JMP     .-6             /NO
8630  4775'         JMS     ERROR2          /YES, NO SINGLE LINE FLAG, OR SDSS DOES NOT SKIP
8631  7604          LAS
8632  7710          SPA CLA                 /LOOP?
8633  5200          JMP     SINGLE          /YES
8634  1371          TAD     (MESS13
8635  3026          DCA     HEAD2
```



```
8636  6771    SING1,  SDSS                  /FLAG STILL UP?
8637  4775'         JMS     ERROR2          /SINGLE LINE FLAG CLEARED BY SDSS
8640  7604          LAS
8641  7710          SPA CLA                 /LOOP?
8642  5236          JMP     SING1           /YES
8643  1370          TAD     (MESS14
8644  3026          DCA     HEAD2
8645  6771    SING2,  SDSS                  /WAIT FOR SINGLE LINE FLAG
8646  5245          JMP     .-1
8647  6007          CAF                     /CLEAR FLAG WITH CAF
8650  7604          LAS
8651  7710          SPA CLA                 /LOOP?
8652  5245          JMP     SING2           /YES
8653  6771    IOT17,  SDSS                  /DID FLAG CLEAR?
8654  5261          JMP     SING3+4         /YES
8655  4775'         JMS     ERROR2          /NO, SINGLE LINE FLAG NOT CLEARED BY CAF
8656  7604          LAS
8657  7710          SPA CLA                 /LOOP?
8660  5245          JMP     SING2           /YES
8661  1374          TAD     (1000
8662  6774    IOT18,  SDLC                  /LOAD COMMAND REGISTER AGAIN

8663  1367          TAD     (MESS15
8664  3026          DCA     HEAD2
8665  6771    SING3,  SDSS                  /WAIT FOR SINGLE LINE FLAG
8666  5265          JMP     .-1
8667  7200          CLA
8670  6775    IOT19,  SDLD                  /ISSUE SDLD TO CLEAR SINGLE LINE FLAG
8671  7604          LAS
8672  7710          SPA CLA                 /LOOP?
8673  5265          JMP     SING3           /YES
8674  6771    IOT20,  SDSS                  /FLAG STILL UP?
8675  5302          JMP     SING4+2         /NO
8676  4775'         JMS     ERROR2          /YES, ERROR, SINGLE LINE FLAG NOT CLEARED BY SDLD;
8677  7604          LAS
8700  7710          SPA CLA
8701  5265          JMP     SING3
8702  1366          TAD     (MESS16
8703  3026          DCA     HEAD2
8704  6771    SING4,  SDSS                  /WAIT FOR SINGLE LINE FLAG
8705  5304          JMP     .-1
8706  7200          CLA
8707  6776    IOT21,  SDRC                  /ISSUE SDRC TO CLEAR SINGLE LINE FLAG
8710  7604          LAS
8711  7710          SPA CLA                 /LOOP?
8712  5304          JMP     SING4           /YES
8713  6771    IOT22,  SDSS                  /FLAG CLEARED?
8714  5321          JMP     SING5+2         /YES
8715  4775'         JMS     ERROR2          /NO, ERROR, SINGLE LINE FLAG NOT CLEARED BY SDRC
8716  7604          LAS
8717  7710          SPA CLA                 /LOOP?
8720  5304          JMP     SING4           /YES
8721  1365          TAD     (MESS17
8722  3026          DCA     HEAD2
```

```
0723  6771  SING5,  SDSS                    /WAIT FOR SINGLE LINE FLAG
0724  5323          JMP     .-1
0725  7200          CLA
0726  6777  IOT23,  SDRD                    /ISSUE SDRD TO CLEAR SINGLE LINE FLAG
0727  7604          LAS
0730  7710          SPA CLA                 /LOOP?
0731  5323          JMP     SING5           /YES
0732  6771  IOT24,  SDSS                    /FLAG CLEARED?
0733  5774'         JMP     SING6-2         /YES
0734  4775'         JMS     ERROR2          /NO, ERROR, SINGLE LINE FLAG NOT CLEARED BY SDRD
0735  7604          LAS
0736  7710          SPA CLA                 /LOOP?
0737  5323          JMP     SING5           /YES
0740  5774'         JMP     SING6-2
0765  5416
0766  5373
0767  5350
0770  5326
0771  5385
0772  7000
0773  5253
0774  1000
0775  0537
0776  5222
0777  5174
      1000          PAGE

1000  1377          TAD     (MESS18
1001  3026          DCA     HEAD2
1002  6771  SING6,  SDSS                    /WAIT FOR SINGLE LINE FLAG
1003  5202          JMP     .-1
1004  6772  IOT25,  SDST                    /ISSUE SDST
1005  7000          NOP
1006  6773  IOT26,  SDSQ                    /SDSQ
1007  7200          CLA
1010  1376          TAD     (1000
1011  6774  IOT27,  SDLC                    /AND SDLC
1012  6771  IOT28,  SDSS                    /DID SDST, SDSQ, OR SDLC CLEAR FLAG?
1013  7410          SKP                     /YES
1014  5221          JMP     SING7           /NO
1015  4775'         JMS     ERROR2          /ERROR, SDST, SDSQ, OR SDLC CLEARED SINGLE LINE FLAG
1016  7604          LAS
1017  7710          SPA CLA                 /LOOP?
1020  5202          JMP     SING6           /YES
1021  6774  SING7,  SDLC
1022  4046          JMS     LOOP1
1023  5774'         JMP     SINGLE


                    /CHECK QUAD LINE SKIP INSTRUCTION AND LOGIC

1024  7300  QUAD,   CLA CLL
1025  1373          TAD     (MESS19
```



```
1026  3025          DCA     HEAD1
1027  1372          TAD     (MESS20
1030  3026          DCA     HEAD2
1031  6774          SDLC                    /CLEAR ALL FLAGS INITIALLY
1032  6773  IOT29,  SDSQ                    /SKIP ON QUAD LINE
1033  7410          SKP
1034  4775'         JMS     ERROR2          /ERROR, SDSQ SHOULD NOT HAVE SKIPPED
1035  1376          TAD     (1000
1036  6774  IOT30,  SDLC                    /LOAD COMMAND REGISTER WITH UB, FND, GO, READ
1037  1371          TAD     (MESS21
1040  3026          DCA     HEAD2
1041  6777  QUAD0,  SDRD                    /CLEAR DANGLING FLAGS
1042  6771  IOT31,  SDSS                    /WAIT FOR 1ST
1043  5242          JMP     .-1             /SINGLE LINE FLAG
1044  6773  IOT32,  SDSQ                    /QUAD FLAG UP?
1045  5250          JMP     .+3             /NO
1046  4775'         JMS     ERROR2          /YES ERROR
1047  5274          JMP     QBLUP
1050  6771  IOT33,  SDSS                    /WAIT FOR SINGLE
1051  7410          SKP                     /LINE FLAG
1052  5250          JMP     .-2             /TO GO AWAY
1053  6773  IOT34,  SDSQ                    /QUAD FLAG UP?
1054  5257          JMP     .+3             /NO
1055  4775'         JMS     ERROR2          /YES, ERROR
1056  5274          JMP     QBLUP
1057  6771  IOT35,  SDSS                    /WAIT FOR NEXT
1060  5257          JMP     .-1             /SINGLE LINE FLAG
1061  6773  IOT36,  SDSQ                    /QUAD FLAG UP?
1062  5265          JMP     .+3             /NO
1063  4775'         JMS     ERROR2          /YES, ERROR
1064  5274          JMP     QBLUP
1065  1370          TAD     (MESS22
1066  3026          DCA     HEAD2
1067  6771  IOT37,  SDSS                    /WAIT FOR SINGLE
1070  7410          SKP                     /LINE FLAG
1071  5267          JMP     .-2             /TO GO AWAY
1072  6773  IOT38,  SDSQ                    /QUAD FLAG UP?
1073  4775'         JMS     ERROR2          /NO, ERROR
1074  7604  QBLUP,  LAS
1075  7710          SPA CLA
1076  5235          JMP     QUAD2-4
1077  1367          TAD     (MESS23
1100  3026          DCA     HEAD2
1101  6773  QUAD1,  SDSQ                    /FLAG STILL UP?
1102  4775'         JMS     ERROR2          /QUAD LINE FLAG CLEARED BY SDSQ
1103  7604          LAS
1104  7710          SPA CLA                 /LOOP?
1105  5301          JMP     QUAD1           /YES

1106  1366          TAD     (MESS24
1107  3026          DCA     HEAD2
1110  6773  QUAD2,  SDSQ                    /WAIT FOR QUAD FLAG
1111  5310          JMP     .-1
1112  6007          CAF                     /CLEAR THE FLAG WITH CAF
1113  7604          LAS
```

```
1114  7710          SPA CLA                 /LOOP?
1115  5310          JMP     QUAD2           /YES
1116  6773  IOT39,  SDSQ                    /DID FLAG CLEAR?
1117  5324          JMP     QUAD3-4         /YES
1120  4775'         JMS     ERROR2          /NO, QUAD LINE FLAG NOT CLEARED BY CAF
1121  7604          LAS
1122  7710          SPA CLA                 /LOOP?
1123  5310          JMP     QUAD2           /YES
1124  1376          TAD     (1000
1125  6774  IOT40,  SDLC                    /LOAD COMMAND REGISTER AGAIN
1126  1365          TAD     (MESS25
1127  3026          DCA     HEAD2
1130  6773  QUAD3,  SDSQ                    /WAIT FOR QUAD FLAG
1131  5330          JMP     .-1
1132  7200          CLA
1133  6775  IOT41,  SDLD                    /ISSUE SDLD TO CLEAR QUAD FLAG
1134  7604          LAS
1135  7710          SPA CLA                 /LOOP?
1136  5330          JMP     QUAD3           /YES
1137  6773  IOT42,  SDSQ                    /FLAG STILL UP?
1140  5764'         JMP     QUAD4-2         /NO
1141  4775'         JMS     ERROR2          /YES, ERROR, QUAD FLAG NOT CLEARED BY SDLD
1142  7604          LAS
1143  7710          SPA CLA                 /LOOP?
1144  5330          JMP     QUAD3           /YES
1145  5764'         JMP     QUAD4-2
1164  1200
1165  5656
1166  5635
1167  5615
1170  5564
1171  5546
1172  5516
1173  5471
1174  0600
1175  0537
1176  1000
1177  5441
      1200          PAGE

1200  1377          TAD     (MESS26
1201  3026          DCA     HEAD2
1202  6773  QUAD4,  SDSQ                    /WAIT FOR QUAD FLAG
1203  5202          JMP     .-1
1204  7200          CLA
1205  6776  IOT43,  SDRC                    /ISSUE SDRC TO CLEAR QUAD FLAG
1206  7604          LAS
1207  7710          SPA CLA                 /LOOP?
1210  5202          JMP     QUAD4           /YES
1211  6773  IOT44,  SDSQ                    /FLAG CLEARED?
1212  5217          JMP     QUAD5-2         /YES
1213  4776'         JMS     ERROR2          /NO, ERROR, QUAD FLAG NOT CLEARED BY SDRC
1214  7604          LAS
1215  7710          SPA CLA                 /LOOP?
```



```
1216  5202          JMP     QUAD4           /YES
1217  1375          TAD     (MESS27
1220  3026          DCA     HEAD2
1221  6773  QUAD5,  SDSQ                    /WAIT FOR QUAD FLAG
1222  5221          JMP     .-1
1223  7200          CLA
1224  6777  IOT45,  SDRD                    /ISSUE SDRD TO CLEAR QUAD LINE FLAG
1225  7604          LAS
1226  7710          SPA CLA                 /LOOP?
1227  5221          JMP     QUAD5           /YES
1230  6773  IOT46,  SDSQ                    /FLAG CLEARED?
1231  5236          JMP     QUAD6-2         /YES
1232  4776'         JMS     ERROR2          /NO, ERROR, QUAD FLAG NOT CLEARED BY SDRD
1233  7604          LAS
1234  7710          SPA CLA                 /LOOP?
1235  5221          JMP     QUAD5           /YES

1236  1374          TAD     (MESS28
1237  3026          DCA     HEAD2
1240  6773  QUAD6,  SDSQ                    /WAIT FOR QUAD FLAG
1241  5240          JMP     .-1
1242  6772  IOT47,  SDST                    /ISSUE SDST
1243  7000          NOP
1244  6771  IOT48,  SDSS                    /SDSS
1245  7000          NOP
1246  7200          CLA
1247  1373          TAD     (1000
1250  6774  IOT49,  SDLC                    /AND SDLC
1251  7604          LAS
1252  7710          SPA CLA                 /LOOP?
1253  5240          JMP     QUAD6           /YES
1254  6773  IOT50,  SDSQ                    /DID STST, SDSS, OR SDLC CLEAR FLAG?
1255  7410          SKP                     /YES
1256  5263          JMP     QUAD7-2         /NO
1257  4776'         JMS     ERROR2          /ERROR, SDST, SDSS, OR SDLC CLEARED QUAD FLAG
1260  7604          LAS
1261  7710          SPA CLA                 /LOOP?
1262  5240          JMP     QUAD6           /YES

1263  1372          TAD     (MESS29
1264  3026          DCA     HEAD2
1265  7300  QUAD7,  CLA CLL
1266  1371          TAD     (-2             /SET LOOP COUNT TO-2
1267  3022          DCA     CNTR1
1270  6775  IOT51,  SDLD                    /CLEAR QUAD FLAG FLIP/FLOPS
1271  6771  IOT52,  SDSS                    /WAIT FOR SINGLE LINE
1272  5271          JMP     .-1             /TO COME
1273  6771  IOT53,  SDSS                    /GO AWAY
1274  7410          SKP
1275  5274          JMP     .-1
1276  6771  IOT54,  SDSS                    /AND COME AGAIN
1277  5276          JMP     .-1             /
1300  2022          ISZ     CNTR1           /TWICE THRU?
1301  5270          JMP     QUAD7-3         /NO
1302  7604          LAS                     /YES
```

```
1303  7710          SPA CLA                 /LOOP?
1304  5265          JMP     QUAD7           /YES
1305  6773  IOT55,  SDSQ                    /NO, IS QUAD FLAG UP?
1306  7410          SKP                     /NO
1307  4776'         JMS     ERROR2          /YES, ERROR QUAD FLAG COUNTER FLIP/FLOPS NOT CLEARED
1310  7604          LAS                     /BY SDLC
1311  7710          SPA CLA                 /LOOP?
1312  5265          JMP     QUAD7           /YES
1313  4046          JMS     LOOP1
1314  5770'         JMP     QUAD

                    /CHECK TIMING ERROR SKIP INSTRUCTION AND LOGIC

1315  7300  TIMING, CLA CLL
1316  1367          TAD     (MESS38
1317  3025          DCA     HEAD1
1320  1366          TAD     (MESS31
1321  3026          DCA     HEAD2
1322  6774          SDLC                    /CLEAR ALL FLAGS INITIALLY
1323  1373          TAD     (1000
1324  6774  IOT56,  SDLC                    /LOAD COMMAND REGISTER WITH U0,FWD,GO,READ
1325  6771  IOT57,  SDSS                    /WAIT FOR SINGLE
1326  5325          JMP     .-1             /LINE FLAG
1327  6772  IOT58,  SDST                    /SKIP ON TIMING ERROR
1330  7410          SKP
1331  4776'         JMS     ERROR2          /ERROR, SDST SHOULD NOT HAVE SKIPPED
1332  1365          TAD     (MESS32
1333  3026          DCA     HEAD2
1334  6773  TIME0,  SDSQ                    /WAIT FOR QUAD FLAG
1335  5334          JMP     .-1
1336  7200          CLA
1337  3022          DCA     CNTR1
1340  2022          ISZ     CNTR1           /WAIT A WHILE SO THAT TIMING ERROR
1341  5340          JMP     .-1             /CAN SET
1342  6772  IOT59,  SDST                    /TIMING ERROR SET?
1343  7410          SKP                     /NO
1344  5351          JMP     TIME1+2         /YES
1345  4776'         JMS     ERROR2          /ERROR, TIMING ERROR NOT SET IN READ MODE
1346  7604          LAS
1347  7710          SPA CLA                 /LOOP?
1350  5334          JMP     TIME0           /YES
1351  1364          TAD     (MESS33
1352  3026          DCA     HEAD2
1353  6772  TIME1,  SDST                    /TIMING ERROR STILL SET?
1354  4776'         JMS     ERROR2          /TIMING ERROR CLEARED BY SDST
1355  7604          LAS
1356  7710          SPA CLA                 /LOOP?
1357  5353          JMP     TIME1           /YES
1360  5763'         JMP     TIME2+2
1361  1400
1362  6140
1363  6101
1364  6052
1365  6026
```



```
1370  1024
1371  7776
1372  7773
1373  1000
1374  7744
1375  7722
1376  0537
1377  7700
      1400          PAGE

1400  1377          TAD     (MESS34
1401  3026          DCA     HEAD2
1402  6772  TIME2,  SDST                    /WAIT FOR TIMING ERROR
1403  5202          JMP     .-1
1404  6007          CAF                     /CLEAR FLAG WITH CAF
1405  7604          LAS
1406  7710          SPA CLA                 /LOOP?
1407  5202          JMP     TIME2           /YES
1410  6772  IOT60,  SDST                    /DID FLAG CLEAR?
1411  5216          JMP     TIME3+4
1412  4776'         JMS     ERROR2          /NO, TIMING ERROR NOT CLEARED BY CAF
1413  7604          LAS
1414  7710          SPA CLA                 /LOOP?
1415  5202          JMP     TIME2           /YES
1416  1373          TAD     (1000
1417  6774  IOT61,  SDLC                    /LOAD COMMAND REGISTER AGAIN

1420  1374          TAD     (MESS35
1421  3026          DCA     HEAD2
1422  6772  TIME3,  SDST                    /WAIT FOR TIMING ERROR
1423  5222          JMP     .-1
1424  6776  IOT62,  SDRC                    /READ DECTAPE COMMAND REGISTER FOR STATUS
1425  3021          DCA     IN              /SAVE
1426  7604          LAS
1427  7710          SPA CLA                 /LOOP?
1430  5222          JMP     TIME3           /YES
1431  1021          TAD     IN              /GET STATUS BACK AGAIN
1432  0373          AND     (100            /MASK TO BIT 5
1433  7440          SZA                     /TIMING ERROR STATUS SET?
1434  5241          JMP     TIME4+4         /YES,  OK
1435  4776'         JMS     ERROR2          /NO, ERROR, TIMING ERROR STATUS NOT SET
1436  7604          LAS
1437  7710          SPA CLA                 /LOOP?
1440  5222          JMP     TIME3           /YES
1441  1373          TAD     (1000
1442  6774  IOT63,  SDLC                    /LOAD COMMAND REGISTER AGAIN
1443  1372          TAD     (MESS36
1444  3026          DCA     HEAD2
1445  6772  TIME4,  SDST                    /WAIT FOR TIMING ERROR
1446  5245          JMP     .-1
1447  6774  IOT64,  SDLC                    /CLEAR FLAG WITH SDLC
1450  7604          LAS
1451  7710          SPA CLA                 /LOOP?
1452  5245          JMP     TIME4           /YES
```

```
1453  6772  IOT65,  SDST                    /DID FLAG CLEAR?
1454  5261           JMP     TIME5+2         /YES
1455  4776'          JMS     ERROR2          /NO, TIMING ERROR NOT CLEARED BY SDLC
1456  7604           LAS
1457  7710           SPA CLA                 /LOOP?
1460  5245           JMP     TIME4

1461  1371           TAD     (MESS37
1462  3026           DCA     HEAD2
1463  7300  TIME5,   CLA CLL
1464  1370           TAD     (3000
1465  6774  IOT66,   SDLC                    /GET TAPE MOVING BACKWARD
1466  6771  IOT67,   SDSS                    /WAIT FOR END ZONE
1467  5266           JMP     .-1
1470  6776  IOT68,   SDRC
1471  0367           AND     (77
1472  1366           TAD     (-22
1473  7640           SZA CLA
1474  5266           JMP     .-0
1475  6776  IOT69,   SDRC                    /SET "WRITE"
1476  0365           AND     (7000
1477  1364           TAD     (400
1500  6774  IOT70,   SDLC
1501  3022           DCA     CNTR1
1502  2022           ISZ     CNTR1           /WAIT A WHILE
1503  5302           JMP     .-1
1504  6772  IOT71,   SDST                    /TIMING ERROR?
1505  4776'          JMS     ERROR2          /NO,ERROR
1506  1363           TAD     (MESS38
1507  3026           DCA     HEAD2
1510  6776  IOT72,   SDRC                    /YES, READ STATUS
1511  0364           AND     (400
1512  7640           SZA CLA                 /"WRITE" CLEARED
1513  4776'          JMS     ERROR2          /NO,ERROR
1514  7604           LAS
1515  7710           SPA CLA                 /LOOP?
1516  5263           JMP     TIME5           /YES


1517  1362           TAD     (MESS39
1520  3026           DCA     HEAD2
1521  7300  TIME6,   CLA CLL
1522  1375           TAD     (1000           /SET UNIT 0 RUNNING FORWARD
1523  6774  IOT73,   SDLC
1524  6771  IOT74,   SDSS                    /WAIT FOR "UP TO SPEED"
1525  5324           JMP     .-1
1526  1361           TAD     (-0
1527  3023           DCA     CNTR2
1530  3022           DCA     CNTR1
1531  6776  IOT75,   SDRC                    /ISSUE MANY SDRC,SDRD,SDLC'S
1532  6777  IOT76,   SDRD
1533  6775  IOT77,   SDLD
1534  2022           ISZ     CNTR1
1535  5331           JMP     .-0
```



```
1536  2023           ISZ     CNTR2
1537  5331           JMP     .-6
1540  7604           LAS
1541  7710           SPA CLA                 /LOOP?
1542  5327           JMP     TIME6+6         /YES
1543  6772  IOT78,   SDST                    /TIMING ERROR?
1544  4776'          JMS     ERROR2          /NO, ERROR
1545  7604           LAS
1546  7710           SPA CLA                 /LOOP?
1547  5321           JMP     TIME6           /YES
1550  4046           JMS     LOOP1
1551  5760'          JMP     TIMING
1552  5757'          JMP     UTSMRK
1557  1600
1560  1315
1561  7773
1562  6324
1563  6276
1564  0400
1565  7000
1566  7756
1567  0077
1570  3000
1571  6252
1572  6231
1573  0100
1574  6177
1575  1000
1576  0537
1577  6157
      1600           PAGE

                     /CHECK UP TO SPEED CIRCUITRY TO CLEAR MARK TRACK WINDOW

1600  7300  UTSMRK,  CLA CLL                 /CLEAR STOP/GO BIT
1601  1377           TAD     (MESS43
1602  3025           DCA     HEAD1
1603  1376           TAD     (MESS44
1604  3026           DCA     HEAD2
1605  6774  IOT82,   SDLC
1606  1375           TAD     (1000           /SET STOP/GO BIT
1607  6774  IOT83,   SDLC
1610  7604           LAS
1611  7710           SPA CLA                 /LOOP?
1612  5200           JMP     UTSMRK          /YES
1613  6776  IOT84,   SDRC                    /READ MARK TRACK
1614  0374           AND     (77
1615  7440           SZA                     /ZERO?
1616  4773'          JMS     ERROR2          /NO, ERROR
1617  7604           LAS
1620  7710           SPA CLA                 /LOOP?
1621  5200           JMP     UTSMRK          /YES
```

```
1622  1372          TAD     (MESS45
1623  3026          DCA     HEAD2
1624  7300  UTSMK1, CLA CLL
1625  1375          TAD     (1000           /SET STOP/GO BIT,
1626  6774  IOT85,  SDLC
1627  6771  IOT86,  SDSS                    /SINGLE LINE FLAG?
1630  5227          JMP     .-1             /NO
1631  6776  IOT87,  SDRC                    /YES, READ MARK TRACK
1632  0374          AND     (77
1633  7650          SNA CLA                 /ZERO?
1634  5227          JMP     .-5             /YES, TRY AGAIN
1635  6774  IOT88,  SDLC                    /CLEAR STOP/GO BIT
1636  7604          LAS
1637  7710          SPA CLA                 /LOOP?
1640  5224          JMP     UTSMK1          /YES
1641  6776  IOT89,  SDRC                    /READ MARK TRACK
1642  0374          AND     (77
1643  7440          SZA                     /ZERO?
1644  4773'         JMS     ERROR2          /NO, ERROR
1645  7604          LAS
1646  7710          SPA CLA                 /LOOP?
1647  5224          JMP     UTSMK1          /YES
1650  1371          TAD     (MESS46
1651  3026          DCA     HEAD2
1652  7300  UTSMK2, CLA CLL
1653  1370          TAD     (3000           /SET STOP/GO AND FWD/REV
1654  6774  IOT90,  SDLC
1655  6771  IOT91,  SDSS
1656  5255          JMP     .-1
1657  6776  IOT92,  SDRC
1660  0374          AND     (77
1661  7650          SNA CLA
1662  5255          JMP     .-5
1663  1375          TAD     (1000           /CLEAR FWD/REV (BIT1)
1664  6774  IOT93,  SDLC
1665  7604          LAS
1666  7710          SPA CLA
1667  5252          JMP     UTSMK2
1670  6776  IOT94,  SDRC
1671  0374          AND     (77
1672  7440          SZA                     /MARK TRACK ZERO?
1673  4773'         JMS     ERROR2          /NO, ERROR
1674  7604          LAS
1675  7710          SPA CLA
1676  5252          JMP     UTSMK2

1677  1367          TAD     (MESS47
1700  3026          DCA     HEAD2
1701  7300  UTSMK3, CLA CLL
1702  1375          TAD     (1000           /SET STOP/GO, CLEAR FWD/REV (BIT1)
1703  6774  IOT95,  SDLC
1704  6771  IOT96,  SDSS
1705  5304          JMP     .-5
1706  6776  IOT97,  SDRC
```



```
1707  0374          AND     (77
1710  7650          SNA CLA
1711  5304          JMP     .-5
1712  1370          TAD     (3000           /SET FWD/REV (BIT 1)
1713  6774  IOT98,  SDLC
1714  7604          LAS
1715  7710          SPA CLA
1716  5301          JMP     UTSMK3
1717  6776  IOT99,  SDRC
1720  0374          AND     (77
1721  7440          SZA                     /MARK TRACK ZERO?
1722  4773'         JMS     ERROR2          /NO, ERROR
1723  7604          LAS
1724  7710          SPA CLA
1725  5301          JMP     UTSMK3
1726  5766'         JMP     UTSMK4+2
1766  2000
1767  6517
1770  3000
1771  6466
1772  6441
1773  0537
1774  0077
1775  1000
1776  6415
1777  6352
      2000          PAGE

2000  1377          TAD     (MESS48
2001  3026          DCA     HEAD2
2002  7300  UTSMK4, CLA CLL
2003  1376          TAD     (1000           /SET STOP/GO, CLEAR UNIT (BIT0)
2004  6774  IOT100, SDLC
2005  6771  IOT101, SDSS
2006  5205          JMP     .-1
2007  6776  IOT102, SDRC
2010  0375          AND     (77
2011  7650          SNA CLA
2012  5205          JMP     .-5
2013  1374          TAD     (5000           /SET UNIT (BIT0)
2014  6774  IOT103, SDLC
2015  7604          LAS
2016  7710          SPA CLA
2017  5202          JMP     UTSMK4
2020  6776  IOT104, SDRC
2021  0375          AND     (77
2022  7440          SZA                     /MARK TRACK 0?
2023  4773'         JMS     ERROR2          /NO
2024  7604          LAS
2025  7710          SPA CLA
2026  5202          JMP     UTSMK4
2027  7604          LAS
2030  7010          RAR
2031  7630          SZL CLA                 /IS THERE A UNIT1?
```

```
2032  5262          JMP     UT9MK6          /NO


2033  1372          TAD     (MESS49
2034  3026          DCA     HEAD2
2035  7388  UT9MK5, CLA CLL
2036  1374          TAD     (5888           /SET STOP/GO, UNIT (BIT8)
2037  6774  IOT105, SDLC
2040  6771  IOT106, SDSS
2041  5248          JMP     .-1
2042  6776  IOT107, SDRC
2043  0375          AND     (77
2044  7658          SNA CLA
2045  5248          JMP     .-5
2046  1376          TAD     (1888           /CLEAR UNIT (BIT8)
2047  6774  IOT108, SDLC
2050  7604          LAS
2051  7710          SPA CLA
2052  5235          JMP     UT9MK5
2053  6776  IOT109, SDRC
2054  0375          AND     (77
2055  7448          SZA                     /MARK TRACK ZERO?
2056  4773'         JMS     ERROR2          /NO, ERROR
2057  7604          LAS
2060  7710          SPA CLA
2061  5235          JMP     UT9MK5
2062  4046  UT9MK6, JMS     LOOP1
2063  5771'         JMP     UT9MRK
2064  1378          TAD     (4888
2065  6774  IOT110, SDLC                    /STOP UNIT 1 IF MOVING
2066  7604          LAS
2067  7006          RTL
2070  7710          SPA CLA
2071  5767'         JMP     DATREG
2072  5766'         JMP     XFER
                    /ROUTINE TO SEARCH AND FIND ALL BLOCK NUMBERS
                    /THE RIGHT HAND REEL MUST HAVE AT LEAST FOUR TURNS OF TAPE ON IT
      2100    *2100
2100  7300  BLKCH,  CLA     CLL
2101  3056          DCA     DISBL
2102  1166          TAD     [-8702
2103  3060          DCA     BLKCN
2104  4061          JMS     BLKREV
2105  4070          JMS     BLKEND
2106  1165          TAD     [1888
2107  1765'         TAD     UNIT
2110  6774  IOT171, SDLC
2111  4764'         JMS     RDQUAD
2112  4764'         JMS     RDQUAD
2113  4101  FBLKCT, JMS     BLKSER
2114  1057          TAD     DISDA
2115  7041          CIA
2116  1056          TAD     DISBL
2117  7640          SZA     CLA
2120  5346          JMP     BLKERR  /BLOCKS DIDN'T COMPARE
```



```
2121  2056          ISZ     DISBL
2122  2060          ISZ     BLKCN
2123  5313          JMP     FBLKCT
2124  4070          JMS     BLKEND
2125  1164          TAD     [2781
2126  3056          DCA     DISBL
2127  1166          TAD     [-8702
2130  3060          DCA     BLKCN
2131  4061          JMS     BLKREV
2132  4101  RBLKCT, JMS     BLKSER
2133  1057          TAD     DISDA
2134  7041          CIA
2135  1056          TAD     DISBL
2136  7640          SZA     CLA
2137  5346          JMP     BLKERR
2140  7040          CMA
2141  1056          TAD     DISBL
2142  3056          DCA     DISBL
2143  2060          ISZ     BLKCN
2144  5332          JMP     RBLKCT
2145  5300          JMP     BLKCH

2146  7300  BLKERR, CLA     CLL
2147  1056          TAD     DISBL
2150  7402          HLT             /AC=THE BLOCK NUMBER THAT WAS BEING SEARCHED FOR
2151  7200          CLA
2152  1057          TAD     DISDA
2153  7402          HLT             /AC=THE BLOCK NUMBER THAT WAS FOUND
2154  5300          JMP     BLKCH   /RETURN TO START OF ROUTINE

2164  4707
2165  2234
2166  3888
2167  0201
2170  4888
2171  1688
2172  6604
2173  0537
2174  5888
2175  0877
2176  1888
2177  6558
      2200    PAGE


                    /TAPE 2
                    /ROUTINE TO RUN FROM END ZONE TO END ZONE
                    /AND DISPLAY THE CURRENT BLOCK NUMBER IN THE AC

2200  7300  DBLOCK, CLA CLL
2201  3233          DCA     DISBLK          /ZERO DISBLK
2202  1377          TAD     (3888
2203  1234          TAD     UNIT
2204  6774  IOT111, SDLC                    /LOAD CONTROL WITH UNIT REV GO READ
```

```
2285  7300          CLA CLL
2286  6771  DISLUP, SDSS                  /WAIT FOR SINGLE LINE FLAG
2287  5286          JMP     .-1
2288  7300          CLA CLL
2289  6777  IOT112, SDRD                  /READ DATA BUFFER
2290  3236          DCA     DISDAT        /AND SAVE
2291  6776  IOT113, SDRC                  /READ MARK TRACK
2292  0376          AND     (77
2293  1375          TAD     (-26
2294  7440          SZA                   /BLOCK NUMBER?
2295  5224          JMP     DISEND        /NO, CHECK FOR END ZONE
2296  1236          TAD     DISDAT        /YES, DISPLAY BLOCK NUMBER
2297  2233          ISZ     DISBLK
2298  5221          JMP     .-1
2299  5286          JMP     DISLUP
2300  1374  DISEND, TAD     (4
2301  7640          SZA CLA         /END ZONE?
2302  5286          JMP     DISLUP        /NO, LOOP
2303  6776  IOT114, SDRC                  /YES, EXTRACT DIRECTION BIT
2304  7006          RTL                   /AND COMPLEMENT
2305  7032          CML RTR
2306  5204          JMP     DISLUP+2      /GO LOAD INTO CONTROL
2307  0000  DISBLK, 0
2310  0000  UNIT,   0
2311  0000  DISTRK, 0
2312  0000  DISDAT, 0
              /ROUTINE TO ROCK DECTAPE UNIT 0
              /FOR A DISTANCE DETERMINED BY ACS

2313  7300  ROCK,   CLA CLL
2314  1373          TAD     (1000
2315  6774  IOT115, SDLC
2316  7604          LAS
2317  7040          CMA
2320  3001          DCA     1
2321  2000          ISZ     0
2322  5245          JMP     .-1
2323  2001          ISZ     1
2324  5245          JMP     .-3
2325  7000          NOP
2326  1377          TAD     (3000
2327  6774  IOT116, SDLC
2330  7604          LAS
2331  7040          CMA
2332  3001          DCA     1
2333  2000          ISZ     0
2334  5257          JMP     .-1
2335  2001          ISZ     1
2336  5257          JMP     .-3
2337  5240          JMP     ROCK+1
2340  0000  MESAGE, 0
2341  3315          DCA     MPNTR
2342  1715          TAD I   MPNTR
2343  7012          RTR
2344  7012          RTR
```



```
2345  7012          RTR
2346  0376          AND     (77
2347  7450          SNA
2350  5664          JMP I   MESAGE
2351  1372          TAD     (-40
2352  7510          SPA
2353  1371          TAD     (100
2354  1370          TAD     (240
2355  4031          JMS     TYPE
2356  1715          TAD I   MPNTR
2357  0376          AND     (77
2360  7450          SNA
2361  5664          JMP I   MESAGE
2362  1372          TAD     (-40
2363  7510          SPA
2364  1371          TAD     (100
2365  1370          TAD     (240
2366  4031          JMS     TYPE
2367  2315          ISZ     MPNTR
2370  5266          JMP     MESAGE+2
2371  0000  MPNTR,  0
2372  0000  OPRINT, 0
2373  3340          DCA     ONUMB
2374  1367          TAD     (-4
2375  3341          DCA     OCNT
2376  1340          TAD     ONUMB
2377  7004          RAL
2400  7004  OPLOOP, RAL
2401  7006          RTL
2402  3340          DCA     ONUMB
2403  1340          TAD     ONUMB
2404  0366          AND     (7
2405  1365          TAD     (260
2406  4031          JMS     TYPE
2407  1340          TAD     ONUMB
2410  2341          ISZ     OCNT
2411  5324          JMP     OPLOOP
2412  7200          CLA
2413  5716          JMP I   OPRINT
2414  0000  ONUMB,  0
2415  0000  OCNT,   0
2416  0260
2417  0007
2420  7774
2421  0240
2422  0100
2423  7740
2424  1000
2425  0004
2426  7752
2427  0077
2430  3000
      2488          PAGE

              /ROUTINE TO RUN FROM ENDZONE TO ENDZONE CHECKING THE MARK TRACK IN BLOCKS
```

```
2400  7300  BLKCHK, CLA CLL
2401  1377        TAD     (3000           /START TAPE MOVING BACKWARD
2402  6774  IOT117, SDLC
2403  4315        JMS     RD6MRK          /WAIT FOR WINDOW TO OPEN
2404  4307  END2,  JMS     RD1MRK          /READ BACK MARK TRACK
2405  1376        TAD     (-22
2406  7640        SZA CLA                 /END2ONE?
2407  5204        JMP     .+3             /NO
2410  6776  IOT118, SDRC                 /TURN AROUND
2411  7006        RTL
2412  7032        CML RTR
2413  6774  IOT119, SDLC
2414  4315        JMS     RD6MRK          /WAIT FOR WINDOW TO OPEN
2415  4307        JMS     RD1MRK          /READ MARK TRACK
2416  1375        TAD     (-26
2417  7650        SNA CLA                 /BLOCK NUMBER?
2420  5236        JMP     RVGARD          /YES, GO CHECK REVERSE GUARD
2421  5215        JMP     .-4             /NO, LOOK AGAIN
2422  4315  FWDEXP, JMS     RD6MRK          /READ MARK TRACK
2423  1374        TAD     (-25
2424  7440        SZA                     /EXPAND CODE?
2425  7402        HLT                     /NO, ERROR
2426  4315  BLKMRK, JMS     RD6MRK          /READ MARK TRACK
2427  1375        TAD     (-26
2430  7450        SNA                     /BLOCK NUMBER?
2431  5236        JMP     RVGARD          /YES, GO CHECK REVERSE GUARD
2432  7001        IAC                     /NO
2433  7440        SZA                     /EXPAND CODE?
2434  7402        HLT                     /NO, UNKNOWN
2435  5204        JMP     END2            /YES, EXPAND CODE, GO LOOK FOR END2ONE
2436  4315  RVGARD, JMS     RD6MRK          /GET MARK TRACK
2437  1373        TAD     (-32
2440  7440        SZA                     /REVERSE GUARD?
2441  7402        HLT                     /NO, ERROR
2442  1372        TAD     (-4             /SET UP
2443  3000        DCA     0               /FOR 4 MARKS
2444  4315        JMS     RD6MRK          /GET MARK TRACK
2445  1371  LOCK,  TAD     (-10
2446  7440        SZA                     /LOCK, REV  CHKSM, REV FINAL, REV PRE-FINAL?
2447  7402        HLT                     /NO, ERROR
2450  2000        ISZ     0
2451  5244        JMP     .-5
2452  1370  DATA,  TAD     (-122           /SET UP
2453  3000        DCA     0               /FOR 82 MARKS
2454  4315        JMS     RD6MRK          /GET MARK TRACK
2455  1367        TAD     (-78
2456  7440        SZA                     /DATA MARK?
2457  7402        HLT                     /NO, ERROR
2460  2000        ISZ     0
2461  5254        JMP     .-5
2462  1372  PREFIN, TAD     (-4             /SET UP
2463  3000        DCA     0               /FOR 4 MARKS
2464  4315        JMS     RD6MRK          /GET MARK TRACK
2465  1366        TAD     (-73
2466  7440        SZA                     /PREFINAL, FINAL, CHKSM, REVLOCK?
```



```
2467  7402        HLT                     /NO, ERROR
2470  2000        ISZ     0
2471  5264        JMP     .-5
2472  4315  GUARD,  JMS     RD6MRK          /GET MARK TRACK
2473  1365        TAD     (-51
2474  7440        SZA                     /GUARD?
2475  7402        HLT                     /NO, ERROR
2476  4315  REVBLK, JMS     RD6MRK          /GET MARK TRACK
2477  1364        TAD     (-45
2500  7440        SZA                     /REVERSE BLOCK NUMBER?
2501  7402        HLT                     /NO, ERROR
2502  4315  REVEXP, JMS     RD6MRK          /GET MARK TRACK
2503  1374        TAD     (-25
2504  7440        SZA                     /REVERSE EXPAND CODE?
2505  7402        HLT                     /NO, ERROR
2506  5222        JMP     FWDEXP
                  /READ 1 SHIFT OF MARK TRACK SUBROUTINE
2507  0000  RD1MRK, 0
2510  6771  IOT120, SDSS
2511  5310        JMP     .-1
2512  6776  IOT121, SDRC
2513  0363        AND     (77
2514  5707        JMP I   RD1MRK
                  /READ 6 SHIFTS OF MARK TRACK SUBROUTINE
2515  0000  RD6MRK, 0
2516  1362        TAD     (-6
2517  3307        DCA     RD1MRK
2520  6771  IOT122, SDSS
2521  5320        JMP     .-1
2522  6776  IOT123, SDRC
2523  2307        ISZ     RD1MRK
2524  5320        JMP     .-4
2525  0363        AND     (77
2526  5715        JMP I   RD6MRK
2562  7772
2563  0077
2564  7733
2565  7727
2566  7705
2567  7710
2570  7656
2571  7770
2572  7774
2573  7746
2574  7753
2575  7752
2576  7756
2577  3000
      2600        PAGE

                  /CHECK SELECT ERROR STATUS BIT AND ABILITY TO CLEAR "WRITE"
                  /UNIT 1 IS "OFF-LINE" OR NON-EXISTANT
                  /UNIT 0 IS "ON LINE" AND "WRITE LOCKED"
2600  7300  SELECT, CLA CLL
```

```
2601  1377          TAD     (MESS58
2602  3025          DCA     HEAD1
2603  1376          TAD     (MESS51
2604  3026          DCA     HEAD2
2605  6774  IOT124, SDLC
2606  6772  IOT125, SDST                    /IS TIMING ERROR SET?
2607  7410          SKP
2610  4775'         JMS     ERROR2          /YES, ERROR
2611  1374          TAD     (MESS52
2612  3026          DCA     HEAD2
2613  1373          TAD     (4000
2614  6774  IOT126, SDLC                    /SET UNIT BIT TO 1
2615  7200          CLA
2616  6776  IOT127, SDRC                    /READ STATUUS
2617  3021          DCA     IN              /SAVE
2620  7604          LAS
2621  7710          SPA CLA                 /LOOP?
2622  5200          JMP     SELECT          /YES
2623  1021          TAD     IN
2624  0372          AND     (100
2625  7650          SNA CLA                 /SELECT ERROR?
2626  4775'         JMS     ERROR2          /NO, ERROR
2627  7604          LAS
2630  7710          SPA CLA                 /LOOP?
2631  5200          JMP     SELECT          /YES
2632  1371          TAD     (MESS53
2633  3026          DCA     HEAD2
2634  1370  SELCT1, TAD     (4000
2635  6774  IOT130, SDLC                    /TRY TO SET "WRITE"
2636  7604          LAS
2637  7710          SPA CLA                 /LOOP?
2640  5234          JMP     SELCT1          /YES
2641  6776  IOT129, SDRC                    /READ STATUS
2642  3021          DCA     IN              /SAVE
2643  1021          TAD     IN
2644  0367          AND     (400
2645  7640          SZA CLA                 /WRITE SET?
2646  4775'         JMS     ERROR2          /YES, ERROR
2647  7604          LAS
2650  7710          SPA CLA                 /LOOP?
2651  5234          JMP     SELCT1          /YES
2652  1366          TAD     (MESS63
2653  3026          DCA     HEAD2
2654  6774  SELCT2, SDLC                    /SELECT UNIT 0
2655  7604          LAS
2656  7710          SPA CLA                 /LOOP?
2657  5254          JMP     SELCT2          /YES
2660  6776  IOT132, SDRC                    /READ STATUS
2661  3021          DCA     IN              /SAVE
2662  1021          TAD     IN
2663  0372          AND     (100
2664  7640          SZA CLA                 /SELECT ERROR?
2665  4775'         JMS     ERROR2          /YES
2666  7604          LAS
2667  7710          SPA CLA                 /LOOP?
```



```
2670  5254          JMP     SELCT2          /YES
2671  4046          JMS     LOOP1
2672  5200          JMP     SELECT
                    /CHECK WRITE LOCK OUT STATUS BIT AND ABILITY TO CLEAR "WRITE"
                    /UNIT 0 IS "WRITE-LOCKED"
2673  7300  WL0,    CLA CLL
2674  1365          TAD     (MESS54
2675  3025          DCA     HEAD1
2676  1364          TAD     (MESS55
2677  3026          DCA     HEAD2
2700  6774          SDLC
2701  6776  IOT131, SDRC                    /READ STATUS
2702  3021          DCA     IN              /SAVE
2703  7604          LAS
2704  7710          SPA CLA                 /LOOP?
2705  5273          JMP     WL0             /YES
2706  1021          TAD     IN
2707  0363          AND     (200
2710  7650          SNA CLA                 /WRITE LOCK OUT BIT SET?
2711  4775'         JMS     ERROR2          /NO, ERROR
2712  7604          LAS
2713  7710          SPA CLA                 /LOOP?
2714  5273          JMP     WL0             /YES
2715  1362          TAD     (MESS56
2716  3026          DCA     HEAD2
2717  1367  WL1,    TAD     (400
2720  6774  IOT132, SDLC                    /TRY TO SET "WRITE"
2721  7604          LAS
2722  7710          SPA CLA                 /LOOP?
2723  5317          JMP     WL1             /YES
2724  6776  IOT133, SDRC                    /READ STATUS
2725  3021          DCA     IN              /SAVE
2726  1021          TAD     IN
2727  0367          AND     (400
2730  7640          SZA CLA                 /WRITE SET?
2731  4775'         JMS     ERROR2                  /YES, ERROR
2732  7604          LAS
2733  7710          SPA CLA                 /LOOP?
2734  5317          JMP     WL1             /YES
2735  4046          JMS     LOOP1
2736  5273          JMP     WL2
2737  1361          TAD     (OK
2740  4760'         JMS     MESAGE
2741  4040          JMS     CRLF
2742  7402          HLT
2743  5342          JMP     .+1
2744  1713  OK,     TEXT    "OK"
2745  0000

2760  2264
2761  2744
2762  6767
2763  0200
2764  6744
2765  6730
```

```
2766  7107
2767  0400
2770  4400
2771  6706
2772  0100
2773  4000
2774  6044
2775  0337
2776  6053
2777  6040
      3000          PAGE

                    /TD8EA READ-WRITE AND SEARCH TEST PROGRAM
                    /COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.  01754

      7200    BUFF1=7200
      7400    BUFF2=7400

                    /ROUTINE TO CHECK DATA TRANSFERS ON TAPE

3000  7300    XFER,   CLA CLL
3001  3021            DCA     IN
3002  1377            TAD     (FILL0
3003  4224            JMS     WREAD           /FILL A BUFFER, THEN WRITE AND READ 0'S
3004  1376            TAD     (FILL1
3005  4224            JMS     WREAD           /FILL A BUFFER, THEN WRITE AND READ =1'S
3006  1375            TAD     (FILL25
3007  4224            JMS     WREAD           /FILL A BUFFER, THEN WRITE AND READ 2525
3010  1374            TAD     (FILPAT
3011  4224            JMS     WREAD           /FILL A BUFFER, THEN WRITE AND READ 2225,
                                              /5522,2555
3012  1373            TAD     (FILINC         /INCREMENT PATTERN
3013  4224            JMS     WREAD
3014  1372            TAD     (FILDEC         /DECREMENT PATTERN
3015  4224            JMS     WREAD
3016  1371            TAD     ($SPEC1         /0101
3017  4224            JMS     WREAD
3020  1370            TAD     ($SPEC2         /3434
3021  4224            JMS     WREAD
3022  4767'           JMS     PASCNT
3023  5202            JMP     XFER+2
                    /ROUTINE TO WRITE AND READ BACK AND COMPARE EVERY 100TH BLOCK ON TAPE
3024  0000    WREAD,  0
3025  3030            DCA     FILPNT
3026  3766'           DCA     SUNIT
3027  1345            TAD     (-200
3030  4430            JMS I   FILPNT          /FILL BUFF1 WITH DATA
3031  7200            BUFF1
3032  3027            DCA     BLK             /ZERO BLOCK NUMBER
3033  1364            TAD     (MES857
3034  3025            DCA     HEAD1
3035  1363    WREL1,  TAD     (MES858
3036  3026            DCA     HEAD2
3037  1027            TAD     BLK
```



```
3040  4762'           JMS     WRITE           /WRITE ONTO TAPE
3041  7200            BUFF1
3042  7600            -200
3043  1361            TAD     (MES859
3044  3026            DCA     HEAD2
3045  1027            TAD     BLK
3046  4760'           JMS     READ            /READ BACK INTO MEMORY
3047  7400            BUFF2
3050  7777            -201
3051  1345            TAD     (-200
3052  4757'           JMS     COMPAR          /COMPARE DATA
3053  7200            BUFF1
3054  7401            BUFF2+1
3055  1356            TAD     (MES860
3056  3026            DCA     HEAD2
3057  1027            TAD     BLK
3060  4755'           JMS     READR           /READ BACK BACKWARDS
3061  7400            BUFF2
3062  7777            -201
3063  1027            TAD     BLK             /BUMP BLOCK NUMBER

3064  1354            TAD     (100
3065  3027            DCA     BLK
3066  1027            TAD     BLK
3067  1353            TAD     (-2701
3070  7710            SPA CLA
3071  5235            JMP     WREL1
3072  1352            TAD     (2701
3073  3027            DCA     BLK             /SET BLOCK NUMBER TO 2701
3074  1351            TAD     (MES861
3075  3025            DCA     HEAD1
3076  1350    WREL2,  TAD     (MES862
3077  3026            DCA     HEAD2
3100  1027            TAD     BLK
3101  4747'           JMS     WRITER          /WRITE ONTO TAPE BACKWARDS
3102  7200            BUFF1
3103  7600            -200
3104  1356            TAD     (MES860
3105  3026            DCA     HEAD2
3106  1027            TAD     BLK
3107  4755'           JMS     READR           /READ BACK INTO MEMORY BACKWARDS
3110  7400            BUFF2
3111  7777            -201
3112  1345            TAD     (-200
3113  4757'           JMS     COMPAR          /COMPARE DATA
3114  7200            BUFF1
3115  7401            BUFF2+1
3116  1361            TAD     (MES859
3117  3026            DCA     HEAD2
3120  1027            TAD     BLK
3121  4760'           JMS     READ            /READ BACK FORWARD
3122  7400            BUFF2
3123  7777            -201
3124  1027            TAD     BLK             /BUMP BLOCK NUMBER
3125  1346            TAD     (-100
```

```
3126  3027          DCA     BLK
3127  1027          TAD     BLK
3130  7700          SMA CLA
3131  5276          JMP     WRBL2
3132  7604          LAS
3133  7010          RAR
3134  7630          SZL     CLA             /TWO UNITS?
3135  5624          JMP I   WREAD           /NO
3136  1766'         TAD     SUNIT           /YES, INCREMENT
3137  1345          TAD     (4000           /UNIT
3140  7450          SNA
3141  5624          JMP I   WREAD           /EXIT IF BACK TO ZERO
3142  3766'         DCA     SUNIT           /STORE BACK IF NO-ZERO
3143  5232          JMP     WREL1-3         /LOOP
3145  4000
3146  7700
3147  3671
3150  7075
3151  7042
3152  2701
3153  7077
3154  0100
3155  3714
3156  7050
3157  3200
3160  4600
3161  7037
3162  4472
3163  7025
3164  7012
3165  7600
3166  4471
3167  3477
3170  3461
3171  3443
3172  3421
3173  3400
3174  3276
3175  3260
3176  3243
3177  3227
      3200    PAGE


                /SUBROUTINE TO COMPARE TWO DATA BUFFERS, INDICATE AN ERROR
                /CALLING SEQUENCE:
                /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO COMPARE
                /       JMS     COMPAR          /CALL SUBROUTINE
                /       GADD                    /1ST ADDRESS OF GOOD DATA
                /       TADD                    /1ST ADDRESS OF TEST DATA
                /                               /RETURN HERE WHEN DONE
3200  0000      COMPAR, 0
3201  3224          DCA     CCNTR
3202  1600          TAD I   COMPAR
```



```
3203  3225          DCA     GPNTR
3204  2200          ISZ     COMPAR
3205  1600          TAD I   COMPAR
3206  3226          DCA     TPNTR
3207  2200          ISZ     COMPAR
3210  1377          TAD     (DATMES
3211  3776'         DCA     DATMD
3212  1625      COMLUP, TAD I   GPNTR
3213  7041          CIA
3214  1626          TAD I   TPNTR
3215  7640          SZA CLA
3216  4775'         JMS     DATERR
3217  2225          ISZ     GPNTR
3220  2226          ISZ     TPNTR
3221  2224          ISZ     CCNTR
3222  5212          JMP     COMLUP
3223  5600          JMP I   COMPAR
3224  0000      CCNTR,  0
3225  0000      GPNTR,  0
3226  0000      TPNTR,  0


                /SUBROUTINE TO FILL MEMORY WITH ZEROES
                /CALLING SEQUENCE:
                /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                /       JMS     FILL0           /CALL SUBROUTINE
                /       ADDR                    /1ST ADDRESS TO FILL

3227  0000      FILL0,  0
3230  3241          DCA     FILL0C
3231  1627          TAD I   FILL0
3232  3242          DCA     FILL0P
3233  2227          ISZ     FILL0
3234  3642          DCA I   FILL0P
3235  2242          ISZ     FILL0P
3236  2241          ISZ     FILL0C
3237  5234          JMP     .-3
3240  5627          JMP I   FILL0
3241  0000      FILL0C, 0
3242  0000      FILL0P, 0
                /SUBROUTINE TO FILL MEMORY WITH -1 (7777)
                /CALLING SEQUENCE:
                /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                /       JMS     FILL1           /CALL SUBROUTINE
                /       ADDR                    /1ST ADDRESS TO FILL

3243  0000      FILL1,  0
3244  3256          DCA     FILL1C
3245  1643          TAD I   FILL1
3246  3257          DCA     FILL1P
3247  2243          ISZ     FILL1
3250  7240          CLA CMA
3251  3657          DCA I   FILL1P
3252  2257          ISZ     FILL1P
```

```
3253  2256            ISZ     FILL1C
3254  5298            JMP     .-4
3255  5643            JMP I   FILL1
3256  0000    FILL1C, 0
3257  0000    FILL1P, 0
                      /SUBROUTINE TO FILL MEMORY WITH 2525
                      /CALLING SEQUENCE:
                      /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                      /       JMS     FILL25          /CALL SUBROUTINE
                      /       ADDR                    /1ST ADDRESS TO FILL

3260  0000    FILL25, 0
3261  3273            DCA     FILL2C
3262  1660            TAD I   FILL25
3263  3275            DCA     FILL2P
3264  2260            ISZ     FILL25
3265  1274            TAD     FILL2K
3266  3675            DCA I   FILL2P
3267  2275            ISZ     FILL2P
3270  2273            ISZ     FILL2C
3271  5265            JMP     .-4
3272  5660            JMP I   FILL25
3273  0000    FILL2C, 0
3274  2525    FILL2K, 2525
3275  0000    FILL2P, 0

                      /SUBROUTINE TO FILL MEMORY WITH 2225,5522,2555
                      /CALLING SEQUENCE:
                      /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL.
                      /       JMS     FILPAT          /CALL SUBROUTINE
                      /       ADDR                    /1ST ADDRESS TO FILL

3276  0000    FILPAT, 0
3277  3323            DCA     FILLC1
3300  1676            TAD I   FILPAT
3301  3321            DCA     FILLP1
3302  2276            ISZ     FILPAT
3303  1325    FILPL1, TAD     FILTP
3304  3322            DCA     FILLP2
3305  1331            TAD     FILTC
3306  3324            DCA     FILLC2
3307  1722    FILPL2, TAD I   FILLP2
3310  3721            DCA I   FILLP1
3311  2321            ISZ     FILLP1
3312  2323            ISZ     FILLC1
3313  7410            SKP
3314  5676            JMP I   FILPAT
3315  2322            ISZ     FILLP2
3316  2324            ISZ     FILLC2
3317  5307            JMP     FILPL2
3320  5303            JMP     FILPL1
3321  0000    FILLP1, 0
3322  0000    FILLP2, 0
3323  0000    FILLC1, 0
```



```
3324  0000    FILLC2, 0
3325  3326    FILTP,  .+1
3326  2225            2225
3327  5522            5522
3330  2555            2555
3331  7775    FILTC,  FILTP-FILTC-1
3375  4000
3376  4040
3377  4042
      3400    PAGE

                      /SUBROUTINE TO FILL MEMORY WITH AN INCREMENT PATTERN
                      /CALLING SEQUENCE:
                      /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                      /       JMS     FILINC          /CALL SUBROUTINE
                      /       ADDR                    /1ST ADDRESS TO FILL

3400  0000    FILINC, 0
3401  3416            DCA     FILICT
3402  1600            TAD I   FILINC
3403  3417            DCA     FILIPT
3404  2200            ISZ     FILINC
3405  3220            DCA     FILIDT
3406  1220            TAD     FILIDT
3407  3617            DCA I   FILIPT
3410  2220            ISZ     FILIDT
3411  7000            NOP
3412  2217            ISZ     FILIPT
3413  2216            ISZ     FILICT
3414  5206            JMP     .-6
3415  5600            JMP I   FILINC
3416  0000    FILICT, 0
3417  0000    FILIPT, 0
3420  0000    FILIDT, 0
                      /SUBROUTINE TO FILL MEMORY WITH A DECREMENT PATTERN
                      /CALLING SEQUENCE:
                      /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                      /       JMS     FILDEC          /CALL SUBROUTINE
                      /       ADDR                    /1ST ADDRESS TO FILL

3421  0000    FILDEC, 0
3422  3240            DCA     FILDCT
3423  1621            TAD I   FILDEC
3424  3241            DCA     FILDPT
3425  2221            ISZ     FILDEC
3426  3242            DCA     FILDDT
3427  1242            TAD     FILDDT
3430  3641            DCA I   FILDPT
3431  7040            CMA
3432  1242            TAD     FILDDT
3433  2241            ISZ     FILDPT
3434  2240            ISZ     FILDCT
3435  5226            JMP     .-9
3436  7200            CLA
```

```
3437  5621          JMP I   FILDEC
3440  0000    FILDCT, 0
3441  0000    FILDPT, 0
3442  0000    FILDDT, 0

                /SUBROUTINE TO FILL MEMORY WITH 6161
                /CALLING SEQUENCE:
                /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                /       JMS     SPEC1           /CALL SUBROUTINE
                /       ADDR                    /1ST ADDRESS TO FILL
                /
3443  0000    SPEC1,  0
3444  3256            DCA     SPICT
3445  1643            TAD I   SPEC1
3446  3257            DCA     SPIPT
3447  2243            ISZ     SPEC1
3450  1260            TAD     SPEC1D
3451  3657            DCA I   SPIPT
3452  2257            ISZ     SPIPT
3453  2256            ISZ     SPICT
3454  5250            JMP     .-4
3455  5643            JMP I   SPEC1
3456  0000    SPICT,  0
3457  0000    SPIPT,  0
3460  6161    SPEC1D, 6161
                /SUBROUTINE TO FILL MEMORY WITH 3434
                /CALLING SEQUENCE:
                /       TAD     (-N             /MINUS (2'S) NUMBER OF WORDS TO FILL
                /       JMS     SPEC2           /CALL SUBROUTINE
                /       ADDR                    /1ST ADDRESS TO FILL
                /
3461  0000    SPEC2,  0
3462  3274            DCA     SP2CT
3463  1661            TAD I   SPEC2
3464  3275            DCA     SP2PT
3465  2261            ISZ     SPEC2
3466  1276            TAD     SPEC2D
3467  3675            DCA I   SP2PT
3470  2275            ISZ     SP2PT
3471  2274            ISZ     SP2CT
3472  5266            JMP     .-4
3473  5661            JMP I   SPEC2
3474  0000    SP2CT,  0
3475  0000    SP2PT,  0
3476  3434    SPEC2D, 3434
3477  0000    PASCNT, 0
3500  4040            JMS     CRLF
3501  1377            TAD     (PASS
3502  4776'           JMS     MESAGE
3503  2021            ISZ     IN
3504  7000            NOP
3505  1021            TAD     IN
3506  4775'           JMS     OPRINT
3507  1374            TAD     (COMP
```



```
3510  4776'           JMS     MESAGE
3511  4040            JMS     CRLF
3512  5677            JMP I   PASCNT
3513  2001    PASS,   TEXT    "PASS "
3514  2323
3515  4000
3516  4003    COMP,   TEXT    " COMPLETE"
3517  1715
3520  2014
3521  0524
3522  0500
3524  3516
3525  2316
3526  2264
3527  3513
      3600    PAGE

                /REVERSE SEARCH SUBROUTINE

3600  0000    RSERCH, 0
3601  3270            DCA     RSLOOK
3602  1054            TAD     M10     /SET P A COUNT OF 10 TIMES
3603  3055            DCA     BLKTRY  /TO SEARCH FOR A BLOCK
3604  1377            TAD     (1000
3605  1776'           TAD     SUNIT
3606  6774    RSRCH0, SDLC
3607  6776    IOT134, SDRC
3610  0375            AND     (180
3611  7640            SZA CLA
3612  5563            JMP I   (SELERR
3613  4774'           JMS     RDQUAD
3614  4774'           JMS     RDQUAD
3615  6771    RSRCH1, SDSS
3616  7410            SKP
3617  6777    IOT135, SDRD
3620  6771    IOT136, SDSS
3621  5220            JMP     .-1
3622  6776    IOT138, SDRC
3623  0373            AND     (77
3624  1372            TAD     (-26    /BLOCK MARK
3625  7450            SNA
3626  5240            JMP     RSRCH2  /YES
3627  1371            TAD     (4      /END ZONE
3630  7640            SZA     CLA
3631  5215            JMP     RSRCH1  /NO,GO READ AGAIN
3632  6776    IOT13A, SDRC            /READ THE C,R,
3633  7006            RTL             /SET THE DIRECTION BIT IN LINK
3634  7032            CML     RTR     /COMPLEMENT IT FOR TURN AROUND
3635  2055            ISZ     BLKTRY  /INCREMENT BLOCK TRY COUNTER
3636  5206            JMP     RSRCH0
3637  5261            JMP     RSTURN+5        /COULDN'T FIND BLOCK AFTER 8 TRIES
3640  6776    RSRCH2, SDRC
3641  7006            RTL
```

```
3642  6777  IOT137, SDRD            /READ THE BLOCK NUMBER
3643  7041          CIA
3644  1278          TAD     RSLOOK
3645  7450          SNA
3646  5265          JMP     RLOC8D
3647  7041          CIA
3650  7420          SNL
3651  1371          TAD     (4
3652  7638          SZL CLA
3653  5215          JMP     RSRCH1
3654  6776  RSTURN, SDRC
3655  7006          RTL
3656  7032          CML RTR
3657  2055          ISZ     BLKTRY
3660  5206          JMP     RSRCH0
3661  7200          CLA
3662  1278          TAD     RSLOOK
3663  7402          HLT             /AC=THE BLOCK THAT IT WAS LOOKING FOR
3664  5263          JMP     .-1     /BUT FAILED TO FIND AFTER 10 TRIES;
3665  7638  RLOC8D, SZL CLA
3666  5215          JMP     RSRCH1
3667  5600          JMP I   RSEARCH
3670  0000  RSLOOK, 0
              /WRITE REVERSE SUBROUTINE

3671  0000  WRITER, 0
3672  3770' WRITE1, DCA     WCNT
3673  1671          TAD I   WRITER
3674  3767'         DCA     WADDR
3675  2271          ISZ     WRITER
3676  1671          TAD I   WRITER
3677  3766'         DCA     WCOUNT
3700  1271          TAD     WRITER
3701  7001          IAC
3702  3765'         DCA     WRITE
3703  4764'         JMS     CSUMRT          /CALCULATE THE CHECKSUM
3704  0025          25
3705  7177          BUFF1-1
3706  7600          -200
3707  4763'         JMS     SDCXOR
3710  4331          JMS     WRTLCK          /CHECK FOR WRITE LOCK OUT
3711  1770'         TAD     WCNT
3712  4200          JMS     RSEARCH
3713  5762'         JMP     WRITE1
              /READ REVERSE SUBROUTINE

3714  0000  READR,  0
3715  3761'         DCA     RCNT
3716  1714          TAD I   READR
3717  3760'         DCA     RADDR
3720  2314          ISZ     READR
3721  1714          TAD I   READR
3722  3757'         DCA     RCOUNT
3723  1364          TAD     READR
3724  7001          IAC
```



```
3725  3756'         DCA     READ
3726  1761'         TAD     RCNT
3727  4200          JMS     RSEARCH
3730  5755'         JMP     READ1

3731  0000  WRTLCK, 0             /ROUTINE TO CHECK FOR WRITE LOCKOUT
3732  1776'         TAD     SUNIT
3733  6774  IOT151, SDLC
3734  6776  IOT19A, SDRC
3735  0304          AND     (200
3736  7648          SZA     CLA
3737  5562          JMP I   [WRDERR
3740  5731          JMP I   WRTLCK

3741  0200
3742  4612
3743  4000
3744  4665
3745  4657
3746  4656
3747  4512
3750  4714
3751  4303
3752  4672
3753  4545
3754  4544
3755  4470
3756  0004
3757  7752
3760  0077
3761  4707
3762  0100
3763  4471
3764  1000
      4000    PAGE

              /DATA ERROR HANDLER

4000  0000  DATERR, 0
4001  7604          LAS
4002  0377          AND     (400
4003  7648          SZA CLA
4004  5233          JMP     DA?HL?+3
4005  1240          TAD     DA?HD
4006  7650          SNA CLA
4007  5226          JMP     DA?NUM
4010  4200          JMS     HEADTP
4011  1240          TAD     DA?HD
4012  4776'         JMS     MESAGE
4013  3240          DCA     DA?HD
4014  4040          JMS     CRLF
4015  1375          TAD     (FORM?1
4016  4776'         JMS     MESAGE
```

```
4817  4848          JMS     CRLF
4820  1774'   DATNUM, TAD     GPNTR
4821  3241          DCA     DATPNT
4822  1641          TAD I   DATPNT
4823  4773'         JMS     OPRINT
4824  1372          TAD     (240
4825  4831          JMS     TYPE
4826  1771'         TAD     TPNTR
4827  3241          DCA     DATPNT
4830  1641          TAD I   DATPNT
4831  4773'         JMS     OPRINT
4832  4848          JMS     CRLF
4833  7684          LAS
4834  0378          AND     (200
4835  7650          SNA CLA
4836  7402    DATHLT, HLT
4837  5600          JMP I   DATERR
4840  0000    DATHD,  0
4841  0000    DATPNT, 0
4842  0401    DATMES, TEXT    "DATA ERROR"
4843  2401
4844  4005
4845  2222
4846  1722
4847  0000

                      /SUBROUTINE TO TYPE OUT HEADER FOR DATA TESTS
4850  0000    HEADTP, 0
4851  4848          JMS     CRLF
4852  1367          TAD     (UMESS
4853  4776'         JMS     MESAGE
4854  1372          TAD     (240
4855  4831          JMS     TYPE
4856  6776    IOT139, SDRC
4857  7710          SPA CLA
4860  7001          IAC
4861  1366          TAD     (200
4862  4831          JMS     TYPE
4863  4848          JMS     CRLF
4864  1365          TAD     (BMESS
4865  4776'         JMS     MESAGE
4866  1372          TAD     (240
4867  4831          JMS     TYPE
4870  1827          TAD     BLK
4871  4773'         JMS     OPRINT
4872  4848          JMS     CRLF
4873  1825          TAD     HEAD1
4874  4776'         JMS     MESAGE
4875  4848          JMS     CRLF
4876  1826          TAD     HEAD2
4877  4776'         JMS     MESAGE
4100  4848          JMS     CRLF
4101  5650          JMP I   HEADTP
4102  2516    UMESS,  TEXT    "UNIT"
4103  1124
```



```
4104  0000
4105  0214    BMESS,  TEXT    "BLOCK"
4106  1703
4107  1300

                      /CHECKSUM ERROR HANDLER

4110  0000    CHKERR, 0
4111  3331          DCA     CHKDAT
4112  6776    IOT140, SDRC                    /STOP TAPE
4113  0364          AND     (4000
4114  6774    IOT141, SDLC
4115  4250          JMS     HEADTP
4116  1363          TAD     (CHKMES
4117  4776'         JMS     MESAGE
4120  4848          JMS     CRLF
4121  7684          LAS
4122  0378          AND     (200
4123  7640          SZA CLA
4124  5710          JMP I   CHKERR
4125  1331          TAD     CHKDAT
4126  7402    CHKHLT, HLT
4127  7200          CLA
4130  5710          JMP I   CHKERR
4131  0000    CHKDAT, 0
4132  0316    CHKMES, TEXT    "CHECKSUM ERROR"
4133  0503
4134  1323
4135  2515
4136  4005
4137  2222
4140  1722
4141  0000

4163  4132
4164  4000
4165  4105
4166  0260
4167  4102
4170  0200
4171  3226
4172  0240
4173  2316
4174  3225
4175  0532
4176  2264
4177  0400
      4200    PAGE

                      /WRITE LOCK OUT ERROR

4200  4777'   WRDERR, JMS     HEADTP
4201  6776    IOT142, SDRC                    /STOP TAPE
```

```
4282  0376          AND     (4000
4283  6774  IOT143, SDLC
4284  1375          TAD     (WRDMES
4285  4774'         JMS     MESAGE
4286  4040          JMS     CRLF
4287  7604          LAS
4210  0373          AND     (200
4211  7650          SNA CLA
4212  7402  WRDHLT, HLT
4213  5772'         JMP     WRSL1+2

4214  2516  WRDMES, TEXT    "UNIT WRITE LOCKED"
4215  1124
4216  4027
4217  2211
4220  2405
4221  4014
4222  1703
4223  1305
4224  0400

                    /SELECT ERROR HANDLER

4225  4777'  SELERR, JMS     HEADTP
4226  1371          TAD     (SELMES
4227  4774'         JMS     MESAGE
4230  4040          JMS     CRLF
4231  7604          LAS
4232  0373          AND     (200
4233  7650          SNA CLA
4234  7402  SELHLT, HLT
4235  5772'         JMP     WRSL1+2

4236  2305  SELMES, TEXT    "SELECT ERROR"
4237  1405
4240  0324
4241  4005
4242  2222
4243  1722
4244  0000

                    /TIMING ERROR HANDLER

4245  0000  TYMERR, 0
4246  6776  IOT144, SDRC                    /STOP TAPE
4247  0376          AND     (4000
4250  6774  IOT145, SDLC
4251  4777'         JMS     HEADTP
4252  1370          TAD     (TYMMES
4253  4774'         JMS     MESAGE
4254  4040          JMS     CRLF
4255  7604          LAS
4256  0373          AND     (200
```



```
4257  7650          SNA CLA
4260  7402  TYMHLT, HLT
4261  5767'         JMP     WREAD+3
4262  2411  TYMMES, TEXT    "TIMING ERROR"
4263  1511
4264  1607
4265  4005
4266  2222
4267  1722
4270  0000

                    /SUBROUTINE TO CLEAR WRITE AFTER QUAD LINE FLAG

4271  0000  CLRWT,  0
4272  6773  IOT146, SDSQ                    /WAIT FOR QUAD LINE FLAG
4273  5272          JMP     .-1
4274  6772  IOT154, SDST                    /TIMING ERROR
4275  7610          SKP     CLA             /NO
4276  4161          JMS     I TYMERR        /YES
4277  6776  IOT147, SDRC                    /READ THE COMMAND REGISTER
4300  0346          AND     (7000           /MASK OFF WRITE BIT
4301  6774  IOT170, SDLC                    /LOAD THE COMMAND REGISTER
4302  5671          JMP I   CLRWT           /EXIT

4303  0000  CSUMRT, 0
4304  1703          TAD   I CSUMRT
4305  3765'         DCA     CHKSUM
4306  2303          ISZ     CSUMRT
4307  1703          TAD   I CSUMRT
4310  3017          DCA     AUTO
4311  2303          ISZ     CSUMRT
4312  1703          TAD   I CSUMRT
4313  3322          DCA     XXX
4314  2303          ISZ     CSUMRT
4315  1417          TAD   I AUTO
4316  4764'         JMS     SBGXOR
4317  2322          ISZ     XXX
4320  5315          JMP     .-2
4321  5703          JMP   I CSUMRT
4322  0000  XXX,    0

4323  0000  CHKCHK, 0
4324  4303          JMS     CSUMRT
4325  0000          0
4326  7377          BUFF2-1
4327  7575          -203
4330  1765'         TAD     CHKSUM
4331  7040          CMA
4332  0363          AND     (77
4333  7440          SZA
4334  4560          JMS   I (CHKERR         /CHECK SUM ERROR
4335  5723          JMP   I CHKCHK          /RETURN

4363  0077
4364  4714
```

```
4365  4744
4366  7888
4367  3827
4370  4262
4371  4236
4372  3937
4373  0288
4374  2264
4375  4214
4376  4888
4377  4898
      4400    PAGE

              /TD8-EA READ-WRITE-AND-SEARCH SUBROUTINES
              /COPYRIGHT 1971, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS. 01754
              /DECTAPE COMMANDS

      6771    SDSS=6771               /SKIP ON SINGLE LINE FLAG
      6772    SDST=6772               /SKIP ON TIMING ERROR
      6773    SDSQ=6773               /SKIP ON QUADRUPLE LINE FLAG
      6774    SDLC=6774               /LOAD COMMAND REGISTER
      6775    SDLD=6775               /LOAD DATA REGISTER, CLEAR FLAGS
      6776    SDRC=6776               /READ COMMAND REGISTER AND MARK TRACK, CLEAR FLAG
      6777    SDRD=6777               /READ DATA REGISTER, CLEAR FLAGS


              /SEARCH SUBROUTINE
              /SUBROUTINE IS ENTERED WITH THE NUMBER OF THE DESIRED BLOCK IN THE AC
              /PROGRAM WILL EXIT WITH TAPE MOVING IN THE FORWARD DIRECTION
              /UNIT BIT IS IN SUNIT, BIT 8, BITS 1 TO 11 ARE 0

4400  0000    SEARCH, 0
4401  3270            DCA     SLOOK   /SAVE BLOCK NUMBER
4402  1894            TAD     M10     /SET UP A COUNT OF 10
4403  3895            DCA     BLKTRY  /TO SEARCH FOR A BLOCK,
4404  1377            TAD     (3000   / PUT IN MOTION BACKWARD
4405  1271            TAD     SUNIT
4406  6774    SRCH0,  SDLC            /LOAD CONTROL WITH UNIT, REV, GO, READ
4407  6776    IOT146, SDRC            /READ STATUS
4410  0376            AND     (100
4411  7648            SZA CLA         /SELECT ERROR?
4412  5563            JMP I   [SELERR /YES
4413  4775'           JMS     RDQUAD  /DELAY TO ASSURE
4414  4775'           JMS     RDQUAD  /MARK WINDOW OPEN
4415  6771    SRCH1,  SDSS            /SINGLE LINE FLAG
4416  7410            SKP             /NO
4417  6777    IOT147, SDRD            /YES
4420  6771    IOT148, SDSS            /SKIP ON SINGLE LINE FLAG
4421  5220            JMP     .-[
4422  6776    IOT149, SDRC            /READ MARK TRACK AND COMMAND REGISTER
4423  0374            AND     (77     /MASK TO MARK TRACK BITS
4424  1373            TAD     (-26    /BLOCK MARK ?
4425  7490            SNA
4426  5240            JMP     SRCH2   /YES,GO READ THE BLOCK NUMBER
4427  1372            TAD     (4      /END ZONE ?
```



```
4430  7640            SZA     CLA
4431  5215            JMP     SRCH1   /NO,GO GET NEXT WORD
4432  6776    IOT14A, SDRC            /READ THE COMMAND REG,
4433  7006            RTL
4434  7032            CML     RTR     /TURN THE TAPE AROUND
4435  2895            ISZ     BLKTRY  /8 TRIES ?
4436  5206            JMP     SRCH0   /NO,TRY AGAIN
4437  5261            JMP     BADBLK  /YES,CAN NOT FIND BLOCK
4440  6776    SRCH2,  SDRC            /READ COMMAND REGISTER
4441  7006            RTL             /MOVE DIRECTION BIT INTO THE LINK
4442  6777    IOT150, SDRD            /GET BLOCK NUMBER FOUND
4443  7041            CIA
4444  1270            TAD     SLOOK   /COMBINE WITH BLOCK LOOKED FOR
4445  7450            SNA             /CURRENT BLOCK?
4446  5265            JMP     LOCSED  /YES, CHECK DIRECTION
4447  7041            CIA             /NO, TAKE 2'S COMPLEMENT
4450  7420            SNL             /LINK IS 1 IF BACKWARD AND NOT AT OR LOWER THAN BLOCK
4451  1371            TAD     (2      /ADD TWO TO ENABLE TURN AROUND
4452  7638            SZL CLA         /TURN AROUND (3 BEYOND)?
4453  5215            JMP     SRCH1   /NO, DON'T TURN AROUND

4454  6776    IOT15B, SDRC            /READ THE COMMAND REGISTER
4455  7006            RTL             /MOVE THE DIRECTION BIT INTO LINK
4456  7032            CML     RTR     /COMPLEMENT THE DIRECTION BIT
4457  2895            ISZ     BLKTRY  /8 TRIES ?
4460  5206            JMP     SRCH0   /NO , GO SEARCH AGAIN
4461  7200    BADBLK, CLA
4462  1270            TAD     SLOOK
4463  7402            HLT             /AC=THE BLOCK BEING SEARCHED FOR BUT FAILED
4464  5263            JMP     .-[     /TO FIND AFTER 8 TRIES
4465  7620    LOCSED, SNL CLA         /FOUND BLOCK FORWARD?
4466  5215            JMP     SRCH1   /NO
4467  5688            JMP I   SEARCH  /YES, EXIT
4470  0000    SLOOK,  0               /BLOCK NUMBER LOOKED FOR
4471  0000    SUNIT,  0               /CURRENT UNIT

              /WRITE SUBROUTINE
              /CALLING SEQUENCE:
              /       TAD     (BLKNO  /FIRST BLOCK TO BE WRITTEN INTO
              /       JMS     WRITE   /CALL SUBROUTINE
              /       ADDRESS         /ADDRESS OF FIRST DATA WORD
              /       -N              /MINUS (2'S) NUMBER OF WORDS TO TRANSFER
              /                       /RETURN HERE
              /128 WORDS PER BLOCK WILL BE WRITTEN FROM MEMORY

4472  0000    WRITE,  0
4473  3278            DCA     WCNT    /SAVE BLOCK NUMBER
4474  1672            TAD I   WRITE
4475  3344            DCA     WADDR   /SAVE ADDRESS
4476  2272            ISZ     WRITE
4477  1672            TAD I   WRITE
4500  3345            DCA     WCOUNT  /SAVE WORD COUNT
4501  2272            ISZ     WRITE
4502  4778'           JMS     CSUMRT
4503  0025            25
```

```
4384  7177          BUFF1-1
4385  7000          -200
4386  4767'         JMS     SBCXOR
4387  4766'         JMS     WRTLCK  /CHECK FOR WRITE LOCKOUT
4510  1278          TAD     WCNT
4511  4200          JMS     SEARCH  /FIND BLOCK
4512  4765'  WRITE1, JMS     REVGRD  /WAIT FOR REVERSE GUARD
4513  4775'         JMS     RDQUAD  /DELAY TWO-THIRDS THRU LOCK
4514  6776   IOT192, SDRC
4515  1344          TAD     (400
4516  6774   IOT193, SDLC            /LOAD CONTROL WITH UNIT, FWD, GO, WRITE
4517  1343          TAD     (25
4520  4762'         JMS     WRQUAD  /WRITE REVERSE CHECKSUM
4521  1744   WRITE2, TAD I   WADDR   /GET THE DATA WORD
4522  2344          ISZ     WADDR   /INCREMENT ADDRESS
4523  7000          NOP             /SAFETY NOP
4524  4762'         JMS     WRQUAD  /WRITE DATA WORD ON TAPE
4525  2345          ISZ     WCOUNT  /WORD 128?
4526  5321          JMP     WRITE2  /NO
4527  4762'         JMS     WRQUAD  /YES WRITE A 0 (WORD 129)
4530  1761'         TAD     CHKSUM
4531  7040          CMA
4532  0374          AND     (77
4533  7100          RTL CLL
4534  7006          RTL
4535  7006          RTL
4536  4762'         JMS     WRQUAD  /WRITE CHECKSUM
4537  4760'         JMS     CLRWT   /WAIT FOR CHECKSUM TO BE WRITTEN,CLEAR "WRITE"
4540  6776   IOT195, SDRC
4541  0387          AND     (4000
4542  6774   IOT196, SDLC            /STOP TAPE
4543  5672          JMP I   WRITE
      4476  WCNT=BLOCK              /BLOCK NUMBER, ALSO BLOCK DATA COUNTER
4544  0000  WADDR,  0               /WORD ADDRESS
4545  0000  WCOUNT, 0               /WORD COUNT

4547  4000
4548  4271
4561  4744
4562  4701
4563  0025
4564  0400
4565  4641
4566  3731
4567  4714
4570  4303
4571  0002
4572  0004
4573  7752
4574  0077
4575  4707
4576  0100
4577  3000
      4600  PAGE
```



```
                /READ SUBROUTINE
                /CALLING SEQUENCE:
                /       TAD     (BLKNO  /FIRST BLOCK TO BE READ FROM
                /       JMS     READ    /CALL SUBROUTINE
                /       ADDRESS         /ADDRESS FOR FIRST DATA WORD
                /       -N              /MINUS (2'S) NUMBER OF WORDS TO TRANSFER
                /                       /RETURN HERE
                /128 WORDS PER BLOCK WILL BE READ INTO MEMORY

4600  0000  READ,   0
4601  3256          DCA     RCNT    /SAVE BLOCK NUMBER
4602  1600          TAD I   READ
4603  3257          DCA     RADDR   /SAVE ADDRESS
4604  2200          ISZ     READ
4605  1600          TAD I   READ
4606  3260          DCA     RCOUNT  /SAVE WORD COUNT
4607  2200          ISZ     READ
4610  1256          TAD     RCNT
4611  4777'         JMS     SEARCH  /FIND BLOCK
4612  6771   READ1,  SDSS            /WAIT FOR REVERSE GUARD
4613  5212          JMP     .-1
4614  6776   IOT19A, SDRC            /READ THE MARK TRACK
4615  0376          AND     (77
4616  1375          TAD     (-32
4617  7490          SNA             /REVERSE GUARD
4620  5225          JMP     .+5     /YES, EXIT
4621  1374          TAD     (10     /NO
4622  7640          SZA     CLA     /END ZONE ?
4623  5212          JMP     READ1   /NO
4624  5374          JMP     IOT162  /YES STOP TAPE
4625  4307          JMS     RDQUAD
4626  4307          JMS     RDQUAD  /WAIT FOR
4627  4307          JMS     RDQUAD  /REVERSE CHECKSUM
4630  0376          AND     (77     / MASK
4631  7410          SKP             /STORE THE WORD
4632  4307   READ2,  JMS     RDQUAD  /GET DATA WORD
4633  3697          DCA I   RADDR
4634  2257          ISZ     RADDR
4635  7000          NOP             /SAFETY NOP
4636  2260          ISZ     RCOUNT  /128 DATA WORDS?
4637  5232          JMP     READ2   /NO
4640  4307          JMS     RDQUAD  /YES, GET WORD 129
4641  3697          DCA I   RADDR   /STORE IT
4642  2257          ISZ     RADDR
4643  4307          JMS     RDQUAD  /GET FORWARD CHECKSUM

4644  0373          AND     (7700
4645  3697          DCA I   RADDR
4646  6772   IOT197, SDST
4647  7410          SKP
4650  4561          JMS I   (TYMERR         /TIMING ERROR
4651  6776   IOT198, SDRC
4652  0372          AND     (4000
```

```
4653  6774  IOT159, SDLC            /STOP TAPE
4654  4771' JMS     CHKCHK  /CALCULATE AND CHECK CHECK SUM
4655  5600  JMP I   READ
4656  0000  RCNT,   0               /BLOCK NUMBER, ALSO BLOCK DATA COUNTER
4657  0000  RADDR,  0               /WORD ADDRESS
4660  0000  RCOUNT, 0               /WORD COUNT

      /WAIT FOR REVERSE GUARD SUBROUTINE
4661  0000  REVGRD, 0
4662  6771  IOT160, SDSS            /WAIT FOR MARK TRACK CHANGE
4663  5262  JMP     .-1
4664  6776  IOT161, SDRC            /READ MARK TRACK
4665  0376  AND     (77
4666  1375  TAD     (-32
4667  7498  SNA                     /REVERSE GUARD?
4670  5661  JMP I   REVGRD  /YES, EXIT
4671  1374  TAD     (10      /NO
4672  7048  SZA CLA                 /END ZONE?
4673  5262  JMP     REVGRD+1        /NO
4674  6776  IOT162, SDRC            /YES, STOP TAPE
4675  0372  AND     (4000
4676  6774  IOT163, SDLC
4677  7402  HLT                     /NON-RECOVERABLE ERROR, PROGRAM
4700  5277  JMP     .-1     /FOUND ENDZONE WHILE LOOKING FOR REV GRD
                            /BLOCK PROBABLY ABOVE 2771
      /WRITE A "QUAD WORD" (12 BIT WORD) SUBROUTINE
4701  0000  WRQUAD, 0
4702  6773  IOT164, SDSQ            /WAIT FOR NEXT QUAD FLAG
4703  5302  JMP     .-1
4704  6775  IOT165, SDLD            /LOAD DATA BUFFER TO WRITE ON TAPE
4705  7600  M0200A, CLA-400        /CLEAR AC
4706  5701  JMP I   WRQUAD  /EXIT


      /READ A "QUAD WORD" (12 BIT WORD) SUBROUTINE
4707  0000  RDQUAD, 0
4710  6773  IOT166, SDSQ            /WAIT FOR QUAD FLAG
4711  5310  JMP     .-1
4712  6777  IOT167, SDRQ            /READ DATA BUFFER, CLEAR FLAG
4713  5707  JMP I   RDQUAD

      /$IXBIT COMPLEMENT XOR SUBROUTINE
      /SUBROUTINE IS ENTERED WITH DATA WORD TO BE XORED IN AC
      /TWO 6-BIT COMPLEMENT XORS WILL TAKE PLACE TO LOCATION CHKSUM
      /WITH THE RESULT IN CHKSUM

4714  0000  SBCXOR, 0
4715  7040  CMA                     /COMPLEMENT WORD
4716  3345  DCA     SBWORD  /AND SAVE
4717  1345  TAD     SBWORD
4720  0344  AND     CHKSUM
4721  7041  CIA
4722  7104  CLL RAL
4723  1345  TAD     SBWORD
4724  1344  TAD     CHKSUM
```



```
4725  3344          DCA     CHKSUM
4726  1345          TAD     SBWORD
4727  7112          RTR CLL; RTR; RTR
4730  7012
4731  7012
4732  3345          DCA     SBWORD
4733  1345          TAD     SBWORD
4734  0344          AND     CHKSUM
4735  7041          CIA
4736  7104          CLL RAL
4737  1345          TAD     SBWORD
4740  1344          TAD     CHKSUM
4741  0376          AND     (77
4742  3344          DCA     CHKSUM
4743  5714          JMP I   SBCXOR
4744  0000  CHKSUM, 0
      4745  SBWORD=.,


4771  4323
4772  4000
4773  7700
4774  0010
4775  7746
4776  0077
4777  4400
      5000  PAGE

      /MESSAGES
5000  1417  MESS1,  TEXT    "LOAD AND READ DATA REGISTER ERROR"
5001  0104
5002  4001
5003  1604
5004  4022
5005  0501
5006  0440
5007  0401
5010  2401
5011  4022
5012  0507
5013  1123
5014  2405
5015  2240
5016  0522
5017  2217
5020  2200
5021  1417  MESS2,  TEXT    "LOAD AND READ COMMAND REGISTER ERROR"
5022  0104
5023  4001
5024  1604
5025  4022
5026  0501
5027  0440
```

```
5838  0317
5831  1515
5832  0116
5833  0440
5834  2205
5835  0711
5836  2324
5837  0522
5840  4005
5841  2222
5842  1722
5843  0000
5844  1116    MESS3,  TEXT    "INITIALIZE TEST"
5845  1124
5846  1101
5847  1411
5850  3205
5851  4024
5852  0523
5853  2400
5854  0301    MESS4,  TEXT    "CAF DID NOT CLEAR COMMAND REGISTER"
5855  0640
5856  0411
5857  0440
5860  1617
5861  2440
5862  0314
5863  0501
5864  2240
5865  0317
5866  1515
5867  0116
5870  0440
5871  2205
5872  0711
5873  2324
5874  0522
5875  0000
5876  0310    MESS5,  TEXT    "CHECK SOLC, SOLD, SORC, SORD AND AC CLEAR"
5877  0503
5100  1340
5101  2304
5102  1403
5103  5440
5104  2304
5105  1404
5106  5440
5107  2304
5110  2203
5111  5440
5112  2304
5113  2204
5114  4001
5115  1604
5116  4001
```



```
5117  0340
5120  0314
5121  0501
5122  2200
5123  2304    MESS6,  TEXT    "SOLC DID NOT CLEAR AC"
5124  1403
5125  4004
5126  1104
5127  4016
5130  1724
5131  4003
5132  1405
5133  0122
5134  4001
5135  0300
5136  2304    MESS7,  TEXT    "SORC DID NOT CLEAR AC"
5137  2203
5140  4004
5141  1104
5142  4016
5143  1724
5144  4003
5145  1405
5146  0122
5147  4001
5150  0300
5151  2304    MESS8,  TEXT    "SOLD CLEARED AC"
5152  1404
5153  4003
5154  1405
5155  0122
5156  0504
5157  4001
5160  0300
5161  2304    MESS9,  TEXT    "SORD DID NOT CLEAR AC"
5162  2204
5163  4004
5164  1104
5165  4016
5166  1724
5167  4003
5170  1405
5171  0122
5172  4001
5173  0300
5174  2311    MESS10, TEXT    "SINGLE LINE FLAG SKIP INSTRUCTION AND LOGIC"
5175  1607
5176  1405
5177  4014
5200  1116
5201  0540
5202  0614
5203  0107
5204  4023
5205  1311
```

```
9206  2040
9207  1116
9210  2324
9211  2225
9212  0324
9213  1117
9214  1640
9215  0116
9216  0440
9217  1417
9220  0711
9221  0380
9222  2311    MES$11, TEXT    "SINGLE LINE FLAG ALWAYS SET OR SDSS ALWAYS SKIPS"
9223  1607
9224  1405
9225  4014
9226  1116
9227  0540
9230  0614
9231  0107
9232  4001
9233  1427
9234  0131
9235  2340
9236  2305
9237  2440
9240  1722
9241  4023
9242  0423
9243  2340
9244  0114
9245  2701
9246  3125
9247  4023
9250  1311
9251  2023
9252  0000
9253  2311    MES$12, TEXT    "SINGLE LINE FLAG DOES NOT SET OR SDSS DOES NOT SKIP"
9254  1607
9255  1405
9256  4014
9257  1116
9260  0540
9261  0614
9262  0107
9263  4004
9264  1705
9265  2340
9266  1617
9267  2440
9270  2305
9271  2440
9272  1722
9273  4023
9274  0423
```



```
9275  2340
9276  0417
9277  0523
9300  4016
9301  1724
9302  4023
9303  1311
9304  2000
9305  2311    MES$13, TEXT    "SINGLE LINE FLAG CLEARED BY SDSS"
9306  1607
9307  1405
9310  4014
9311  1116
9312  0540
9313  0614
9314  0107
9315  4003
9316  1405
9317  0122
9320  0504
9321  4002
9322  3140
9323  2304
9324  2323
9325  0000
9326  2311    MES$14, TEXT    "SINGLE LINE FLAG NOT CLEARED BY CAF"
9327  1607
9330  1405
9331  4014
9332  1116
9333  0540
9334  0614
9335  0107
9336  4016
9337  1724
9340  4003
9341  1405
9342  0122
9343  0504
9344  4002
9345  3140
9346  0301
9347  0400
9350  2311    MES$15, TEXT    "SINGLE LINE FLAG NOT CLEARED BY $DLD"
9351  1607
9352  1405
9353  4014
9354  1116
9355  0540
9356  0614
9357  0107
9360  4016
9361  1724
9362  4003
9363  1405
```

```
5364  0122
5365  0584
5366  4002
5367  3140
5370  2304
5371  1404
5372  0000
5373  2311    MES$16, TEXT    "SINGLE LINE FLAG NOT CLEARED BY $DRC"
5374  1607
5375  1405
5376  4014
5377  1116
5400  0540
5401  0614
5402  0107
5403  4016
5404  1724
5405  4003
5406  1405
5407  0122
5410  0584
5411  4002
5412  3140
5413  2304
5414  2203
5415  0000
5416  2311    MES$17, TEXT    "SINGLE LINE FLAG NOT CLEARED BY $DRD"
5417  1607
5420  1405
5421  4014
5422  1116
5423  0540
5424  0614
5425  0107
5426  4016
5427  1724
5430  4003
5431  1405
5432  0122
5433  0584
5434  4002
5435  3140
5436  2304
5437  2204
5440  0000
5441  2311    MES$18, TEXT    "SINGLE LINE FLAG CLEARED BY $DST, $DSS, OR $DLC"
5442  1607
5443  1405
5444  4014
5445  1116
5446  0540
5447  0614
5450  0107
5451  4003
5452  1405
```



```
5453  0122
5454  0584
5455  4002
5456  3140
5457  2304
5460  2304
5461  5440
5462  2304
5463  2301
5464  5440
5465  1722
5466  4003
5467  0414
5470  0300
5471  2129    MES$19, TEXT    "QUAD LINE FLAG SKIP INSTRUCTION AND LOGIC"
5472  0104
5473  4014
5474  1116
5475  0540
5476  0614
5477  0107
5500  4003
5501  1311
5502  2040
5503  1116
5504  2304
5505  2225
5506  0304
5507  1117
5510  1040
5511  0116
5512  0440
5513  1417
5514  0711
5515  0300
5516  2129    MES$20, TEXT    "QUAD LINE FLAG ALWAYS SET OR $DSQ ALWAYS SKIPS"
5517  0104
5520  4014
5521  1116
5522  0540
5523  0614
5524  0107
5525  4001
5526  1427
5527  0131
5530  2340
5531  2309
5532  2440
5533  1722
5534  4003
5535  0403
5536  2140
5537  0114
5540  2701
5541  3103
```

```
5542  4023
5543  1311
5544  2823
5545  0000
5546  2125    MESS21, TEXT    "QUAD LINE FLAG SET TOO SOON"
5547  0104
5550  4014
5551  1116
5552  0540
5553  0614
5554  0107
5555  4023
5556  0524
5557  4024
5560  1717
5561  4023
5562  1717
5563  1600
5564  2125    MESS22, TEXT    "QUAD LINE FLAG DOES NOT SET OR SDSQ DOES NOT SKIP"
5565  0104
5566  4014
5567  1116
5570  0540
5571  0614
5572  0107
5573  4004
5574  1705
5575  2340
5576  1617
5577  2440
5600  2305
5601  2440
5602  1722
5603  4023
5604  0423
5605  2140
5606  0417
5607  0523
5610  4016
5611  1724
5612  4023
5613  1311
5614  2000
5615  2125    MESS23, TEXT    "QUAD LINE FLAG CLEARED BY SDSQ"
5616  0104
5617  4014
5620  1116
5621  0540
5622  0614
5623  0107
5624  4003
5625  1405
5626  0122
5627  0504
5630  4002
```



```
5631  3140
5632  2304
5633  2321
5634  0000
5635  2125    MESS24, TEXT    "QUAD LINE FLAG NOT CLEARED BY CAF"
5636  0104
5637  4014
5640  1116
5641  0540
5642  0614
5643  0107
5644  4016
5645  1724
5646  4003
5647  1405
5650  0122
5651  0504
5652  4002
5653  3140
5654  0301
5655  0600
5656  2125    MESS25, TEXT    "QUAD LINE FLAG NOT CLEARED BY SDLD"
5657  0104
5660  4014
5661  1116
5662  0540
5663  0614
5664  0107
5665  4016
5666  1724
5667  4003
5670  1405
5671  0122
5672  0504
5673  4002
5674  3140
5675  2304
5676  1404
5677  0000
5700  2125    MESS26, TEXT    "QUAD LINE FLAG NOT CLEARED BY SDRC"
5701  0104
5702  4014
5703  1116
5704  0540
5705  0614
5706  0107
5707  4016
5710  1724
5711  4003
5712  1405
5713  0122
5714  0504
5715  4002
5716  3140
5717  2304
```

```
5720  2203
5721  0000
5722  2125    MES527, TEXT    "QUAD LINE FLAG NOT CLEARED BY SDRB"
5723  0104
5724  4014
5725  1116
5726  0540
5727  0614
5730  0107
5731  4016
5732  1724
5733  4003
5734  1405
5735  0122
5736  0504
5737  4002
5740  3140
5741  2304
5742  2204
5743  0000
5744  2125    MES528, TEXT    "QUAD LINE FLAG CLEARED BY SDST, SDSS, OR SDLC"
5745  0104
5746  4014
5747  1116
5750  0540
5751  0614
5752  0107
5753  4003
5754  1405
5755  0122
5756  0504
5757  4002
5760  3140
5761  2304
5762  2324
5763  5440
5764  2304
5765  2323
5766  5440
5767  1722
5770  4023
5771  0414
5772  0300
5773  2125    MES529, TEXT    "QUAD LINE FLAG COUNTER FLIP/FLOP NOT PROPERLY CLEARED"
5774  0104
5775  4014
5776  1116
5777  0540
6000  0014
6001  0107
6002  4003
6003  1725
6004  1624
6005  0522
6006  4006
```



```
6007  1411
6010  2057
6011  0014
6012  1720
6013  4016
6014  1724
6015  4020
6016  2217
6017  2005
6020  2214
6021  3140
6022  0314
6023  0501
6024  2205
6025  0400

6026  2411    MES530, TEXT    "TIMING ERROR SKIP INSTRUCTION AND LOGIC"
6027  1511
6030  1607
6031  4005
6032  2222
6033  1722
6034  4023
6035  1311
6036  2040
6037  1116
6040  2324
6041  2225
6042  0324
6043  1117
6044  1640
6045  0116
6046  0440
6047  1417
6050  0711
6051  0300
6052  2411    MES531, TEXT    "TIMING ERROR ALWAYS SET OR SDST ALWAYS SKIPS"
6053  1511
6054  1607
6055  4005
6056  2222
6057  1722
6060  4001
6061  1427
6062  0131
6063  2340
6064  2305
6065  2440
6066  1722
6067  4023
6070  0423
6071  2440
6072  0114
```

```
6073  2701
6074  3123
6075  4023
6076  1311
6077  2023
6100  0000
6101  2411    MESS32, TEXT    "TIMING ERROR DOES NOT SET IN READ MODE OR SDST DOES NOT SKIP"
6102  1511
6103  1607
6104  4005
6105  2222
6106  1722
6107  4004
6110  1705
6111  2340
6112  1617
6113  2440
6114  2305
6115  2440
6116  1116
6117  4022
6120  0501
6121  0440
6122  1517
6123  0405
6124  4017
6125  2240
6126  2304
6127  2324
6130  4004
6131  1705
6132  2340
6133  1617
6134  2440
6135  2313
6136  1120
6137  0000
6140  2411    MESS33, TEXT    "TIMING ERROR CLEARED BY SDST"
6141  1511
6142  1607
6143  4005
6144  2222
6145  1722
6146  4003
6147  1405
6150  0122
6151  0504
6152  4002
6153  3140
6154  2304
6155  2324
6156  0000
6157  2411    MESS34, TEXT    "TIMING ERROR NOT CLEARED BY CAF"
6160  1511
6161  1607
```



```
6162  4005
6163  2222
6164  1722
6165  4016
6166  1724
6167  4003
6170  1405
6171  0122
6172  0504
6173  4002
6174  3140
6175  0301
6176  0600
6177  2411    MESS35, TEXT    "TIMING ERROR STATUS BIT NOT SET IN COMMAND REGISTER"
6200  1511
6201  1607
6202  4005
6203  2222
6204  1722
6205  4023
6206  2401
6207  2405
6210  2340
6211  0211
6212  2440
6213  1617
6214  2440
6215  2305
6216  2440
6217  1116
6220  4003
6221  1715
6222  1501
6223  1604
6224  4022
6225  0507
6226  1123
6227  2405
6230  2200
6231  2411    MESS36, TEXT    "TIMING ERROR NOT CLEARED BY SDLC"
6232  1511
6233  1607
6234  4005
6235  2222
6236  1722
6237  4016
6240  1724
6241  4003
6242  1405
6243  0122
6244  0504
6245  4002
6246  3140
6247  2304
6250  1403
```

```
6291  0000
6292  2411    MES$37, TEXT    "TIMING ERROR DOES NOT SET IN WRITE MODE"
6293  1511
6294  1607
6295  4005
6296  2222
6297  1722
6298  4004
6299  1705
6300  2348
6301  1617
6302  2448
6303  2305
6304  2448
6305  1116
6306  4027
6307  2211
6308  2405
6309  4015
6310  1704
6311  0500
6312  2411    MES$38, TEXT    "TIMING ERROR DOES NOT CLEAR WRITE FLIP/FLOP"
6313  1511
6314  1607
6315  4005
6316  2222
6317  1722
6318  4004
6319  1705
6320  2348
6321  1617
6322  2448
6323  0314
6324  0501
6325  2240
6326  2722
6327  1124
6328  0540
6329  0614
6330  1128
6331  5706
6332  1417
6333  2000
6334  2411    MES$39, TEXT    "TIMING ERROR NOT SET BY $DRC $DRD, OR $DLC"
6335  1511
6336  1607
6337  4005
6338  2202
6339  1722
6340  4016
6341  1724
6342  4003
6343  0524
6344  4002
6345  3140
```



```
6346  2304
6347  2203
6350  4003
6351  0422
6352  0454
6353  4017
6354  2240
6355  2304
6356  1403
6357  0000
6360  2520    MES$43, TEXT    "UP TO SPEED CIRCUITRY CHECK USING 'CLEAR MARK TRACK REGISTER' FEATURE"
6361  4024
6362  1740
6363  2300
6364  0505
6365  0440
6366  0311
6367  2203
6370  2511
6371  2402
6372  3140
6373  0310
6374  0503
6375  1340
6376  2523
6377  1116
6400  0740
6401  4703
6402  1405
6403  0122
6404  4015
6405  0102
6406  1340
6407  2402
6410  0103
6411  1340
6412  2205
6413  0711
6414  2324
6415  0522
6416  4740
6417  0005
6420  0124
6421  2522
6422  0500
6423  1301    MES$44, TEXT    "MARK TRACK REGISTER NOT CLEARED BY 'GO'"
6424  2213
6425  4024
6426  2201
6427  0313
6430  4002
6431  0507
6432  1103
6433  2405
6434  2340
```

```
6427  1617
6430  2440
6431  0314
6432  0581
6433  2285
6434  0440
6435  0231
6436  4047
6437  0717
6440  4700
6441  1581    MESS45, TEXT    "MARK TRACK REGISTER NOT CLEARED BY /STOP/"
6442  2213
6443  4024
6444  2281
6445  0313
6446  4022
6447  0587
6450  1123
6451  2405
6452  2240
6453  1617
6454  2440
6455  0314
6456  0581
6457  2285
6460  0440
6461  0231
6462  4047
6463  2324
6464  1720
6465  4700
6466  1581    MESS46, TEXT    "MARK TRACK REGISTER NOT CLEARED BY /REV/ TO /FWD/"
6467  2213
6470  4024
6471  2281
6472  0313
6473  4022
6474  0587
6475  1123
6476  2405
6477  2240
6500  1617
6501  2440
6502  0314
6503  0581
6504  2285
6505  0440
6506  0231
6507  4047
6510  2285
6511  2647
6512  4024
6513  1740
6514  4706
6515  2704
```



```
6516  4700
6517  1581    MESS47, TEXT    "MARK TRACK REGISTER NOT CLEARED BY /FWD/ TO /REV/"
6520  2213
6521  4024
6522  2281
6523  0313
6524  4022
6525  0587
6526  1123
6527  2405
6530  2240
6531  1617
6532  2440
6533  0314
6534  0581
6535  2285
6536  0440
6537  0231
6540  4047
6541  0627
6542  0447
6543  4024
6544  1740
6545  4722
6546  0526
6547  4700
6550  1581    MESS48, TEXT    "MARK TRACK REGISTER NOT CLEARED BY /UNIT 0/ TO /UNIT 1/"
6551  2213
6552  4024
6553  2281
6554  0313
6555  4022
6556  0587
6557  1123
6560  2405
6561  2240
6562  1617
6563  2440
6564  0314
6565  0581
6566  2285
6567  0440
6570  0231
6571  4047
6572  2516
6573  1124
6574  4060
6575  4740
6576  2417
6577  4047
6600  2516
6601  1124
6602  4061
6603  4700
6604  1581    MESS49, TEXT    "MARK TRACK REGISTER NOT CLEARED BY /UNIT 1/ TO /UNIT 0/"
```

```
6605  2213
6606  4824
6607  2201
6610  0313
6611  4822
6612  0587
6613  1123
6614  2485
6615  2240
6616  1617
6617  2440
6620  0314
6621  0581
6622  2285
6623  0440
6624  0231
6625  4847
6626  2516
6627  1124
6630  4861
6631  4748
6632  2417
6633  4847
6634  2516
6635  1124
6636  4868
6637  4788
6640  4723    MES550, TEXT    "/SELECT ERROR' TESTS"
6641  0514
6642  0583
6643  2440
6644  0522
6645  2217
6646  2247
6647  4824
6650  0523
6651  2423
6652  0000
6653  2411    MES551, TEXT    "TIMING ERROR SET"
6654  1511
6655  1687
6656  4885
6657  2222
6660  1722
6661  4823
6662  0524
6663  0000
6664  1617    MES552, TEXT    "NO SELECT ERROR STATUS FROM UNIT %"
6665  4823
6666  0514
6667  0583
6670  2440
6671  0522
6672  2217
6673  2240
```



```
6674  2324
6675  0124
6676  2523
6677  4886
6700  2217
6701  1548
6702  2516
6703  1124
6704  4861
6705  0000
6706  4727    MES553, TEXT    "/WRITE' NOT CLEARED BY SELECT ERROR"
6707  2211
6710  2485
6711  4748
6712  1617
6713  2440
6714  0314
6715  0581
6716  2285
6717  0440
6720  0231
6721  4823
6722  0514
6723  0583
6724  2440
6725  0522
6726  2217
6727  2280
6730  4727    MES554, TEXT    "/WRITE LOCK OUT' TESTS"
6731  2211
6732  2485
6733  4814
6734  1783
6735  1348
6736  1725
6737  2447
6740  4824
6741  0523
6742  2423
6743  0000
6744  1617    MES555, TEXT    "NO WRITE LOCK OUT STATUS FROM UNIT %"
6745  4827
6746  2211
6747  2485
6750  4814
6751  1783
6752  1348
6753  1725
6754  2440
6755  2324
6756  0124
6757  2523
6760  4886
6761  2217
6762  1548
```

```
6763  2516
6764  1124
6765  4860
6766  0000
6767  4727    MES596, TEXT    "'WRITE' NOT CLEARED BY WRITE LOCK OUT"
6770  2211
6771  2405
6772  4740
6773  1617
6774  2440
6775  0314
6776  0501
6777  2205
7000  0440
7001  0231
7002  4827
7003  2211
7004  2405
7005  4814
7006  1703
7007  1340
7010  1725
7011  2400
7012  0401    MES597, TEXT    "DATA WRITTEN FORWARD"
7013  2401
7014  4827
7015  2211
7016  2424
7017  0516
7020  4806
7021  1722
7022  2701
7023  2204
7024  0000
7025  2722    MES598, TEXT    "WRITE DATA FORWARD"
7026  1124
7027  0540
7030  0401
7031  2401
7032  4806
7033  1722
7034  2701
7035  2204
7036  0000
7037  2209    MES599, TEXT    "READ DATA FORWARD"
7040  0104
7041  4804
7042  0124
7043  0140
7044  0617
7045  2227
7046  0122
7047  0400
7050  2205    MES600, TEXT    "READ DATA BACKWARD"
7051  0104
```



```
7052  4804
7053  0124
7054  0140
7055  0201
7056  0313
7057  2701
7060  2204
7061  0000
7062  0401    MES601, TEXT    "DATA WRITTEN BACKWARD"
7063  2401
7064  4827
7065  2211
7066  2424
7067  0516
7070  4802
7071  0103
7072  1327
7073  0122
7074  0400
7075  2722    MES602, TEXT    "WRITE DATA BACKWARD"
7076  1124
7077  0540
7100  0401
7101  2401
7102  4802
7103  0103
7104  1327
7105  0122
7106  0400
7107  1617    MES603, TEXT    "NO UNIT 0 SELECTED"
7110  4825
7111  1611
7112  2440
7113  6040
7114  2305
7115  1405
7116  0324
7117  0504
7120  0000
```

```
              7200    PAGE
                      /ROUTINE TO CHANGE ALL TD8E IOTS IN PROGRAM FOR MULTIPLE UNIT
                      /ROUTINE IS STARTED AT LOCATION "MODIFY" WITH AC6,7 AND 8 INDICATING
                      /DEVICE SELECTOR BITS 6, 7, AND 8 OF THE CONTROL TO BE TESTED (4,5,6, OR 7)

7200  7604    MODIFY, LAS
7201  0221            AND     MODMS1
7202  3223            DCA     MODCON
7203  1230            TAD     MODTAB
7204  3224            DCA     MODPT1
7205  1227            TAD     MODSIZ
7206  3226            DCA     MODCNT
7207  1624    MODLUP, TAD I   MODPT1
```

```
7210  3225          DCA     MODPT2
7211  1625          TAD I   MODPT2
7212  0222          AND     MODMS2
7213  1223          TAD     MODCON
7214  3625          DCA I   MODPT2
7215  2224          ISE     MODPT1
7216  2226          ISE     MODCNT
7217  5207          JMP     MODLUP
7220  7402          HLT
7221  0030    MODMS1, 30
7222  7747    MODMS2, 7747
7223  0000    MODCON, 0
7224  0000    MODPT1, 0
7225  0000    MODPT2, 0
7226  0000    MODCNT, 0
7227  7461    MODSIZ, .-MODEND+MODTAB
7230  7231    MODTAB, .+1
7231  0207          IOT1
7232  0212          IOT2
7233  0214          IOT3
7234  0244          IOT4
7235  0246          IOT5
7236  0313          IOT6
7237  0320          IOT7
7240  0406          IOT8
7241  0423          IOT9
7242  0425          IOT10
7243  0442          IOT11
7244  0457          IOT12
7245  0461          IOT13
7246  0606          IOT14
7247  0612          IOT15
7250  0621          IOT16
7251  0636          SING1
7252  0645          SING2
7253  0653          IOT17
7254  0662          IOT18
7255  0665          SING3
7256  0670          IOT19
7257  0674          IOT20
7260  0704          SING4
7261  0707          IOT21
7262  0713          IOT22
7263  0723          SING5
7264  0726          IOT23
7265  0732          IOT24
7266  1002          SING6
7267  1004          IOT25
7270  1006          IOT26
7271  1011          IOT27

7272  1012          IOT28
7273  1032          IOT29
7274  1036          IOT30
```



```
7275  1041          QUAD8
7276  1042          IOT31
7277  1044          IOT32
7300  1050          IOT33
7301  1053          IOT34
7302  1057          IOT35
7303  1061          IOT36
7304  1067          IOT37
7305  1072          IOT38
7306  1101          QUAD1
7307  1110          QUAD2
7310  1116          IOT39
7311  1125          IOT40
7312  1130          QUAD3
7313  1133          IOT41
7314  1137          IOT42
7315  1202          QUAD4
7316  1205          IOT43
7317  1211          IOT44
7320  1221          QUAD5
7321  1224          IOT45
7322  1230          IOT46
7323  1240          QUAD6
7324  1242          IOT47
7325  1244          IOT48
7326  1250          IOT49
7327  1254          IOT50
7330  1270          IOT51
7331  1271          IOT52
7332  1273          IOT53
7333  1276          IOT54
7334  1305          IOT55
7335  1324          IOT56
7336  1325          IOT57

7337  1327          IOT58
7340  1334          TIME0
7341  1342          IOT59
7342  1353          TIME1
7343  1402          TIME2
7344  1410          IOT60
7345  1417          IOT61
7346  1422          TIME3
7347  1424          IOT62
7350  1442          IOT63
7351  1445          TIME4
7352  1447          IOT64
7353  1453          IOT65
7354  1465          IOT66
7355  1466          IOT67
7356  1470          IOT68
7357  1475          IOT69
7360  1500          IOT70
7361  1504          IOT71
```

```
7362  1510          IOT72
7363  1523          IOT73
7364  1524          IOT74
7365  1531          IOT75
7366  1532          IOT76
7367  1533          IOT77
7370  1543          IOT78
7371  1605          IOT82
7372  1607          IOT83
7373  1613          IOT84
7374  1626          IOT85
7375  1627          IOT86
7376  1631          IOT87
7377  1635          IOT88
7400  1641          IOT89
7401  1654          IOT90
7402  1655          IOT91
7403  1657          IOT92
7404  1664          IOT93
7405  1670          IOT94
7406  1703          IOT95
7407  1704          IOT96
7410  1706          IOT97
7411  1713          IOT98
7412  1717          IOT99
7413  2004          IOT100
7414  2005          IOT101

7415  2007          IOT102
7416  2014          IOT103
7417  2020          IOT104
7420  2037          IOT105
7421  2040          IOT106
7422  2042          IOT107
7423  2047          IOT108
7424  2053          IOT109
7425  2065          IOT110
7426  2204          IOT111
7427  2206          DISLUP
7430  2211          IOT112
7431  2213          IOT113
7432  2227          IOT114
7433  2241          IOT115
7434  2253          IOT116
7435  2402          IOT117
7436  2410          IOT118
7437  2413          IOT119
7440  2510          IOT120
7441  2512          IOT121
7442  2520          IOT122
7443  2522          IOT123
7444  2605          IOT124
7445  2606          IOT125
7446  2614          IOT126
```



```
7447  2616          IOT127
7450  2635          IOT128
7451  2641          IOT129
7452  2654          SELCT2
7453  2660          IOT130
7454  2701          IOT131
7455  2720          IOT132
7456  2724          IOT133
7457  3606          RSRCH0
7460  3607          IOT134

7461  3615          RSRCH1
7462  3617          IOT135
7463  3620          IOT136
7464  3632          IOT13A
7465  3642          IOT137
7466  3622          IOT138
7467  3640          RSRCH2
7470  3654          RSTURN
7471  4056          IOT139
7472  4112          IOT140
7473  4114          IOT141
7474  4201          IOT142
7475  4203          IOT143
7476  4246          IOT144
7477  4250          IOT145
7500  4406          SRCH0
7501  4407          IOT146
7502  4415          SRCH1
7503  4417          IOT147
7504  4420          IOT148
7505  4422          IOT149
7506  4432          IOT14A
7507  4442          IOT150
7510  4454          IOT15B
7511  4440          SRCH2
7512  3733          IOT151
7513  3734          IOT15A
7514  4514          IOT152
7515  4516          IOT153
7516  4274          IOT154
7517  4540          IOT155
7520  4542          IOT156
7521  4646          IOT157
7522  4651          IOT158
7523  4653          IOT159
7524  4662          IOT160
7525  4614          IOT16A
7526  4612          READ1
7527  4664          IOT161
7530  4674          IOT162
7531  4676          IOT163
7532  4702          IOT164
7533  4704          IOT165
```

```
7534  4710          IOT166
7535  4712          IOT167
7536  4272          IOT168
7537  4277          IOT169
7540  4301          IOT170
7541  2110          IOT171
7542  2064          IOT172
7543  2071          IOT173
7544  2073          IOT174
7545  2102          IOT175
7546  2104          IOT176
7547  2106    MODEND, IOT177


                      $

0160  4110
0161  4245
0162  4200
0163  4225
0164  2701
0165  1000
0166  5076
0167  6777
0170  7752
0171  7756
0172  0077
0173  4707
0174  2234
0175  3000
0176  0212
0177  0215
```



```
0000  00000000  20000001  11111111  111111f1  11111111  11111111  f111f111  11f11111
0100  11111111  11110000  00000000  00000000  00000000  00000000  f111f111  11f11111

0200  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
0300  11111111  111111f1  11111111  11100000  00000000  00000000  00000111  11f11111

0400  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
0500  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111

0600  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
0700  11111111  11f11111  11111111  11f111f1  10000000  00000000  00000111  11f11111

1000  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
1100  11111111  111111f1  11111111  111111f1  11111100  00000000  00001111  11f11111

1200  11111111  11f111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
1300  11111111  11f111f1  11111111  111111f1  11111111  11111111  f0011111  11f11111

1400  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
1500  11111111  11f111f1  11111111  111111f1  11111111  11100001  f111f111  11f11111

1600  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
1700  11111111  11f111f1  11111110  00000000  00000000  00000000  00000011  11f11111

2000  11111111  11f111f1  11111111  111111f1  11111111  11111111  f111f111  11f00000
2100  11111111  11f111f1  11111111  111111f1  11111111  11111000  00001111  11f11111

2200  11111111  11f111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
2300  11111111  11f111f1  11111111  111111f1  11000000  00000000  00000111  11f11111

2400  11111111  11111111  11111111  111111f1  11111111  11111111  f111f111  11f11111
2500  11111111  111111f1  11111110  00000000  00000000  00000000  00111111  11f11111

2600  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
2700  11111111  11f111f1  11111111  111111f1  11111100  00000000  f111f111  11f11111

3000  11f11111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
3100  11111111  11f111f1  11111111  111111f1  11110111  11111111  f111f111  11f11111

3200  11111111  11f111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
3300  11111111  11f111f1  11111111  11000000  00000000  00000000  00000000  00000111

3400  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
3500  11111111  111111f1  11100000  00000000  00000000  00000000  00000000  00001111

3600  11111111  111111f1  11111111  111111f1  11111111  11111111  f111f111  11f11111
3700  11111111  11f111f1  11111111  111111f1  10000000  00001111  f111f111  11f11111
```

```
4000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4100  11111111  11111111  11111111  11111111  11000000  00000000  00011111  11111111

4200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4300  11111111  11111111  11111111  11111100  00000000  00000000  00011111  11111111

4400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4500  11111111  11111111  11111111  11111111  11111100  00000001  11111111  11111111

4600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
4700  11111111  11111111  11111111  11111111  11111000  00000000  00000000  01111111


5000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

5600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
5700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


6000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

6200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

6400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

6600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
6700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111


7000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7100  11111111  11111111  10000000  00000000  00000000  00000000  00000000  00000000

7200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

7400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
7500  11111111  11111111  11111111  11111111  11111111  00000000  00000000  00000000

7600
7700
```



```
AUTO    0017        DISEND  2284        IOT102  2807        IOT149  4422
BADBLK  4461        DISLUP  2286        IOT103  2814        IOT14A  4432
BLK     0027        DISTRK  2285        IOT104  2838        IOT15   0012
BLKCH   2188        EIHLT   0038        IOT105  2837        IOT150  4442
BLKCHK  2408        END2    2484        IOT106  2848        IOT151  3733
BLKCN   0068        ERR1HT  0585        IOT107  2842        IOT152  4514
BLKEND  0078        ERR2HT  0556        IOT108  2847        IOT153  4516
BLKERR  2146        ERROR1  0476        IOT109  2853        IOT154  4274
BLKMRK  2426        ERROR2  0557        IOT11   0442        IOT155  4540
BLKREV  0061        FBLKCT  2113        IOT110  2865        IOT156  4542
BLKSER  0101        FILDCT  3448        IOT111  2284        IOT157  4646
BLKTRY  0055        FILDOT  3442        IOT112  2271        IOT158  4651
BMESS   4105        FILDEC  3481        IOT113  2213        IOT159  4653
BUFF1   7208        FILDPT  3441        IOT114  2287        IOT15A  3734
BUFF2   7408        FILICT  3456        IOT115  2241        IOT15B  4654
CAF     6007        FILIOT  3420        IOT116  2253        IOT16   0021
CCNTR   3224        FILINC  3408        IOT117  2402        IOT160  4662
CHKCHK  4323        FILIPT  3417        IOT118  2410        IOT161  4664
CHKCLA  0408        FILL2   3227        IOT119  2413        IOT162  4674
CHKDAT  4131        FILL2C  3241        IOT12   0497        IOT163  4676
CHKERR  4110        FILL2P  3242        IOT120  2518        IOT164  4702
CHKHLT  4126        FILL1   3243        IOT121  2512        IOT165  4704
CHKMES  4132        FILL1C  3296        IOT122  2508        IOT166  4710
CHKSUM  4744        FILL1P  3297        IOT123  2522        IOT167  4712
CLOOP   0266        FILL23  3268        IOT124  2605        IOT168  4272
CLRWT   4271        FILL2C  3293        IOT125  2606        IOT169  4277
CNTR1   0022        FILL2K  3274        IOT126  2614        IOT16A  4014
CNTR2   0023        FILL2P  3275        IOT127  2616        IOT17   0093
COMLUP  3212        FILLC1  3323        IOT128  2635        IOT170  4301
COMP    3516        FILLC2  3324        IOT129  2641        IOT171  2110
COMPAR  3208        FILLP1  3321        IOT13   0441        IOT172  0044
COMREG  0236        FILLP2  3322        IOT130  2648        IOT173  0071
CRLF    0040        FILPAT  3276        IOT131  2701        IOT174  0073
CSDLC   0405        FILPL1  3303        IOT132  2720        IOT175  0102
CSDLD   0441        FILPL2  3307        IOT133  2724        IOT176  0104
CSDRC   0422        FILPNT  0030        IOT134  3607        IOT177  0106
CSDRD   0456        FILTC   3331        IOT135  3617        IOT18   0042
CSUMRT  4303        FILTP   3325        IOT136  3620        IOT19   0070
DATA    2452        FORMT1  0532        IOT137  3642        IOT2    0212
DATERR  4008        FWDEXP  2422        IOT138  3622        IOT20   0074
DATWD   4040        GOOD    0024        IOT139  4096        IOT21   0707
DATHLT  4036        GPNT#   3225        IOT13A  3632        IOT22   0713
DATLUP  0232        GUARD   2472        IOT14   0606        IOT23   0720
DATMES  4042        HEAD1   0025        IOT140  4112        IOT24   0732
DATNUM  4020        HEAD2   0026        IOT141  4114        IOT25   1004
DATPNT  4041        HEADTP  4050        IOT142  4201        IOT26   1006
DATREG  2201        IN      0021        IOT143  4203        IOT27   1011
DBLOCK  2288        INITST  0305        IOT144  4246        IOT28   1012
DISBL   0256        IOT1    0207        IOT145  4250        IOT29   1032
DISBLK  2233        IOT10   0425        IOT146  4407        IOT3    0214
DISDA   0057        IOT100  2004        IOT147  4417        IOT30   1036
DISDAT  2236        IOT101  2005        IOT148  4420        IOT31   1042
```

```
IOT32   1044    IOT02   1605    ME8834  6137    PASS    3513
IOT33   1098    IOT03   1607    ME8835  6177    PREFIN  2462
IOT34   1053    IOT04   1613    ME8836  6231    QBLUP   1074
IOT35   1057    IOT05   1626    ME8837  6292    QUAD    1024
IOT36   1061    IOT06   1627    ME8838  6276    QUAD0   1041
IOT37   1067    IOT07   1631    ME8839  6324    QUAD1   1101
IOT38   1072    IOT08   1635    ME884   5854    QUAD2   1118
IOT39   1116    IOT09   1641    ME8843  6352    QUAD3   1138
IOT4    0244    IOT9    0423    ME8844  6415    QUAD4   1202
IOT40   1125    IOT10   1654    ME8845  6441    QUAD5   1221
IOT41   1133    IOT11   1655    ME8846  6466    QUAD6   1240
IOT42   1137    IOT12   1657    ME8847  6517    QUAD7   1265
IOT43   1285    IOT13   1664    ME8848  6558    RADDR   4057
IOT44   1211    IOT14   1670    ME8849  6604    RBLKCT  2132
IOT45   1224    IOT15   1703    ME885   5876    RCNT    4056
IOT46   1238    IOT16   1704    ME8850  6648    RCOUNT  4060
IOT47   1242    IOT17   1706    ME8851  6693    RD1MRK  2567
IOT48   1244    IOT18   1713    ME8852  6664    RD6MRK  2515
IOT49   1298    IOT19   1717    ME8853  6706    RDQUAD  4707
IOT5    0246    LOOBED  4465    ME8854  6738    READ    4088
IOT50   1294    LOCK    2445    ME8855  6744    READ1   4012
IOT51   1270    LOOP1   0046    ME8856  6747    READ2   4032
IOT52   1271    M8288A  4705    ME8857  7012    READR   3714
IOT53   1273    M18     0004    ME8858  7035    REVBLK  2476
IOT54   1276    MEBAGE  2264    ME8859  7037    REVEXP  2502
IOT55   1309    ME881   9008    ME886   5123    REVGRD  4061
IOT56   1324    ME8810  9174    ME8860  7098    RLOCDD  3069
IOT57   1325    ME8811  9222    ME8861  7042    ROCK    2237
IOT58   1327    ME8812  9293    ME8862  7075    RSEACH  3008
IOT59   1342    ME8813  9305    ME8863  7107    RSLOOK  3078
IOT6    0313    ME8814  9326    ME887   5136    RSRCH0  3006
IOT60   1418    ME8815  9330    ME888   5191    RSRCH1  3015
IOT61   1417    ME8816  9373    ME889   5141    RSRCH2  3048
IOT62   1424    ME8817  9416    MODCNT  7226    RSTURN  3054
IOT63   1442    ME8818  9441    MODCON  7223    RVBARD  2436
IOT64   1447    ME8819  9471    MODEND  7247    SBCXOR  4714
IOT65   1453    ME882   9021    MODIFY  7208    SBWORD  4745
IOT66   1469    ME8820  9516    MODLUP  7207    SDLC    6774
IOT67   1466    ME8821  9546    MODM81  7221    SDLD    6779
IOT68   1478    ME8822  9564    MODM82  7222    SDRC    6776
IOT69   1475    ME8823  9615    MODPT1  7224    SDRD    6777
IOT7    0328    ME8824  9695    MODPT2  7225    SDSD    6773
IOT70   1508    ME8825  9696    MODSIR  7227    SDSS    6771
IOT71   1504    ME8826  9708    MODTAB  7238    SDST    6772
IOT72   1518    ME8827  9722    MPNTR   23E9    SEARCH  4488
IOT73   1523    ME8828  9744    OCNT    2341    SELCT1  2034
IOT74   1524    ME8829  9773    OK      2744    SELCT2  2094
IOT75   1531    ME883   9044    ONUMB   2348    SELECT  2088
IOT76   1532    ME8830  6006    OPLOOP  2304    SELERR  4229
IOT77   1533    ME8831  6092    OPRINT  23E6    SELHLT  4234
IOT78   1543    ME8832  6101    OUT     0000    SELMES  4236
IOT8    0406    ME8833  6148    PASCNT  3497    SING1   0036
```



```
SING2   0645    WRITE2  4521
SING3   0669    WRITER  3671
SING4   0704    WRDERR  4208
SING5   0723    WRGHLT  4212
SING6   1002    WRQUAD  4701
SING7   1021    WRTLCK  3731
SINGL0  0619    WRRL1   3005
SINGLE  0600    WRRL2   3076
SLOOK   4478    XFER    3000
SP1CT   3456    XXX     4302
SP1PT   3457
SP2CT   3474
SP2PT   3475
SPEC1   3443
SPEC1D  3460
SPEC2   3461
SPEC2D  3476
SRCH0   4406
SRCH1   4419
SRCH2   4440
SUNIT   4471
TIME0   1334
TIME1   1393
TIME2   1402
TIME3   1422
TIME4   1445
TIME5   1463
TIME6   1521
TIMING  1319
TPNTR   3226
TYMERR  4249
TYMHLT  4260
TYMMES  4262
TYPE    0031
UMESS   4102
UNIT    2234
UTSMK1  1624
UTSMK2  1652
UTSMK3  1701
UTSMK4  2002
UTSMK5  2039
UTSMK6  2062
UTSMRK  1600
WADDR   4544
WCNT    4478
WCOUNT  4549
WL0     2673
WL1     2717
WRDMES  4214
WREAD   3824
WRITE   4472
WRITE1  4512
```

```
ERRORS DETECTED: 0
LINKS GENERATED: 147
RUN-TIME:  21 SECONDS
3K CORE USED
```