TAPEDATA

IDENTIFICATION

PRODUCT CODE:	MAINDEC-12-D3FB-D
PRODUCT NAME:	PDP-12 TAPE DATA TEST
DATE CREATED:	NOVEMBER, 1 197Ø
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	WALTER MANTER

1. ABSTRACT

The tape data test is designed to:

A) Test tape read and write ability in both pause and no pause mode.

B) Test read-write amplifier recovery.

C) Test a worst case read-write condition.

D) Provide a long read-write scope loop over blocks 0-777 using WRI and RDE instructions.

E) Provide scope loops on all tests

2. REQUIREMENTS

2.1 Equipment

A) A standard PDP-12

B) A TC-12, PDP-12 linc-tape controller

C) A ASR-33 teletype on equivàlent

2.2 Preliminary Programs

Tape Data test should be preceded by tape control test parts I and II and followed by tape Exerciser test.

2.3 Storage

This program uses instructions field 2 and data field 3 of core. (locations 4000 to 7777)

3. LOADING PROCEDURE

3.1 Method

This program must be loaded with the binary loader.

A) Set the teletype reader switch to FREE.

B) Open the teletype reader and insert the program tape so that the arrows on the tape are visible to and pointing toward the operàtor.

C) Close the reader and set the reader switch to START

D) Set the teletype front panel switch to ON LINE

E) Set the LEFT switches to 7777.

F) Set the RIGHT switches to 4ØØØ.

G) Set the MODE switch to 8 mode.

H) Depress I/O preset

I) Depress START LS

J) When the program tape has been read the ACCUMULATOR must be ØØØØ. If it is not, a read in error has occurred and one might try reloading the binary loader.

K) Remove the program tape from the reader

4. STARTING PROCEDURE

The setting of the LEFT, RIGHT and SENSE switches for normal operation is all switches Ø.

A) Set the mode switch to L-MODE

B) Depress I/O Preset.

C) Depress START 2Ø

Mount tape ?
Unit # ? 1

The program is running; consult the listing for test descriptions.

5. CONTROL SWITCH SETTINGS

5.1 Sense Switch Settings

A) SNS Ø = 1 Ignore any error

B) SNS 1 = 1 Loop on particular test

C) SNS 2 = 1 Loop on Write portion of test

D) SNS 3 = 1 Loop on read portion of test

E) SNS 4 = Ø Fixed data pattern (left and right switches)

F) SNS 4 = 1 Random Data pattern (left and right switches not both zero)

G) SNS 5 = 1 long Scope loop test

Takes 5 + 20 ; rings bell

5.2 Left and Right Switches

Control fixed data pattern and determine starting point of ramdom data pattern.

6. ERRORS

6.1 Error Halts

Correct interperation of error halts must be done utilizing the program listing. All error halts are documented and easily interperated in the program listing.

6.2 Error Printouts

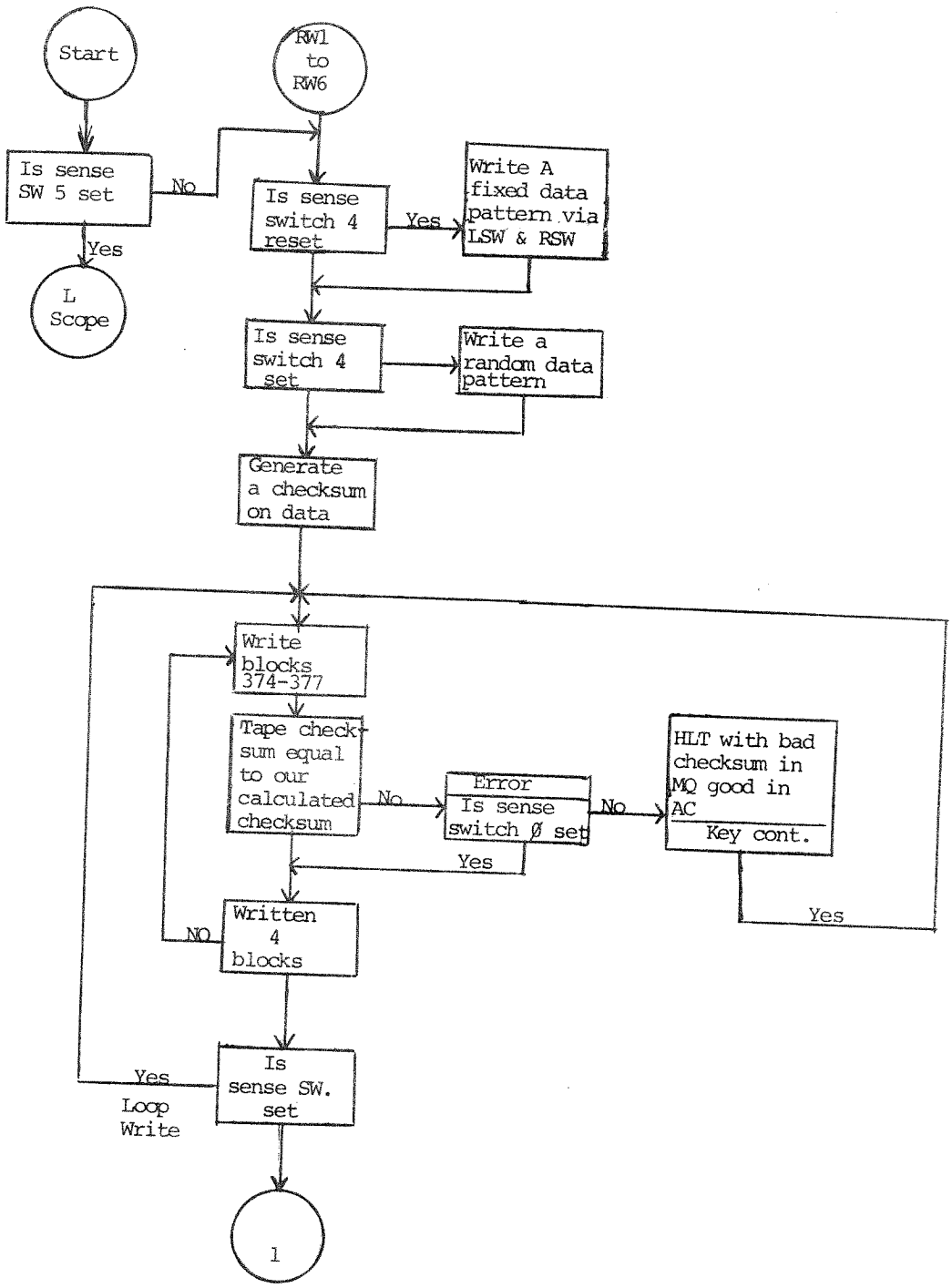
None

6.3 Error Recovery

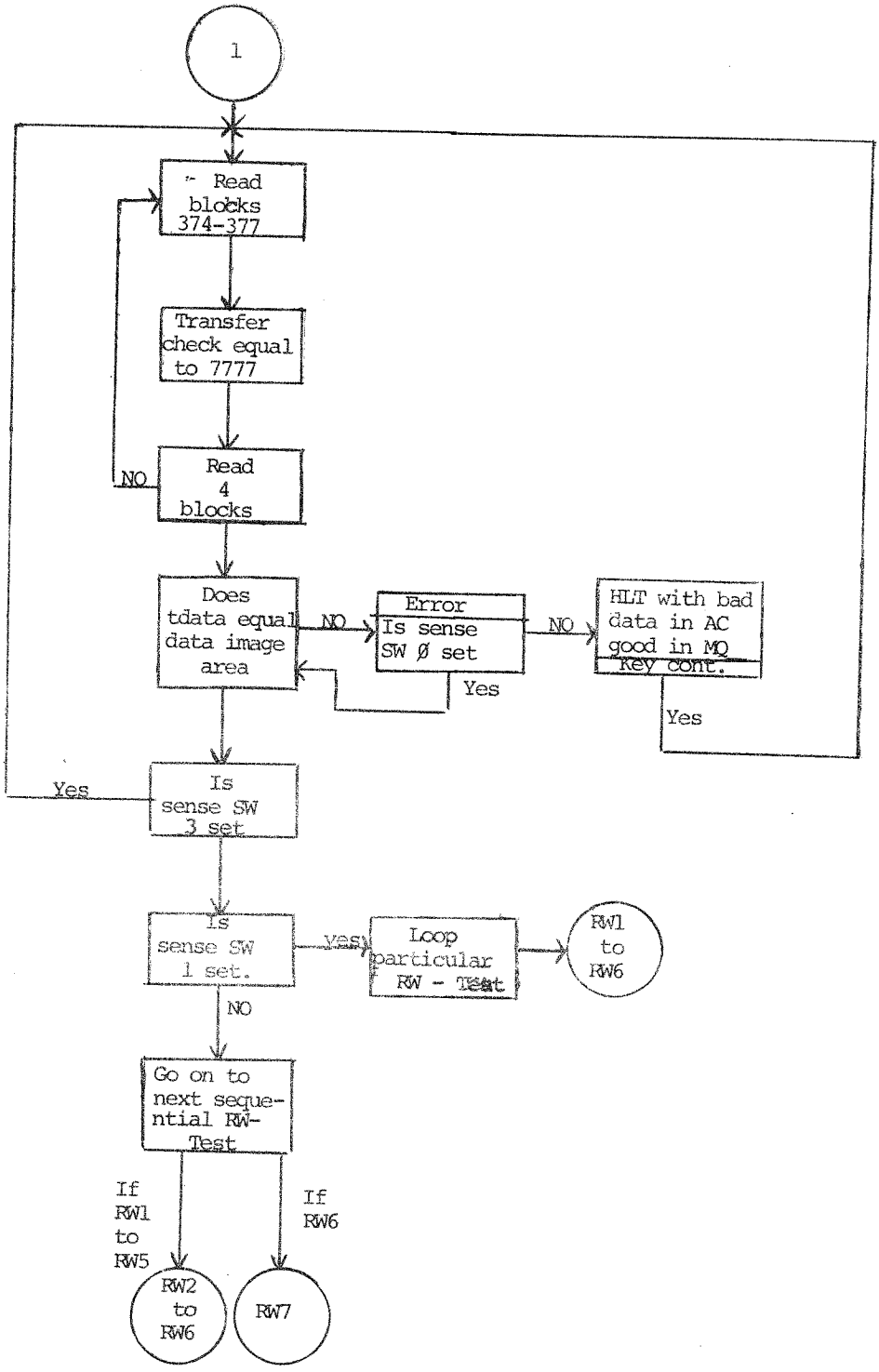
A) KEY CONTINUE puts you back in the main program at the start of test which failed.

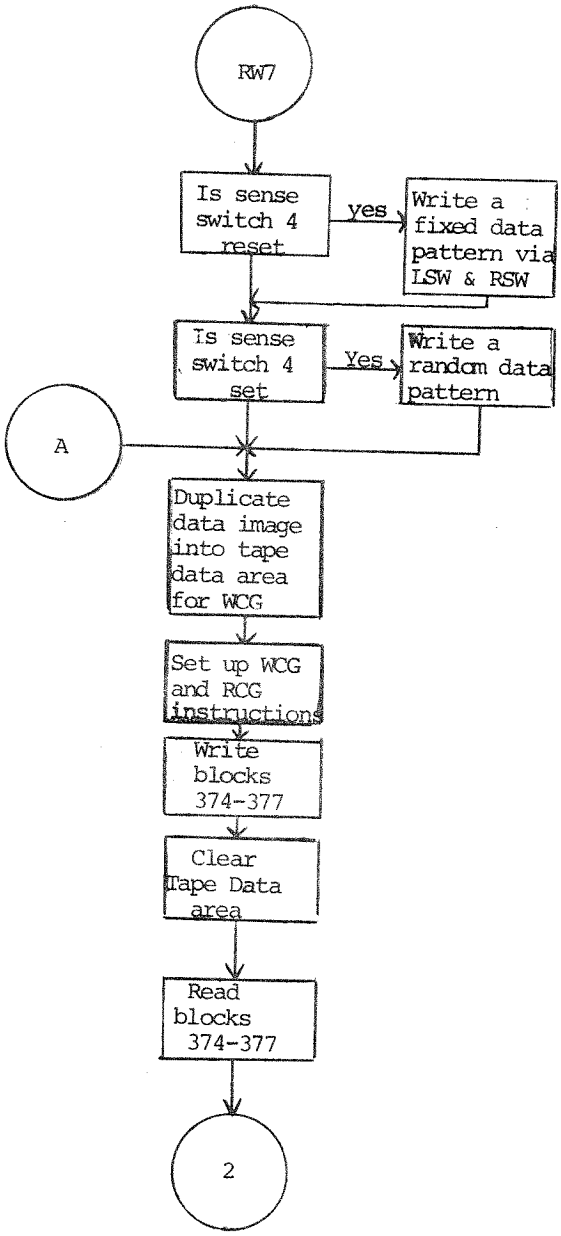
B) If SENSE SWITCH Ø is set (depressed) the error is ignored and program continues in normal sequence.

FLOW CHARTS FOR TAPE DATA TEST

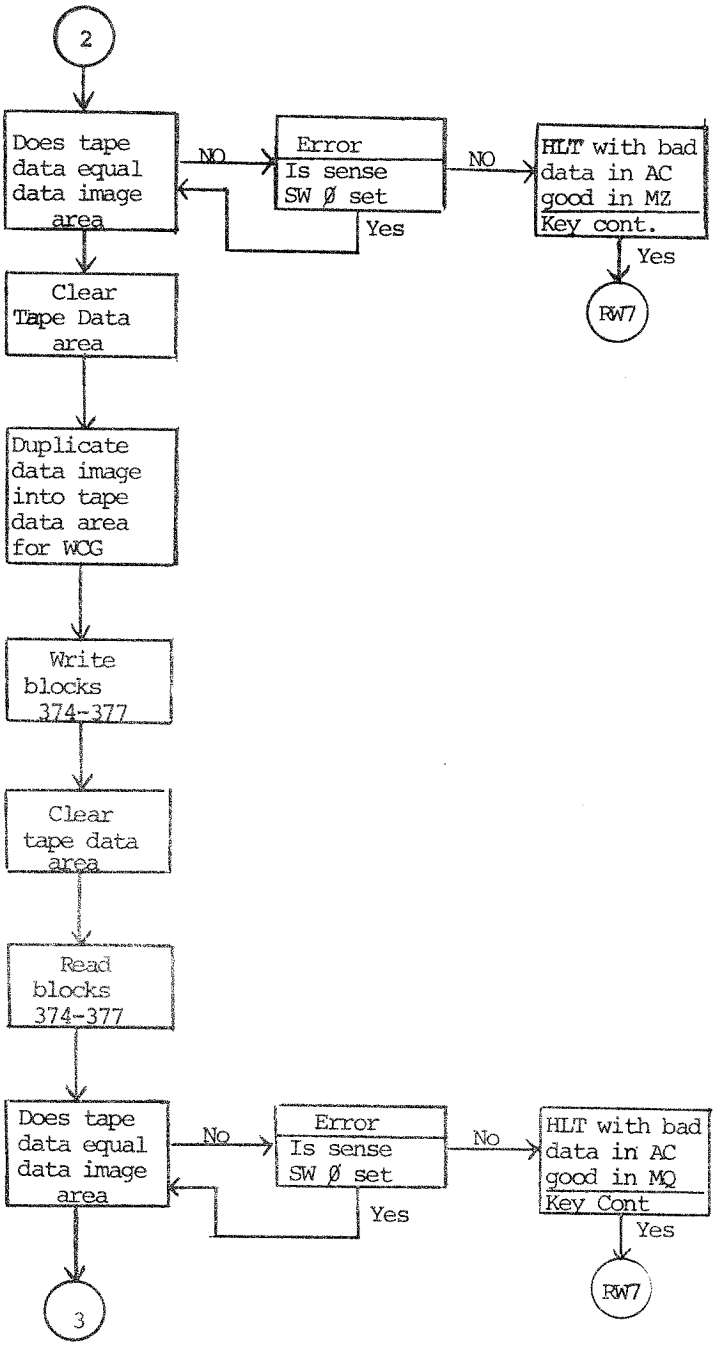


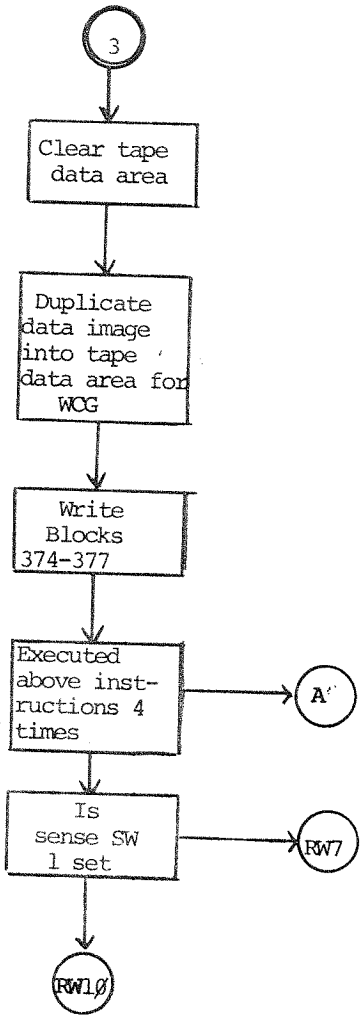
A

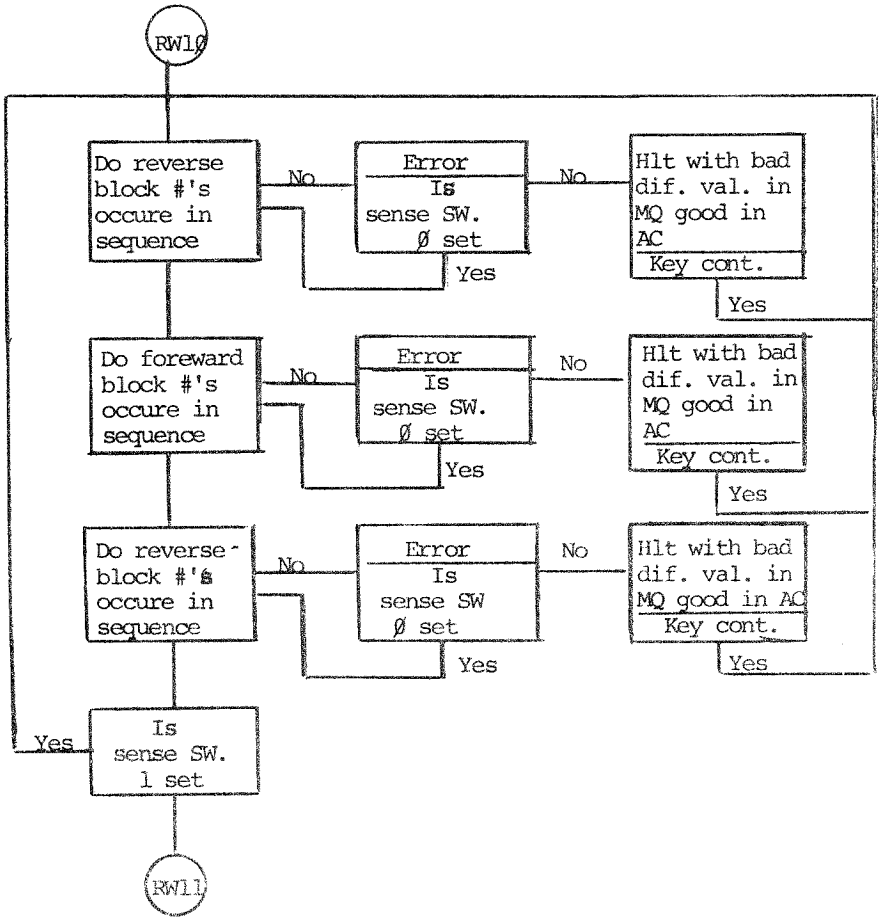

1
Read blocks 374-377
Transfer check equal to 7777
NO
Read 4 blocks
Does tdata equal data image area
NO
Error
Is sense SW Ø set
NO
HLT with bad data in AC good in MQ
Key cont.
Yes
Yes
Yes
Is sense SW 3 set
Is sense SW 1 set.
yes
Loop particular RW - Test
RW1 to RW6
NO
Go on to next sequential RW-Test
If RW1 to RW5
If RW6
RW2 to RW6
RW7

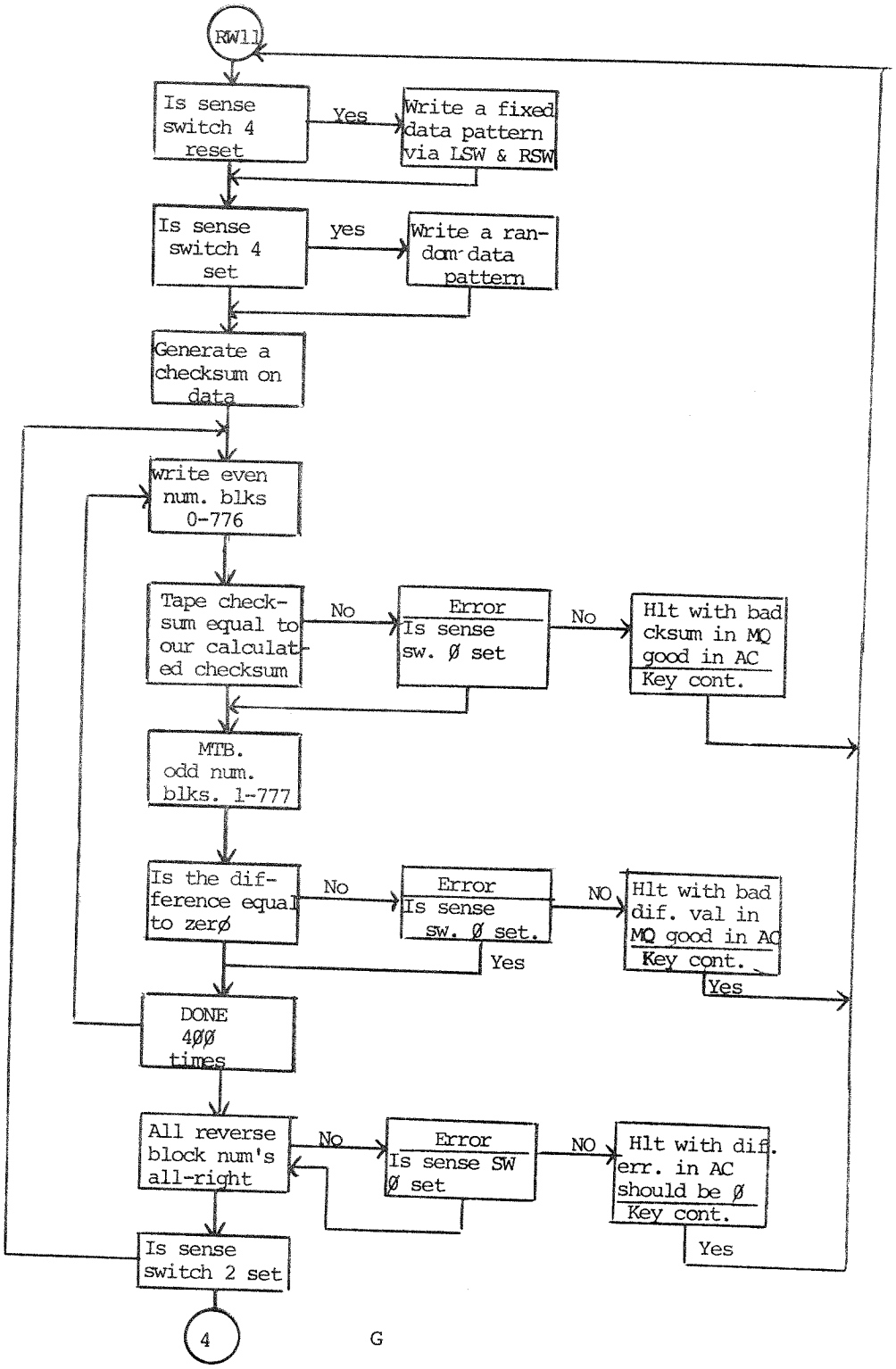

RW7
Is sense switch 4 reset
yes
Write a fixed data pattern via LSW & RSW
Is sense switch 4 set
Yes
Write a random data pattern
A
Duplicate data image into tape data area for WCG
Set up WCG and RCG instructions
Write blocks 374-377
Clear Tape Data area
Read blocks 374-377
2

C

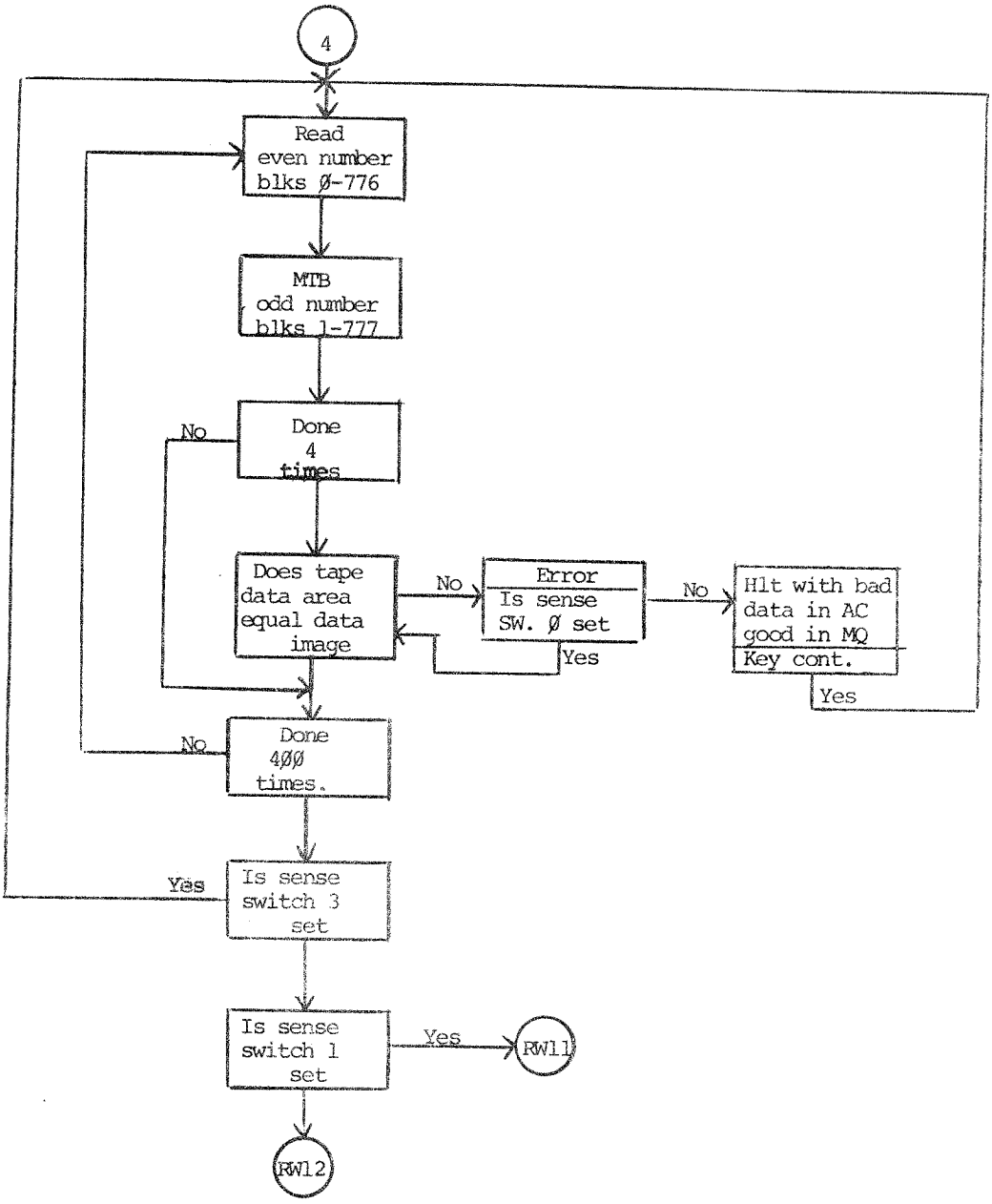

2
Does tape data equal data image area
NO
Error
Is sense SW Ø set
NO
HLT with bad data in AC good in MZ
Key cont.
Yes
RW7
Yes
Clear Tape Data area
Duplicate data image into tape data area for WCG
Write blocks 374-377
Clear tape data area
Read blocks 374-377
Does tape data equal data image area
No
Error
Is sense SW Ø set
No
HLT with bad data in AC good in MQ
Key Cont
Yes
RW7
Yes
3

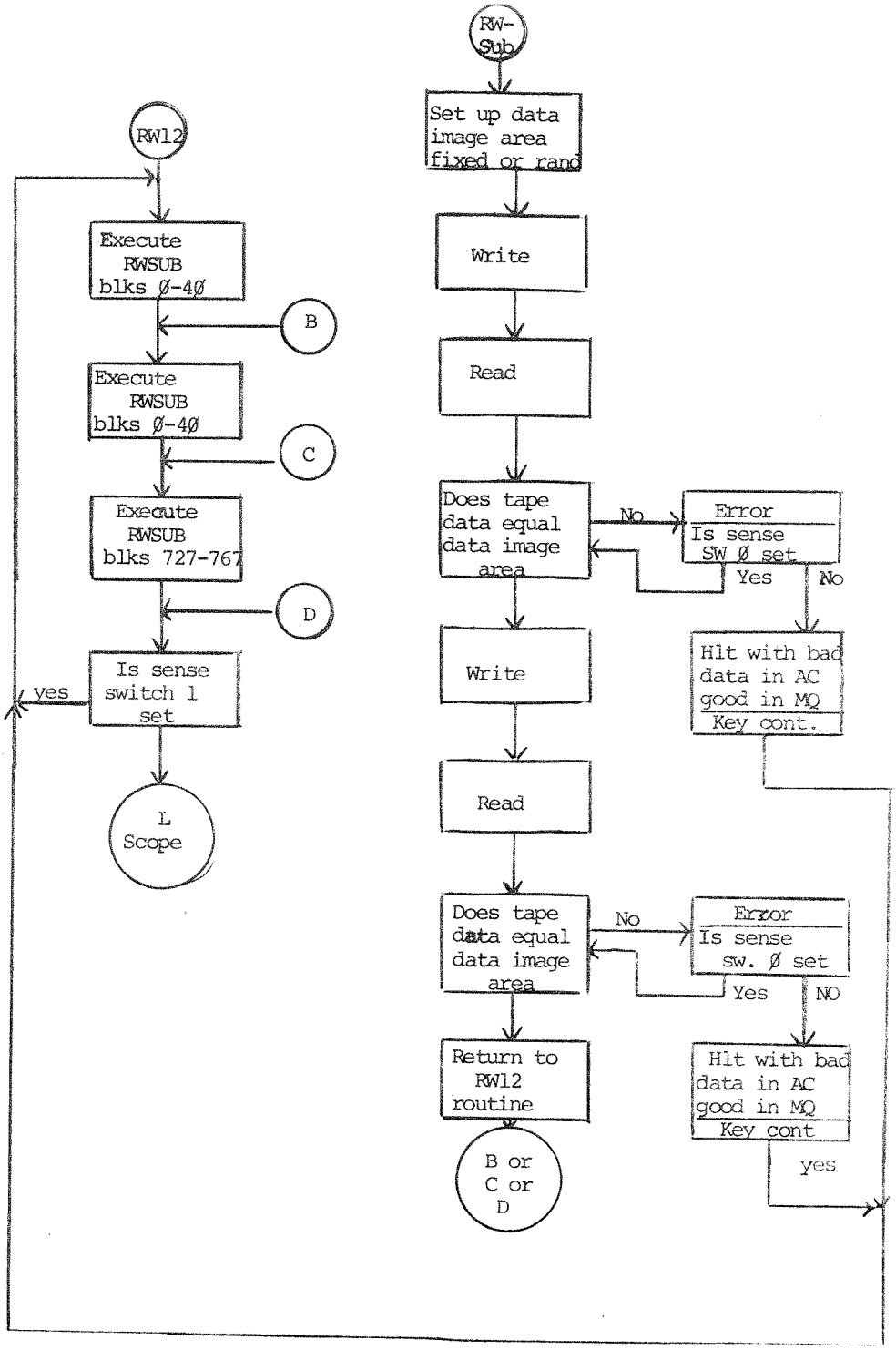

3
Clear tape data area
Duplicate data image into tape data area for WCG
Write Blocks 374-377
Executed above inst-ructions 4 times
A'
Is sense SW 1 set
RW7
RW1Ø

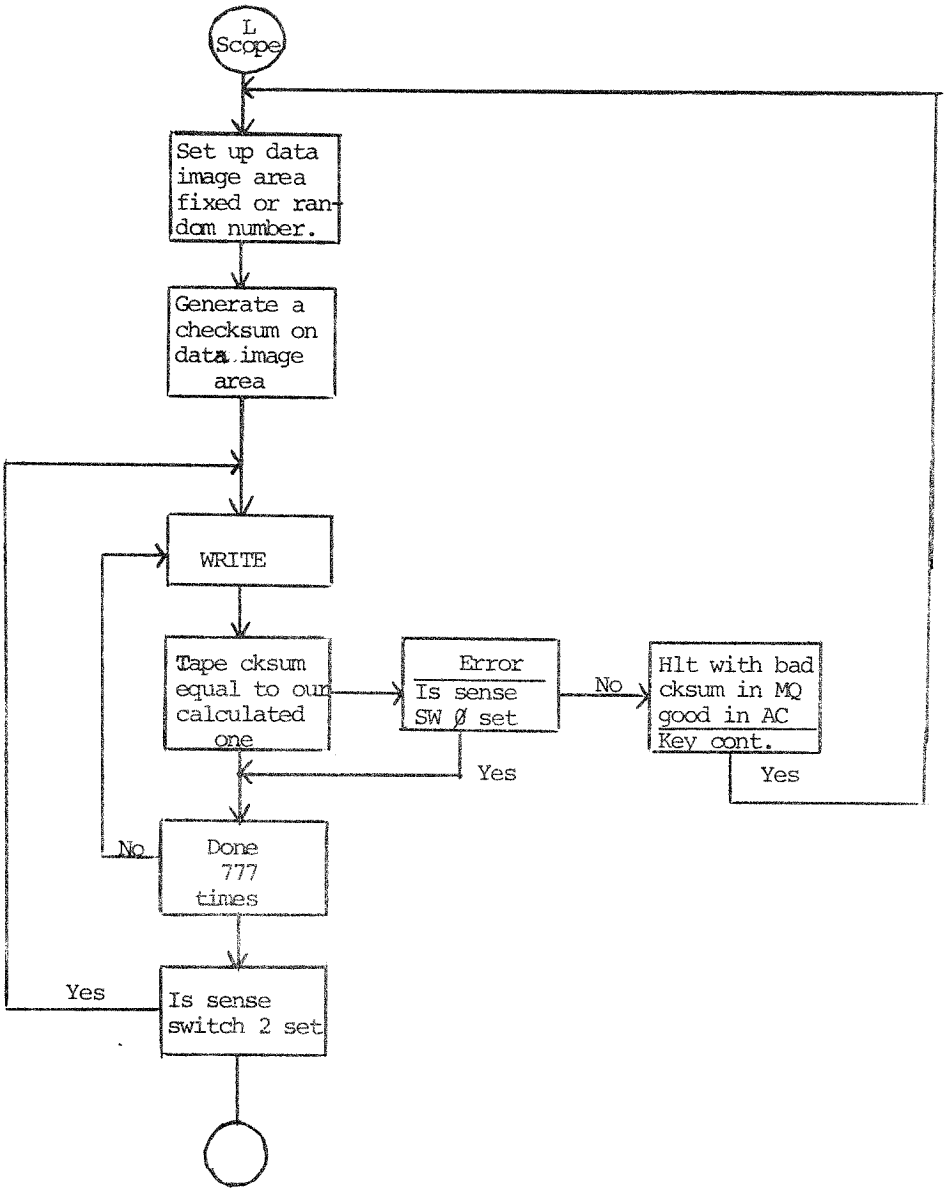

RW1Ø
Do reverse block #'s occure in sequence
No
Error Is sense SW. Ø set
No
Hlt with bad dif. val. in MQ good in AC
Key cont.
Yes
Yes
Do foreward block #'s occure in sequence
No
Error Is sense SW. Ø set
No
Hlt with bad dif. val. in MQ good in AC
Key cont.
Yes
Yes
Do reverse block #'s occure in sequence
No
Error Is sense SW Ø set
No
Hlt with bad dif. val. in MQ good in AC
Key cont.
Yes
Yes
Is sense SW. 1 set
Yes
RW11


RW11
Is sense switch 4 reset
Yes
Write a fixed data pattern via LSW & RSW
Is sense switch 4 set
yes
Write a random data pattern
Generate a checksum on data
write even num. blks 0-776
Tape checksum equal to our calculated checksum
No
Error
Is sense sw. Ø set
No
Hlt with bad cksum in MQ good in AC
Key cont.
MTB. odd num. blks. 1-777
Is the difference equal to zerø
No
Error
Is sense sw. Ø set.
NO
Hlt with bad dif. val in MQ good in AC
Key cont.
Yes
Yes
DONE 4ØØ times
All reverse block num's all-right
No
Error
Is sense SW Ø set
NO
Hlt with dif. err. in AC should be Ø
Key cont.
Yes
Is sense switch 2 set
4

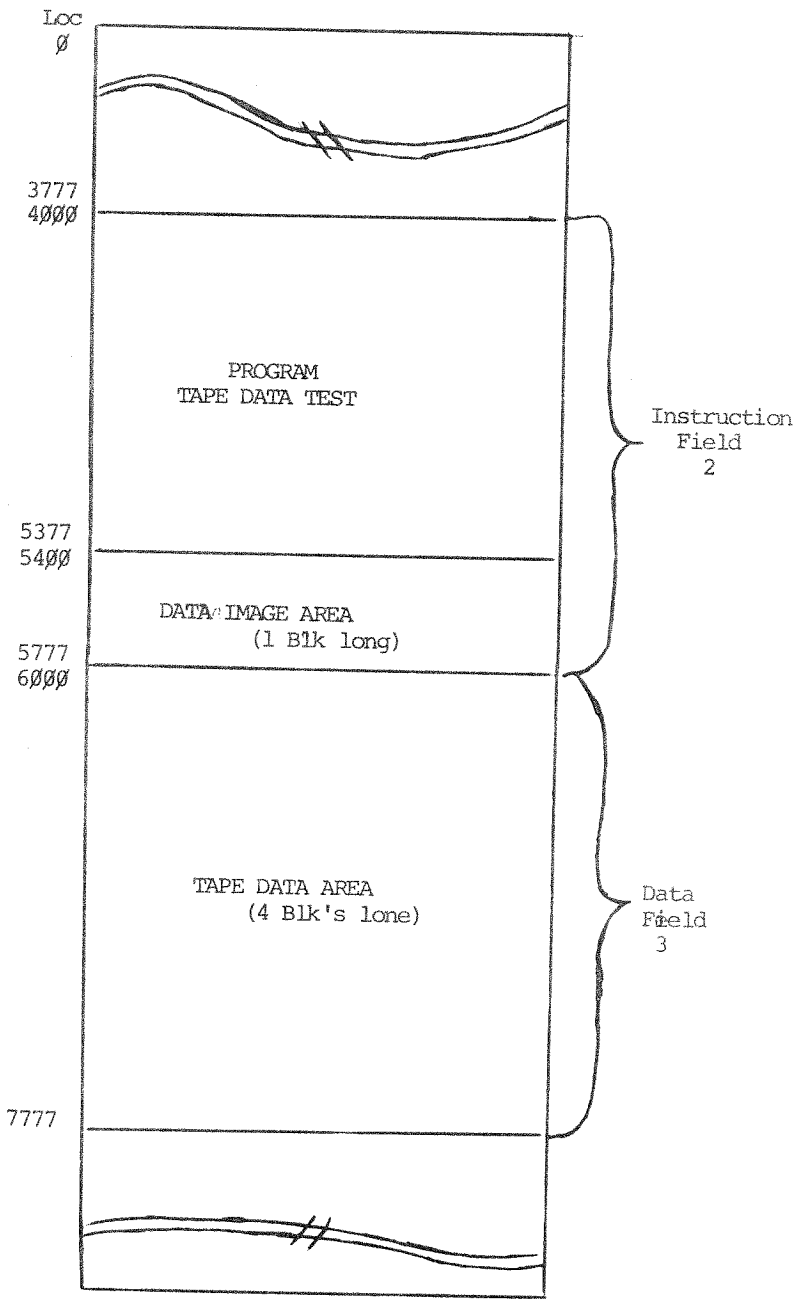
G


4
Read
even number
blks Ø-776
MTB
odd number
blks 1-777
No
Done
4
times
Does tape
data area
equal data
image
No
Error
Is sense
SW. Ø set
No
Hlt with bad
data in AC
good in MQ
Key cont.
Yes
Yes
No
Done
4ØØ
times.
Yes
Is sense
switch 3
set
Is sense
switch 1
set
Yes
RW11
RW12


RW-Sub
Set up data image area fixed or rand
Write
Read
Does tape data equal data image area
No
Error
Is sense SW Ø set
Yes
No
Hlt with bad data in AC good in MQ
Key cont.
Write
Read
Does tape data equal data image area
No
Error
Is sense sw. Ø set
Yes
NO
Hlt with bad data in AC good in MQ
Key cont
yes
Return to RW12 routine
B or C or D
RW12
Execute RWSUB blks Ø-4Ø
B
Execute RWSUB blks Ø-4Ø
C
Execute RWSUB blks 727-767
D
Is sense switch 1 set
yes
L Scope


L
Scope
Set up data image area fixed or random number.
Generate a checksum on data image area
WRITE
Tape cksum equal to our calculated one
Error
Is sense SW Ø set
No
Hlt with bad cksum in MQ good in AC
Key cont.
Yes
Yes
No
Done 777 times
Yes
Is sense switch 2 set

APPENDIX A
MEMORY MAP



```
0000                    *20
0001                            *20
0002
0003                    /TDATA - TAPE DATA TEST    MAINDEC 12-D3FA-A
0004                    /COPYRIGHT 1970, DIGITAL EQUIPMENT CORP.,MAYNARD, MASS.
0005                    /MAINTAINER - DIAGNOSTIC GROOP
0006                    /AUTHOR - WALTER MANTER
0007                    /
0010                    /TDATA1 TESTS:
0011                    /    TAPE READ AND WRITE ABILITY IN BOTH PAUSE AND NO PAUSE MODES
0012                    /    WRITE-READ AMPLIFIER RECOVERY
0013                    /    WORST CASE READ-WRITE CONDITIONS
0014                    /    LONG READ-WRITE SCOPE LOOP OVER BLOCKS 0 TO 777
0015                    /
0016                    /SCOPE LOOPS ARE PROVIDED VIA THE SENSE SWITCHES FOR ALL TESTS
0017                    /
0020                    /SENSE SWITCH SETTINGS ARE:
0021                    /    SNS 0 = 1  IGNORE ANY ERROR
0022                    /    SNS 1 = 1  LOOP ON PARTICULAR TEST
0023                    /    SNS 2 = 1  LOOP ON WRITE PORTION OF TEST
0024                    /    SNS 3 = 1  LOOP ON READ PORTION OF TEST
0025                    /    SNS 4 = 1  FIXED DATA PATTERN (LEFT AND RIGHT SWITCHES)
0026                    /    SNS 4 = 0  RANDOM DATA PATTERN
0027                    /    SNS 5 = 1  LONG SCOPE LOOP TEST
0030                    /
0031                    /
0032                    /NOTE: THIS PROGRAM WRITES OVER BLOCKS 0 TO 777 (USE A SCRATCH TAPE)
0033                    /
0034                    /START LOCATION 20
0035
0036
0037                            LMODE
0040
0041                            *20
0042
0043    0020  0465      START,  SNS I 5          /IS SNS SW 5 SET
0044    0021  6676              JMP LSCOPE      /YES-GO TO LONG SCOPE LOOP ROUTINE
0045
0046
0047
0050
0051
0052                    /0
0053
0054                            EJECT
```

```
0055
0056                    /RW1 TESTS WRI AND RDE TAPE INSTRUCTIONS IN PAUSE MODE
0057                    /WITH THE I BIT (7) SET LEAVING TAPE MOVING IN THE DIRECTION
0060                    /IT WAS MOVING AT COMPLETION OF TAPE INSTRUCTION
0061                    /USES BLOCKS 374-377
0062
0063
0064    0022  0011  RW1,    CLR             /CLEAR
0065    0023  0001          AXO             /EXTENDED OPERATIONS BUFFER
0066    0024  6752          JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0067    0025  7033          JMP GCKSUM      /GENERATE CHECKSUM FOR DATA IMAGE AREA
0070    0026  0011  T2,     CLR
0071    0027  3373          ADD C3374       /SET UP WRI FROM DATA IMAGE TO TAPE
0072    0030  4034          STC X2
0073    0031  0061          SET I 1         /WANT TO WRITE 4 BKS
0074    0032  7773          -4
0075    0033  0736          WRI I U         /WRITE
0076    0034  0000  X2,     0
0077    0035  0003          TAC             /TAC TO AC
0100    0036  1440          SAE             /IS IT EQUAL TO
0101    0037  1044          CKSUM           /OUR CALCULATED CHECKSUM
0102    0040  7271          JMP E2          /NO-ERROR
0103    0041  1000          LDA             /UPDATE TBLK
0104    0042  0034          X2
0105    0043  3367          ADD C1          /BY A COUNT OF 1
0106    0044  4034          STC X2
0107    0045  0221          XSK I 1         /WRITTEN 4 BLOCKS
0110    0046  6033          JMP X2-1        /NO-DO IT AGAIN
0111    0047  0462          SNS I 2         /SNS SW 2 SET
0112    0050  6026          JMP T2          /YES-LOOP ON WRI
0113    0051  0011  T3,     CLR
0114    0052  3374          ADD C4374       /SET UP RDE FROM TAPE TO TDATA AREA
0115    0053  4057          STC X3
0116    0054  0061          SET I 1         /WANT TO RDE 4 BLKS
0117    0055  7773          -4
0120    0056  0732          RDE I U         /READ
0121    0057  0000  X3,     0
0122    0060  1460          SAE I           /IS THE TRANSFER CHECK
0123    0061  7777          7777            /EQUAL TO 7777
0124    0062  7300          JMP E3          /NO-ERROR
0125    0063  1000          LDA             /UPDATE MBLK AND TBLK
0126    0064  0057          X3
0127    0065  3372          ADD C1001       /BY A COUNT OF 1
0130    0066  4057          STC X3
0131    0067  0221          XSK I 1         /READ 4 BLOCKS
0132    0070  6056          JMP X3-1        /NO-DO AGAIN
0133    0071  7141          JMP CMPR        /COMPARE TDATA AREA WITH DATA IMAGE AREA
0134    0072  0463          SNS I 3         /SNS SW 3 SET
0135    0073  6051          JMP T3          /YES-LOOP RDE
0136    0074  0461          SNS I 1         /SNS SW 1 SET
0137    0075  6022          JMP RW1         /YES-LOOP READ-WRITE TEST
0140
0141
0142
0143                    /1
0144
0145                        EJECT
```

```
0146
0147                    /RW2 TESTS WRI AND RDE INSTRUCTIONS     PAUSE MODE
0150                    /WITH THE I BIT (7) RESET
0151                    /DETERMINES IF INSTRUCTIONS DECODE AND EXECUTE
0152                    /USES BLOCK 374
0153
0154
0155    0076  6752  RW2,    JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0156    0077  7033          JMP GCKSUM      /GENERATE CHECKSUM FOR DATA IMAGE AREA
0157    0100  0011  T4,     CLR
0160    0101  0716          WRI U           /WRITE
0161    0102  3374          3374
0162    0103  0003          TAC             /TAC TO AC
0163    0104  1440          SAE             /IS IT EQUAL TO
0164    0105  1044          CKSUM           /OUR CALCULATED CHECKSUM
0165    0106  7304          JMP E4          /NO-ERROR
0166    0107  0462          SNS I 2         /SNS SW 2 SET
0167    0110  6100          JMP T4          /YES-LOOP ON WRI
0170    0111  0011  T5,     CLR
0171    0112  0712          RDE U           /READ
0172    0113  4374          4374
0173    0114  1460          SAE I           /IS THE TRANSFER CHECK
0174    0115  7777          7777            /EQUAL TO 7777
0175    0116  7313          JMP E5          /NO-ERROR
0176    0117  7136          JMP CMPR1       /COMPARE TAPE DATA WITH DATA IMAGE AREA
0177    0120  0463          SNS I 3         /SNS SW 3 SET
0200    0121  6111          JMP T5          /YES-LOOP RDE
0201    0122  0461          SNS I 1         /SNS SW 1 SET
0202    0123  6076          JMP RW2         /YES-LOOP READ-WRITE TEST
0203
0204
0205
0206                    /2
0207
0210                    EJECT
```

```
0211
0212                    /RW3 TESTS WRI AND RDE INSTRUCTIONS IN NO PAUSE MODE
0213                    /WITH THE I BIT (7) SET LEAVING TAPE MOVING IN THE DIRECTION
0214                    /IT WAS GOING AT COMPLETION OF THE INSTRUCTION
0215                    /USES BLOCKS 374-377
0216
0217
0220    0124  1020  RW3,    LDA I           /SET BIT 8 IN AC
0221    0125  0010          10
0222    0126  0001          AXO             /AC TO EXTENDED OPERATIONS BUFFER SETTING NO PAUSE MODE
0223    0127  6752          JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0224    0130  7033          JMP GCKSUM      /GENERATE CHECKSUM FOR DATA IMAGE AREA
0225    0131  0011  T6,     CLR
0226    0132  3373          ADD C3374       /SET UP SECOND WORD OF WRI INST
0227    0133  4137          STC X6
0230    0134  0061          SET I 1         /WANT TO WRITE 4 BLOCKS
0231    0135  7773          -4
0232    0136  0736          WRI I U         /WRITE
0233    0137  0000  X6,     0
0234    0140  0021          XOA             /EXTENDED OPERATIONS BUFFER TO AC
0235    0141  1560          BCL I           /MASK
0236    0142  7767          7767            /ALL BUT NO PAUSE BIT
0237    0143  0450          AZE             /SKIP IF NO PAUSE MODE
0240    0144  0000          HLT             /ERROR-NO PAUSE NOT SET
0241    0145  0416          STD             /SKIP ON TAPE DONE
0242    0146  6145          JMP .-1         /NOT DONE WAIT
0243    0147  0003          TAC             /TAC TO AC
0244    0150  1440          SAE             /IS IT EQUAL TO OUR CALCULATED CHECKSUM
0245    0151  1044          CKSUM
0246    0152  7317          JMP E6          /NO-ERROR
0247    0153  1000          LDA             /UPDATE TBLK
0250    0154  0137          X6
0251    0155  3367          ADD C1          /BY A COUNT OF 1
0252    0156  4137          STC X6
0253    0157  0221          XSK I 1         /WRITTEN 4 BLOCKS
0254    0160  6136          JMP X6-1        /NO-DO IT AGAIN
0255    0161  0462          SNS I 2         /SNS SW 2 SET
0256    0162  6131          JMP T6          /YES-LOOP WRI
0257
0260
0261                    /3
0262
0263
0264                    EJECT
-
```

```
0267    0163  0011  T7,     CLR
0270    0164  3374          ADD C4374       /SET UP SECOND WORD OF RDE INS)
0271    0165  4171          STC X7
0272    0166  0061          SET I 1         /WANT TO READ 4 BLKS
0273    0167  7773          -4
0274    0170  0732          RDE I U         /READ
0275    0171  0000  X7,     0
0276    0172  1020          LDA I           /BIT 5 IN AC ON
0277    0173  0100          100
0300    0174  0500          IOB             /TEST EXECUTION OF 8 MODE IOT
0301    0175  6151          6151            /SKIP ON TAPE DONE FLIP FLOP SET
0302    0176  6174          JMP .-2
0303    0177  0003          TAC             /TAC TO AC
0304    0200  1460          SAE I           /IS THE TRANSFER CHECK
0305    0201  7777          7777            /EQUAL TO 7777
0306    0202  7326          JMP E7          /NO-ERROR
0307    0203  1000          LDA             /UPDATE MBLK AND TBLK
0310    0204  0171          X7
0311    0205  3372          ADD C1001       /BY A COUNT OF 1
0312    0206  4171          STC X7
0313    0207  0221          XSK I 1         /READ 4 BLOCKS
0314    0210  6170          JMP X7-1        /NO-DO IT AGAIN
0315    0211  7141          JMP CMPR        /COMPARE TAPE DATA WITH DATA IMAGE AREA
0316    0212  0463          SNS I 3         /SNS SW 3 SET
0317    0213  6163          JMP T7          /YES-LOOP READ TEST
0320    0214  0461          SNS I 1         /SNS SW 1 SET
0321    0215  6124          JMP RW3         /YES-LOOP WRITE-READ TEST
0322
0323
0324
0325                /4
0326
0327                        EJECT
```

```
0330
0331                    /RW4 TESTS WRC AND RDC INSTRUCTIONS IN PAUSE MODE
0332                    /WITH THE I BIT (7) SET LEAVING TAPE MOVING IN THE DIRECTION
0333                    /IT WAS MOVING AT COMPLETION OF TAPE INSTRUCTION
0334                    /USES BLOCKS 374-377
0335
0336
0337    0216  0011  RW4,    CLR             /CLEAR THE
0340    0217  0001          AXO             /EXTENDED OPERATIONS BUFFER
0341    0220  6752          JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0342    0221  0011  T10,    CLR
0343    0222  3373          ADD C3374       /SET UP WRC FROM DATA IMAGE AREA TO TAPE
0344    0223  4227          STC X10
0345    0224  0061          SET I 1         /WANT TO WRITE 4 BLKS
0346    0225  7773          -4
0347    0226  0734          WRC I U         /WRITE
0350    0227  0000  X10,    0
0351    0230  1000          LDA             /UPDATE TBLK BY
0352    0231  0227          X10
0353    0232  3367          ADD C1          /A COUNT OF 1
0354    0233  4227          STC X10
0355    0234  0221          XSK I 1         /WRITTEN 4 BLOCKS
0356    0235  6226          JMP X10-1       /NO-DO IT AGAIN
0357    0236  0462          SNS I 2         /SNS SW 2 SET
0360    0237  6221          JMP T10         /YES-LOOP WRC TEST
0361    0240  0011  T11,    CLR
0362    0241  3374          ADD C4374       /SET UP SECOND WORD OF RDE INST
0363    0242  4246          STC X11
0364    0243  0061          SET I 1         /WANT TO READ 4 BLKS
0365    0244  7773          -4
0366    0245  0730          RDC I U         /READ
0367    0246  0000  X11,    0
0370    0247  1000          LDA             /UPDATE MBLK AND TBLK BY
0371    0250  0246          X11
0372    0251  3372          ADD C1001       /A COUNT OF 1
0373    0252  4246          STC X11
0374    0253  0221          XSK I 1         /READ 4 BLOCKS
0375    0254  6245          JMP X11-1       /NO-DO IT AGAIN
0376    0255  7141          JMP CMPR        /COMPARE TDATA AREA WITH DATA IMAGE AREA
0377    0256  0463          SNS I 3         /SNS SW 3 SET
0400    0257  6240          JMP T11         /YES-LOOP RDC TEST
0401    0260  0461          SNS I 1         /SNS SW 1 SET
0402    0261  6216          JMP RW4         /YES-LOOP READ-WRITE TEST
0403
0404
0405
0406                /5
0407
0410                        EJECT
-
```

```
0411
0412                    /RW5 TESTS WRC AND RDC INSTRUCTIONS    PAUSE MODE
0413                    /WITH THE I BIT (7) RESET ALLOWING TAPE TO ENTER TURNAROUND
0414                    /AT COMPLETION OF THE INSTRUCTION
0415                    /USES BLOCKS 374-377
0416
0417
0420    0262  6752  RW5,    JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0421    0263  0011  T12,    CLR
0422    0264  3373          ADD C3374       /SET UP SECOND WORD OF WRC INST
0423    0265  4271          STC X12
0424    0266  0061          SET I 1         /WANT TO WRITE 4 BLKS
0425    0267  7773          -4
0426    0270  0714          WRC U           /WRITE
0427    0271  0000  X12,    0
0430    0272  7130          JMP DELAY       /DELAY TO ALLOW TAPES TO ENTER TURNAROUND
0431    0273  1000          LDA             /UPDATE TBLK BY
0432    0274  0271          X12
0433    0275  3367          ADD C1          /A COUNT OF 1
0434    0276  4271          STC X12
0435    0277  0221          XSK I 1         /WRITTEN 4 BLOCKS
0436    0300  6270          JMP X12-1       /NO-DO IT AGAIN
0437    0301  0462          SNS I 2         /SNS SW 2 SET
0440    0302  6263          JMP T12         /YES-LOOP WRC TEST
0441    0303  0011  T13,    CLR
0442    0304  3374          ADD C4374       /SET UP SECOND WORD OF RDC INST
0443    0305  4311          STC X13
0444    0306  0061          SET I 1         /WANT TO READ 4 BLOCKS
0445    0307  7773          -4
0446    0310  0710          RDC U           /READ
0447    0311  0000  X13,    0
0450    0312  7130          JMP DELAY       /DELAY TO ALLOW TAPES TO ENTER TURNAROUND
0451    0313  1000          LDA             /UPDATE MBLK AND TBLK BY
0452    0314  0311          X13
0453    0315  3372          ADD C1001       /A COUNT OF 1
0454    0316  4311          STC X13
0455    0317  0221          XSK I 1         /READ 4 BLOCKS
0456    0320  6310          JMP X13-1       /NO-DO IT AGAIN
0457    0321  7141          JMP CMPR        /COMPARE TAPE DATA WITH DATA IMAGE AREA
0460    0322  0463          SNS I 3         /SNS SW 3 SET
0461    0323  6303          JMP T13         /YES-LOOP RDC TEST
0462    0324  0461          SNS I 1         /SNS SW 1 SET
0463    0325  6262          JMP RW5         /YES-LOOP READ-WRITE TEST
0464
0465
0466
0467                /6
0470
0471                        EJECT
-
```

```
0472
0473                    /RW6 TESTS WRC AND RDC INSTRUCTIONS IN NO PAUSE MODE
0474                    /WITH THE I BIT (7) SET LEAVING TAPE MOVING IN THE DIRECTION
0475                    /IT WAS GOING AT COMPLETION OF THE INSTRUCTION
0476                    /USES BLOCKS 374-377
0477
0500
0501
0502    0326  1020  RW6,    LDA I           /SET BIT 8 IN THE AC
0503    0327  0010          10
0504    0330  0001          AXO             /AC TO EXTENDED OPERATIONS BUFFER SETTING NO PAUSE MODE
0505    0331  6752          JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0506    0332  0011  T14,    CLR
0507    0333  3373          ADD C3374       /SET UP SECOND WORD OF WRC INST
0510    0334  4340          STC X14
0511    0335  0061          SET I 1         /WANT TO WRITE 4 BLKS
0512    0336  7773          -4
0513    0337  0734          WRC I U         /WRITE
0514    0340  0000  X14,    0
0515    0341  0416          STD             /SKIP IF TAPE DONE
0516    0342  6341          JMP .-1         /WAIT
0517    0343  1000          LDA             /UPDATE TBLK BY
0520    0344  0340          X14
0521    0345  3367          ADD C1          /A COUNT OF 1
0522    0346  4340          STC X14
0523    0347  0221          XSK I 1         /WRITTEN 4 BLOCKS
0524    0350  6337          JMP X14-1       /NO-DO IT AGAIN
0525    0351  0462          SNS I 2         /SNS SW 2 SET
0526    0352  6332          JMP T14         /YES-LOOP WRC TEST
0527    0353  0011  T15,    CLR
0530    0354  3374          ADD C4374       /SET UP SECOND WORD OF RDC INST
0531    0355  4361          STC X15
0532    0356  0061          SET I 1         /WANT TO READ 4 BLOCKS
0533    0357  7773          -4
0534    0360  0730          RDC I U         /READ
0535    0361  0000  X15,    0
0536    0362  0416          STD             /SKIP ON TAPE DONE
0537    0363  6362          JMP .-1         /WAIT
0540    0364  1000          LDA             /UPDATE MBLK AND TBLK BY
0541    0365  0361          X15
0542    0366  3372          ADD C1001       /A COUNT OF 1
0543    0367  4361          STC X15
0544    0370  0221          XSK I 1         /READ 4 BLOCKS
0545    0371  6360          JMP X15-1       /NO-DO IT AGAIN
0546    0372  7141          JMP CMPR        /COMPARE TAPE DATA WITH DATA IMAGE AREA
0547    0373  0463          SNS I 3         /SNS SW 3 SET
0550    0374  6353          JMP T15         /YES-LOOP RDC INST
0551    0375  0461          SNS I 1         /SNS SW 1 SET
0552    0376  6326          JMP RW6         /YES-LOOP READ-WRITE TEST
0553
0554
0555
0556                    /7
0557
0560                    EJECT
-
```

```
0561
0562                    /RW7 TESTS WCG AND RCG INSTRUCTIONS    NO PAUSE MODE
0563                    /WITH THE I BIT (7) SET LEAVING TAP    OVING IN THE DIRECTION
0564                    /IT WAS GOING AT COMPLETION OF THE INSTRUCTION
0565                    /USES BLOCKS 374-377
0566
0567
0570
0571    0377  1020  RW7,   LDA I          /SET BIT 8 IN THE
0572    0400  0010         10             /AC
0573    0401  0001         AXO            /AC TO EXTENDED OPERATIONS BUFFER SETTING NO PAUSE MODE
0574    0402  6752         JMP DATSET     /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0575    0403  7107         JMP DUPDAT     /SET TAPE DATA AREA EQUAL TO DATA IMAGE AREA
0576    0404  0061         SET I 1        /WANT TO EXECUTE TEST 4 TIMES
0577    0405  7773         -4
0600    0406  0735  T16,   WCG I U        /WRITE
0601    0407  3374         3374
0602    0410  0416         STD            /SKIP ON TAPE DONE
0603    0411  6410         JMP .-1
0604    0412  7075         JMP CLTDAT     /CLEAR OUT TAPE DATA AREA
0605    0413  0731         RCG I U        /READ
0606    0414  3374         3374
0607    0415  0416         STD            /SKIP ON TAPE DONE
0610    0416  6415         JMP .-1
0611    0417  7141         JMP CMPR       /COMPARE DATA IMAGE AREA WITH TAPE DATA
0612    0420  7063         JMP CLEAR      /CLEAR DATA IMAGE AREA
0613    0421  7107         JMP DUPDAT     /DUPLICATE DATA IMAGE AREA INTO TAPE DATA AREA FOR WCG
0614    0422  6424         JMP .+2        /SKIP THE NEXT INSTRUCTION
0615    0423  6377         JMP RW7        /ERROR RETURN FROM COMPARE ROUTINE IF KEY CONTINUE IS HIT
0616    0424  0735         WCG I U        /WRITE
0617    0425  3374         3374
0620    0426  0416         STD            /SKIP ON TAPE DONE
0621    0427  6426         JMP .-1
0622    0430  6752         JMP DATSET     /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0623    0431  7107         JMP DUPDAT     /DUPLICATE DATA IMAGE AREA INTO TAPE DATA AREA
0624    0432  0715         WCG U          /WRITE
0625    0433  3374         3374
0626    0434  0416         STD            /SKIP ON TAPE DONE
0627    0435  6434         JMP .-1
0630    0436  7075         JMP CLTDAT     /CLEAR OUT TAPE DATA AREA
0631    0437  0711         RCG U          /READ
0632    0440  3374         3374
0633    0441  0416         STD            /SKIP ON TAPE DONE
0634    0442  6441         JMP .-1
0635    0443  7141         JMP CMPR       /COMPARE DATA IMAGE AREA WITH TAPE DATA
0636    0444  7063         JMP CLEAR      /CLEAR OUT DATA IMAGE AREA
0637    0445  7107         JMP DUPDAT     /DUPLICATE DATA IMAGE AREA INTO TAPE DATA AREA FOR WCG 4 BLKS LONG
0640    0446  6450         JMP .+2        /SKIP NEXT INSTRUCTION
0641    0447  6377         JMP RW7        /ERROR RET FROM COMPARE ROUTINE IF KEY CONTINUE HIT
0642    0450  0715         WCG U          /WRITE
0643    0451  3374         3374
0644    0452  0416         STD            /SKIP ON TAPE DONE
0645    0453  6452         JMP .-1
0646    0454  0221         XSK I 1        /EXECUTED TEST 4 TIMES
0647    0455  6406         JMP T16        /NO-DO IT AGAIN
0650    0456  0461         SNS I 1        /IS SNS SW 1 SET
0651    0457  6377         JMP RW7        /YES-LOOP READ WRITE TESTS AGAIN
0652
0653
0654
0655                /8
0656
0657
```

```
0660
0661                    /TEST BLOCK NUMBERS
0662                    /CHECKS ALL FORWARD AND REVERSE BLOCK NUMBERS AND THE SEQUENCE THEY OCCUR IN
0663                    /USES BLOCKS 0-777
0664
0665
0666        0460  0011  RW10,   CLR             /CLEAR
0667        0461  0001          AXO             /EXTENDED OPERATIONS BUFFER
0670        0462  7202          JMP REVBK       /CK REV BK NUMBERS TO 0
0671        0463  7233  T17,    JMP FWDBK       /CK FWD BK NUMBERS FROM APPROXIMATELY 0 TO 777
0672        0464  7202          JMP REVBK       /CK REV BK NUMBERS FROM APPROXIMATELY 777 TO 0
0673        0465  0461          SNS I 1         /SNS SW 1 SET
0674        0466  6463          JMP T17         /YES-LOOP BLOCK NUMBER TEST
0675
0676
0677                    /READ-WRITE AMPLIFIER RECOVERY TEST
0700                    /WRITES A BLOCK THAN MOVES TOWARD THE NEXT BLOCK (MTB) ETC
0701                    /USES BLOCKS 0-777
0702
0703
0704        0467  6752  RW11,   JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
0705        0470  7033          JMP GCKSUM      /GENERATE CHECKSUM FOR DATA IMAGE AREA
0706        0471  1020  T20,    LDA I           /FORMAT FOR SECOND WORD OF WRI INST
0707        0472  3000          3000
0710        0473  4477          STC X20         /STORE IT
0711        0474  0061          SET I 1         /WANT TO WRI AND MOVE TOWARD BLOCK 400 TIMES
0712        0475  7377          -400
0713        0476  0736          WRI I U         /WRITE
0714        0477  0000  X20,    0
0715        0500  0003          TAC             /TAC TO AC
0716        0501  1440          SAE             /IS IT EQUAL TO
0717        0502  1044          CKSUM           /OUR CALCULATED CHECKSUM
0720        0503  7332          JMP E20         /NO-ERROR
0721        0504  1000          LDA             /UPDATE X20 AND SET UP X20A
0722        0505  0477          X20
0723        0506  3367          ADD C1          /UPDATE TBLK FOR MTB INST BY A COUNT OF 1
0724        0507  1040          STA             /STORE IT
0725        0510  0514          X20A
0726        0511  3367          ADD C1          /UPDATE TBLK FOR WRI INST BY COUNT OF 2
0727        0512  4477          STC X20
0730        0513  0733          MTB I U         /MOVE TOWARD BLOCK
0731        0514  0000  X20A,   0
0732        0515  0450          AZE             /IS THE DIFFERENCE VALUE ZERO
0733        0516  7341          JMP E20A        /NO-ERROR
0734        0517  0221          XSK I 1         /WRITTEN AND MTB 400 TIMES
0735        0520  6476          JMP X20-1       /NO-DO IT AGAIN
0736        0521  7202          JMP REVBK       /CK REV BK NUMBERS FROM APPROXIMATELY 777 TO 0
0737        0522  0462          SNS I 2         /SNS SW 2 SET
0740        0523  6471          JMP T20         /YES-LOOP WRITE MTB REV TEST
0741
0742
0743
0744                    /9
0745
0746                    EJECT
```

```
0747
0750    0524  1020  T21,    LDA I          /FORMAT FOR _COND WORD OF RDE INST
0751    0525  4000          4000
0752    0526  4534          STC X21        /STORE IT
0753    0527  0061          SET I 1        /EXECUTE RDE AND MTB 400 TIMES
0754    0530  7377          -400
0755    0531  0062          SET I 2        /AFTER 4 RDE INST CK DATA
0756    0532  7773          -4
0757    0533  0732          RDE I U        /READ
0760    0534  0000  X21,    0
0761    0535  1000          LDA
0762    0536  0534          X21
0763    0537  3367          ADD C1         /UPDATE TBLK OF 2 WORD OF MTB INST BY A COUNT OF 1
0764    0540  1040          STA
0765    0541  0546          X21A
0766    0542  1100          ADA
0767    0543  1372          C1001          /ALSO TBLK AND MBLK OF SECOND WORD OF RDC INST
0770    0544  4534          STC X21        /BY A COUNT OF 2AND 1 RESPECTIVELY
0771    0545  0733          MTB I U        /MOVE TOWARD BLOCK
0772    0546  0000  X21A,    0
0773    0547  0450          AZE            /ARE WE THERE
0774    0550  7345          JMP E21A       /NO-ERROR
0775    0551  0222          XSK I 2        /DONE 4 TIMES
0776    0552  6570          JMP DONE       /NO-UPDATE EXECUTION CNTR
0777    0553  7141          JMP CMPR       /YES-COMPARE DATA IMAGE AREA WITH TAPE DATA
1000    0554  0062          SET I 2        /SET UP NEXT COMPARE AFTER READING 4 MORE BLOCKS
1001    0555  7773          -4
1002    0556  6560          JMP .+2        /SKIP THE NEXT INSTRUCTION
1003    0557  6467          JMP RW11       /ERROR RETURN FROM COMPARE ROUTINE IF KEY CONTINUE HIT
1004    0560  1000          LDA            /SET UP TBLK AND MBLK FOR NEXT 4 PASSES
1005    0561  0534          X21
1006    0562  3370          ADD C1M        /BY SUBTRACTING 1 FROM TBLK
1007    0563  1560          BCL I          /AND CLEARING BITS 0-2 OF AC <MBLK>
1010    0564  7000          7000
1011    0565  1620          BSE I          /SET MBLK EQUAL TO 4
1012    0566  4000          4000
1013    0567  4534          STC X21        /NOW DATA WILL BE READ FROM TAPE INTO START OF TAPE DATA AREA
1014    0570  0221  DONE,    XSK I 1        /ARE WE AT BLK 777 YET <FINISHED>
1015    0571  6533          JMP X21-1      /NO-DO IT AGAIN
1016    0572  0463          SNS I 3        /IS SNS SW 3 SET
1017    0573  6524          JMP T21        /YES-LOOP READ AND MTB INSTRUCTIONS
1020    0574  0461          SNS I 1        /IS SNS SW 1 SET
1021    0575  6467          JMP RW11       /YES-LOOP WRITE READ TESTS
1022
1023
1024
1025                  /10
1026
1027                  EJECT
-
```

```
1030
1031                    /A WORST CASE READ-WRITE TEST
1032                    /USES BLOCKS 366-401, 0-14, AND 750-764
1033
1034
1035    0576  6752  RW12,   JMP DATSET       /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
1036    0577  0061          SET I 1          /EXECUTE 14 TIMES
1037    0600  7762          -15
1040    0601  1020          LDA I            /SET UP WRC AND RDC INSTRUCTIONS (START AT BLK 366)
1041    0602  3365          3365
1042    0603  0221          XSK I 1          /EXECUTED RWSUB 14 TIMES
1043    0604  6631          JMP RWSUB        /NO-EXECUTE READ WRITE SUBROUTINE AGAIN
1044    0605  0061          SET I 1          /EXECUTE 14 TIMES
1045    0606  7762          -15
1046    0607  1020          LDA I            /SET UP WRC AND RDC INSTRUCTIONS (START AT BLK 1)
1047    0610  3000          3000
1050    0611  0221          XSK I 1          /EXECUTED RWSUB 14 TIMES
1051    0612  6631          JMP RWSUB        /NO-EXECUTE READ WRITE SUBROUTINE AGAIN
1052    0613  0061          SET I 1          /EXECUTE 14 TIMES
1053    0614  7762          -15
1054    0615  1020          LDA I            /SET UP WRC AND RDC INSTRUCTIONS (START AT BLK 750)
1055    0616  3747          3747
1056    0617  0221          XSK I 1          /EXECUTED RWSUB 14 TIMES
1057    0620  6631          JMP RWSUB        /NO EXECUTE READ WRITE SUBROUTINE AGAIN
1060    0621  7202          JMP REVBK        /CK REV BLK NUMBERS
1061    0622  7233          JMP FWDBK        /CK FWD BLK NUMBERS
1062    0623  0461          SNS I 1          /SNS SW 1 SET
1063    0624  6576          JMP RW12         /YES-LOOP READ WRITE TEST
1064    0625  1020          LDA I
1065    0626  0207          207              /SET UP AC TO RING BELL IN TYPE ROUTINE
1066    0627  7056          JMP TYPE
1067    0630  6020          JMP START        /GO BACK TO BEGINNING OF PROGRAM AND START OVER
1070
1071
1072
1073                    /11
1074
1075                    EJECT
-
```

```
1076
1077    0631  3367  RWSUB,  ADD C1          /ADD 1 TO TBLK
1100    0632  1040          STA             /STORE IN SECOND WORD OF FIRST WRC
1101    0633  0657          X22
1102    0634  3367          ADD C1          /ADD 1 TO TBLK
1103    0635  1040          STA             /STORE IN SECOND WORD OF SECOND WRC
1104    0636  0664          X22B
1105    0637  1000          LDA             /LOAD SECOND WORD OF FIRST WRC INTO AC FOR
1106    0640  0657          X22             /MODIFICATION OF MBLK PORTION TO SET UP RDC INSTRUCTIONS
1107    0641  1560          BCL I           /CLEAR OUT AC BITS 0-2 LEAVING REST ALONE
1110    0642  7000          7000
1111    0643  1620          BSE I           /INSERT 4 IN BITS 0-2 OF AC LEAVING REST ALONE
1112    0644  4000          4000
1113    0645  1040          STA             /STORE IN SECOND WORD OF FIRST RDC INSTRUCTION
1114    0646  0661          X22A
1115    0647  3367          ADD C1          /ADD 1 TO TBLK
1116    0650  4670          STC X22C        /STORE IN SECOND WORD OF SECOND RDC INSTRUCTION
1117    0651  1000          LDA             /SET UP RET JUMP FROM READ WRITE SUBROUTINE
1120    0652  0000          0
1121    0653  1120          ADA I           /BY SUBTRACTING 2 FROM CONTENTS OF LOC 0
1122    0654  7775          7775
1123    0655  4006          STC 6           /AND STORING IN LOCATION 6 FOR RETURN JUMP
1124    0656  0734          WRC I U         /WRITE
1125    0657  0000  X22,     0
1126    0660  0710          RDC U           /READ
1127    0661  0000  X22A,    0
1130    0662  7136          JMP CMPR1       /COMPARE TAPE DATA WITH DATA IMAGE AREA 1 BLK AT A TIME
1131    0663  0714          WRC U           /WRITE
1132    0664  0000  X22B,    0
1133    0665  6667          JMP ,+2         /SKIP THE NEXT INSTRUCTION
1134    0666  6576          JMP RW12        /ERROR RETURN FROM CMPR1 ROUTINE IF KEY CONTINUE HIT
1135    0667  0730          RDC I U         /READ
1136    0670  0000  X22C,    0
1137    0671  7136          JMP CMPR1       /COMPARE TAPE DATA WITH DATA IMAGE 1 BLK AT A TIME
1140    0672  1000          LDA             /SET UP INCREMENTING OF TBLK AT BEGINING OF
1141    0673  0657          X22             /NEXT PASS THROUGH RWSUB IF NOT DONE
1142    0674  6006          JMP 6           /RETURN TO MAIN PROGRAM
1143    0675  6576          JMP RW12        /ERROR RETURN FROM CMPR1 IF KEY CONTINUE HIT
1144
1145
1146
1147                  /12
1150
1151
-                             EJECT
```

```
1152
1153                    /PROVIDES LONG SCOPE LOOPS USING WRI AND RDE INST
1154                    /WRITES AND READS BLOCKS 0-777
1155
1156
1157    0676  0011  LSCOPE, CLR             /CLEAR THE
1160    0677  0001          AXO             /EXTENDED OPERATIONS BUFFER
1161    0700  6752          JMP DATSET      /SEL AND STORE DATA PATTERN IN DATA IMAGE AREA
1162    0701  7033          JMP GCKSUM      /GENERATE CHECKSUM FOR DATA IMAGE AREA
1163    0702  0061  T23,    SET I 1         /WANT TO WRITE 777 BLOCKS
1164    0703  7000          -777
1165    0704  1020          LDA I           /SET UP SECOND WORD OF WRI INST
1166    0705  3000          3000
1167    0706  4710          STC X23
1170    0707  0736          WRI I U         /WRITE
1171    0710  0000  X23,    0
1172    0711  0003          TAC             /TAC TO AC
1173    0712  1440          SAE             /IS IT EQUAL TO
1174    0713  1044          CKSUM           /OUR CALCULATED CHECKSUM
1175    0714  7354          JMP E23         /NO-ERROR
1176    0715  1000          LDA             /UPDATE TBLK
1177    0716  0710          X23
1200    0717  3367          ADD C1          /BY A COUNT OF 1
1201    0720  4710          STC X23
1202    0721  0221          XSK I 1         /WRITTEN 777 BLOCKS
1203    0722  6707          JMP X23-1       /NO-DO IT AGAIN
1204    0723  0462          SNS I 2         /SNS SW 2 SET
1205    0724  6702          JMP T23         /YES-LOOP ON WRI
1206    0725  0061  T24,    SET I 1         /WANT TO READ 777 BLKS
1207    0726  7000          -777
1210    0727  1020          LDA I           /SET UP SECOND WORD OF RDE INST
1211    0730  4000          4000
1212    0731  4733          STC X24
1213    0732  0732          RDE I U         /READ
1214    0733  0000  X24,    0
1215    0734  1460          SAE I           /IS THE TRANSFER CHECK
1216    0735  7777          7777            /EQUAL TO 7777
1217    0736  7363          JMP E24         /NO-ERROR
1220    0737  1000          LDA             /UPDATE TBLK
1221    0740  0733          X24
1222    0741  3367          ADD C1          /BY A COUNT OF 1
1223    0742  4733          STC X24
1224    0743  0221          XSK I 1         /READ 777 TIMES
1225    0744  6732          JMP X24-1       /NO-DO IT AGAIN
1226    0745  0463          SNS I 3         /SNS SW 3 SET
1227    0746  6725          JMP T24         /YES-LOOP ON RDE INST
1230    0747  0461          SNS I 1         /SNS SW 1 SET
1231    0750  6676          JMP LSCOPE      /YES-LOOP LONG SCOPE LOOP TEST
1232    0751  6020          JMP START       /GO BACK TO BEGINNING OF PROGRAM
1233
1234
1235
1236                    /13
1237
1240                    EJECT
```

```
1241
1242                    /SENSE SWITCH 4 SELECTS EITHER A FIXED OR RANDOM DATA PATTERN
1243
1244
1245    0752  0047  DATSET, SET 7              /SET UP RET JUMP FROM LOC 7 TO MAIN PROG
1246    0753  0000          Ø
1247    0754  0464          SNS I 4           /SNS SW 4 SET
1250    0755  6760          JMP FXDAT         /YES-FIXED DATA PATTERN
1251    0756  0444          SNS 4             /SNS SW 4 NOT SET
1252    0757  6776          JMP RDAT          /RANDOM DATA PATTERN
1253
1254                    /FIXED NUMBER GENERATOR
1255                    /STORES CONTENTS OF LEFT AND RIGHT SWITCHES ALTERNATLY
1256                    /INTO DATA IMAGE AREA (LOCATIONS 1400-1777 INST FIELD 2)
1257
1260
1261    0760  0071  FXDAT,  SET I 11          /STARTING LOC OF DATA TABLE-1
1262    0761  1377          DATA-1
1263    0762  0073          SET I 13          /EXECUTE 400 TIMES (1 BLK)
1264    0763  7377          -400
1265    0764  0516  FXLOAD, RSW               /RIGHT SW TO AC
1266    0765  1071          STA I 11          /INC AND STORE IN DATA TABLE
1267    0766  0233          XSK I 13          /DONE 400 TIMES
1270    0767  6771          JMP .+2           /NO-CONTINUE
1271    0770  6007          JMP 7             /YES-RET TO MAIN PROG
1272    0771  0517          LSW               /LEFT SW TO AC
1273    0772  1071          STA I 11          /INC AND STORE IN DATA TABLE
1274    0773  0233          XSK I 13          /DONE 400 TIMES
1275    0774  6764          JMP FXLOAD        /NO-DO IT AGAIN
1276    0775  6007          JMP 7             /YES-RET TO MAIN PROG
1277
1300
1301
1302                    /14
1303
1304                          EJECT
-
```

```
1305
1306                    /RANDOM NUMBER GENERATOR
1307                    /STORES RANDOM NUMBERS IN DATA IMAGE AREA (LOCATIONS 1400-1777 INST FIELD 2)
1310
1311
1312    0776  0071  RDAT,   SET I 11          /STARTING LOC OF DATA TABLE-1
1313    0777  1377          DATA-1
1314    1000  0073          SET I 13          /EXECUTE 400 TIMES (1 BLK)
1315    1001  7377          -400
1316    1002  0516          RSW               /RIGHT SW TO AC
1317    1003  0450          AZE               /SKIP IF AC EQUALS ZERO
1320    1004  7007          JMP .+3
1321    1005  1020          LDA I             /AC IS ZERO ADD 11 TO IT
1322    1006  0011          11
1323    1007  5024          STC RNA           /STORE AC IN RANDOM NUMBER A
1324    1010  0517          LSW               /LEFT SW TO AC
1325    1011  5025          STC RNB           /STORE AC IN RANDOM NUMBER B
1326    1012  7026  RLOAD,  JMP RADD          /ADD ROUTINE
1327    1013  5024          STC RNA           /STORE IN RNA
1330    1014  0241          ROL 1             /ROT LINK INTO AC BIT 11
1331    1015  7026          JMP RADD          /ADD ROUTINE
1332    1016  1071          STA I 11          /INC AND STORE AC IN DATA TABLE
1333    1017  5024          STC RNA           /STORE IN RNA
1334    1020  0241          ROL 1             /ROT LINK INTO AC BIT 11
1335    1021  0233          XSK I 13          /DONE 400 TIMES (DATA TABLE FULL)
1336    1022  7012          JMP RLOAD         /NO-DO IT AGAIN
1337    1023  6007          JMP 7             /RET TO MAIN PROG
1340    1024  0000  RNA,     0                /RANDOM NUMBER A
1341    1025  0000  RNB,     0                /RANDOM NUMBER B
1342
1343    1026  1200  RADD,   LAM               /ADD RNA TO AC
1344    1027  1024          RNA
1345    1030  1200          LAM               /ADD RNB TO AC
1346    1031  1025          RNB
1347    1032  6000          JMP 0             /RET TO RLOAD ROUTINE
1350
1351
1352
1353                    /15
1354
1355                            EJECT
-
```

```
1356
1357                    /GENERATES A CHECKSUM OF DATA IMAGE  .EA (1 BK LONG)
1360                    /WHICH IS COMPARED WITH THE CONTENTS OF THE TAC ON A WRI INST
1361                    /THE CHECKSUM IS THE TWOS COMPLIMENT OF <DATA SUM PLUS 7777>
1362
1363
1364    1033  0073  GCKSUM, SET I 13          /EXECUTE 400 TIMES (1 BLOCK)
1365    1034  7377          -400
1366    1035  0071          SET I 11          /STARTING ADDRESS OF DATA TABLE-1
1367    1036  1377          DATA-1
1370    1037  0011          CLR               /AC INITIALLY CLEARED
1371    1040  5044          STC CKSUM         /ZERO LOC CKSUM
1372    1041  1031          LDA I 11          /INC AND LOAD AC FROM DATA TABLE
1373    1042  0006          DJR               /DISABLE JUMP RETURN SAVE
1374    1043  1220          LAM I             /2S COMPLIMENT ADD AC TO LOC CKSUM
1375    1044  0000  CKSUM,  0
1376    1045  0011          CLR               /CLEAR AC
1377    1046  1031          LDA I 11          /INC AND LOAD AC FROM DATA TABLE
1400    1047  0233          XSK I 13          /DONE 400 TIMES
1401    1050  7042          JMP CKSUM-2       /NO-DO IT AGAIN
1402    1051  0011          CLR               /CLEAR AC AND LINK
1403    1052  0017          COM               /COMPLIMENT THE AC
1404    1053  1200          LAM               /2S COMPLIMENT ADD AC
1405    1054  1044          CKSUM             /TO LOC CKSUM
1406    1055  6000          JMP 0             /RET TO MAIN PROG
1407
1410
1411                    /ROUTINE TO RING THE TTY BELL AT END OF TEST
1412
1413    1056  0006  TYPE,   DJR               /DISABLE JUMP RETURN
1414    1057  0500          IOB               /EXECUTE 8 MODE INSTRUCTION
1415    1060  6046          6046              /AC TO TTY AND PUNCH BUFFER AND CLEAR FLAG
1416    1061  0011          CLR
1417    1062  6000          JMP 0             /RETURN TO MAIN PROGRAM
1420
1421
1422                    /16
1423
1424                          EJECT
```

```
1425
1426                    /CLEARS OUT DATA IMAGE AREA (LOCATIONS 1400-1777 INST FIELD 2)
1427                    /BY STORING ZEROS
1430
1431
1432    1063    0071    CLEAR,  SET I 11        /STARTING ADDRESS OF DATA TABLE -1
1433    1064    1377            DATA-1
1434    1065    0073            SET I 13        /EXECUTE 400 TIMES (400 LOC IN DATA TABLE)
1435    1066    7377            -400
1436    1067    0011            CLR             /CLEAR AC
1437    1070    0006    CL,     DJR             /DISABLE JUMP RETURN
1440    1071    1071            STA I 11        /INC AND STORE AC IN DATA TABLE
1441    1072    0233            XSK I 13        /DONE 400 TIMES
1442    1073    7070            JMP CL          /NO-DO IT AGAIN
1443    1074    6000            JMP 0           /RET TO MAIN PROG
1444
1445                    /CLEARS OUT TDATA AREA (LOCATIONS 2000-3777 DATA FIELD 3)
1446                    /BY STORING ZEROS
1447
1450
1451    1075    0072    CLTDAT, SET I 12        /STARTING ADDRESS OF TAPE DATA TABLE
1452    1076    3777            TDATA-1!2000
1453    1077    0073            SET I 13        /EXECUTE 2000 TIMES (2000 LOC IN TAPE DATA TABLE)
1454    1100    6000            -1777
1455    1101    0011            CLR             /CLEAR AC
1456    1102    0006            DJR             /DISABLE JUMP RETURN
1457    1103    1072            STA I 12        /INC AND STORE ZERO AC INTO TAPE DATA TABLE
1460    1104    0233            XSK I 13        /DONE 2000 TIMES
1461    1105    7102            JMP .-3         /NO-DO IT AGAIN
1462    1106    6000            JMP 0           /RET TO MAIN PROG
1463
1464
1465
1466                    /17
1467
1470                            EJECT
-
```

```
1471
1472                    /WRITES DATA IMAGE AREA (LOCATIONS .   /-1777 INST FIELD 2)
1473                    /INTO TDATA AREA (LOCATIONS 2000-3777 DATA FIELD 3)
1474                    /FOR EXECUTION OF WCG INSTRUCTION
1475
1476
1477    1107  0047  DUPDAT, SET 7             /SET UP RET JUMP FROM LOC 7 TO MAIN PROG
1500    1110  0000          0
1501    1111  0070          SET I 10         /DUPLICATE 4 BLOCKS
1502    1112  7773          -4
1503    1113  0072          SET I 12         /STARTING LOC OF TAPE DATA TABLE-1
1504    1114  3777          TDATA-1!2000
1505    1115  0071          SET I 11         /STARTING LOC OF DATA TABLE-1
1506    1116  1377          DATA-1
1507    1117  0073          SET I 13         /EXECUTE 400 TIMES (1 BLK)
1510    1120  7377          -400
1511    1121  1031  DUP,    LDA I 11         /INC AND LOAD AC FROM DATA TABLE
1512    1122  1072          STA I 12         /INC AND STORE IN TAPE DATA TABLE
1513    1123  0233          XSK I 13         /DONE 400 TIMES
1514    1124  7121          JMP DUP          /NO-DO IT AGAIN
1515    1125  0230          XSK I 10         /DONE 4 BLOCKS
1516    1126  7115          JMP DUP-4        /NO-DO IT AGAIN
1517    1127  6007          JMP 7            /RET TO MAIN PROG
1520
1521
1522                    /DELAY ROUTINE PROVIDES A 6 TO 7 MS DELAY
1523                    /ALLOWING TAPES TO ENTER THE TURNAROUND STATE
1524
1525
1526    1130  0011  DELAY,  CLR              /CLEAR THE AC
1527    1131  4015          STC 15
1530    1132  0006          DJR              /DISABLE JUMP RETERN
1531    1133  0235          XSK I 15         /LOC 15 EQUAL 1777
1532    1134  7132          JMP .-2          /NO-DO IT AGAIN
1533    1135  6000          JMP 0            /RETURN TO THE MAIN PROG
1534
1535
1536
1537                    /18
1540
1541                          EJECT
-
```

```
1542
1543                    /COMPARES DATA IMAGE OF DATA WRITTEN ON TAPE (LOCATIONS 1400-1777 INST FIELD 2)
1544                    /WITH DATA READ FROM TAPE AND STORED IN TDATA AREA (LOCATIONS 2000-3777 DATA FIELD 3)
1545                    /CMPR1 ROUTINE IS USED ONLY BY RW2 AND RW12 ROUTINES OF THE MAIN PROGRAM
1546                    /CMPR1 ALTERATION COMPARES 1 BLOCK OF TAPE DATA WITH DATA IMAGE AREA
1547
1550
1551    1136  1020  CMPR1,  LDA I           /SET UP A 1 BLOCK COMPARE
1552    1137  7776           -1
1553    1140  5144           STC CMPR+3      /BY CHANGING CONSTANT IN CMPR+3
1554    1141  0047  CMPR,    SET 7           /SET UP RET JUMP FROM LOC 7 TO MAIN PROGRAM
1555    1142  0000           0
1556    1143  0070           SET I 10        /EXECUTE 4 TIMES UNLESS MODIFIED FOR 1 EXECUTION
1557    1144  7773           -4
1560    1145  0072           SET I 12        /STARTING ADDRESS OF TAPE DATA TABLE-1
1561    1146  3777           TDATA-1!2000
1562    1147  0071           SET I 11        /STARTING ADDRESS OF DATA IMAGE AREA-1
1563    1150  1377           DATA-1
1564    1151  0073           SET I 13        /400 MEMORY LOCATIONS EQUAL ONE TAPE BLK
1565    1152  7377           -400
1566    1153  0011  TST,     CLR
1567    1154  1031           LDA I 11        /INC LOC 11 AND LOAD AC WITH ENTRY FROM DATA IMAGE AREA
1570    1155  1472           SAE I 12        /INC LOC 12 AND COMPARE AC WITH CORRESPONDING ENTRY IN TAPE DATA TABLE
1571    1156  7167           JMP ECMPR       /ERROR-ENTRIES OF BOTH TABLES NOT EQUAL
1572    1157  0233           XSK I 13        /INC LOC 13 HAVE WE DONE IT 400 TIMES
1573    1160  7153           JMP TST         /NO-DO IT AGAIN
1574    1161  0230           XSK I 10        /HAVE WE CHECKED ALL BLOCKS READ IN
1575    1162  7147           JMP TST-4       /NO-DO IT AGAIN
1576    1163  1020           LDA I           /SET LOC CMPR+3 BACK TO ORIGIONAL CONTENTS
1577    1164  7773           -4              /AS IT MAY HAVE BEEN ALTERED
1600    1165  5144           STC CMPR+3
1601    1166  6007           JMP 7           /RETURN TO MAIN PROGRAM
1602
1603
1604                    /COMPARE ERROR ROUTINE
1605                    /EXAMINE LOC 7 TO DETERMINE WHICH ROUTINE IN THE MAIN PROG HAD THE COMPARE ERROR
1606                    /EXAMINE LOC 13 TO DETERMINE WHICH WORD IS INCORRECT
1607                    /EXAMINE LOC 10 TO DETERMINE WHICH BLOCK ERROR IS IN
1610
1611
1612    1167  0460  ECMPR,  SNS I 0          /SNS SW 0 SET
1613    1170  6000           JMP 0           /YES-IGNORE ERROR-RET TO NEXT INST IN NORMAL SEQUENCE
1614    1171  0314           ROR 14          /ROTATE CORRECT DATA INTO MQ
1615    1172  1012           LDA 12          /LOAD THE AC WITH THE INCORRECT DATA READ BACK FROM TAPE
1616    1173  0000           HLT             /HALT-UNTIL KEY CONTINUE IS HIT
1617    1174  1020           LDA I           /SET LOC CMPR+3 BACK TO ORIGIONAL CONTENTS
1620    1175  7773           -4              /AS IT MAY HAVE BEEN ALTERED
1621    1176  5144           STC CMPR+3
1622    1177  2007           ADD 7           /ALTER RETERN JUMP IN LOC 7 BY ADDING
1623    1200  3371           ADD C3          /PLUS 3 AND
1624    1201  6007           JMP 7           /RETURN TO MAIN PROGRAM AND RE-EXECUTE TEST WHICH FAILED
1625
1626
1627
1630                    /19
1631
1632                            EJECT
```

```
1633
1634                    /TESTS REVERSE BLOCK NUMBERS SEQUEN   LLY
1635                    /IF TAPE POSITIONED AT BLOCK 777 THIS TEST
1636                    /CHECKS REVERSE BLOCK NUMBERS 777-0
1637
1640
1641    1202  0047  REVBK,  SET 7            /SET UP RET JUMP FROM LOC 7
1642    1203  0000          0
1643    1204  0733          MTB I U         /MOVE TOWARD BLK 0 (GET TAPE MOVING IN RIGHT DIRECTION)
1644    1205  0000          0
1645    1206  0733          MTB I U         /MOVE TOWARD BLOCK 0
1646    1207  0000          0
1647    1210  1120          ADA I           /ADD 1 TO DIFFERENCE BETWEEN FIRST TAPE BLOCK NUMBER ENCOUNTERED
1650    1211  0001          1               /AND TBLK (DIFFERENCE IS LEFT IN AC AFTER MTB INST)
1651    1212  1060          STA I           /STORE IN LOCATION REVTST
1652    1213  0000  REVTST, 0
1653    1214  0733          MTB I U         /MOVE TOWARD BLK 0
1654    1215  0000          0
1655    1216  1440          SAE             /IS THE DIFFERENCE EQUAL TO THE CONTENTS OF LOC REVTST
1656    1217  1213          REVTST
1657    1220  7224          JMP EREV        /NO-ERROR
1660    1221  0450          AZE             /ARE WE AT BLOCK 0 YET (DIFFERENCE EQUAL TO ZERO)
1661    1222  7210          JMP REVTST-3    /NO-DO IT AGAIN
1662    1223  6007          JMP 7           /RET TO MAIM PROGRAM
1663
1664
1665
1666                    /REVERSE BLOCK NUMBER ERROR
1667                    /THE BLOCK NUMBERS DID NOT OCCURE IN SEQUENCE
1670                    /DIFFERENCE=NUMBER OF BLOCKS BETWEEN PRESENT TAPE POSITION
1671                    /AND THE BLOCK NUMBER YOU ARE MOVING TOWARD
1672                    /MQ = INCORRECT DIFFERENCE
1673                    /AC = CORRECT DIFFERENCE
1674
1675
1676    1224  0460  EREV,   SNS I 0          /SNS SW 0 SET
1677    1225  6000          JMP 0           /YES-IGNORE ERROR AND RET TO MAIN PROG IN NORMAL SEQUENCE
1700    1226  0314          ROR 14          /PUT INCORRECT DIFFERENCE IN MQ REGISTER
1701    1227  1000          LDA             /PUT CORRECT DIFFERENCE IN AC
1702    1230  1213          REVTST
1703    1231  0000          HLT             /HALT-UNTIL KEY CONT IS HIT
1704    1232  6007          JMP 7           /RET TO MAIN PROGRAM AT START OF TEST WHICH FAILED
1705
1706
1707                    /20
1710
1711                          EJECT
-
```

```
1712
1713                    /TESTS FOWARD BLOCK NUMBERS SEQUENTIALLY
1714                    /IF TAPE POSITIONED AT BLOCK 0 THIS TEST
1715                    /CHECKS FOWARD BLOCK NUMBERS 0-777
1716
1717
1720    1233  0047  FWDBK,  SET 7              /SET UP RET JUMP FROM LOC 7
1721    1234  0000          0
1722    1235  0733          MTB I U           /MOVE TOWARD BLOCK 777 (GET TAPE MOVING IN RIGHT DIRECTION)
1723    1236  0777          777
1724    1237  0733          MTB I U           /MOVE TOWARD BLOCK 777
1725    1240  0777          777
1726    1241  1120          ADA I             /SUBTRACT 1 FROM DIFFERENCE BETWEEN
1727    1242  7776          -1                /FIRST TAPE BLK NUMBER ENCOUNTERED AND TBLK
1730    1243  1060          STA I             /STORE IN LOC FWDTST
1731    1244  0000  FWDTST, 0
1732    1245  0733          MTB I U           /MOVE TOWARD BLOCK 777
1733    1246  0777          777
1734    1247  1440          SAE               /IS THE DIFFERENCE EQUAL TO THE CONTENTS OF LOC FWDTST
1735    1250  1244          FWDTST
1736    1251  7262          JMP EFWD          /NO ERROR
1737    1252  1460          SAE I             /IS THE DIFFERENCE EQUAL TO 1
1740    1253  0001          1
1741    1254  7241          JMP FWDTST-3      /NO-DO IT AGAIN
1742    1255  0733          MTB I U           /YES-THIS IS THE LAST MTB AS WE ARE AT BLK 776
1743    1256  0777          777
1744    1257  0450          AZE               /AC ZERO (DIFFERENCE EQUAL TO ZERO)
1745    1260  7262          JMP EFWD          /NO-ERROR
1746    1261  6007          JMP 7             /RETURN TO MAIN PROGRAM
1747
1750
1751
1752                    /FOREWARD BLOCK NUMBER ERROR
1753                    /THE BLOCK NUMBERS DID NOT OCCUR IN SEQUENCE
1754                    /DIFFERENCE=NUMBER OF BLOCKS BETWEEN PRESENT TAPE POSITION
1755                    /AND THE BLOCK NUMBER YOU ARE MOVING TOWARD
1756                    /MQ = INCORRECT DIFFERENCE
1757                    /AC = CORRECT DIFFERENCE
1760
1761
1762
1763    1262  0460  EFWD,   SNS I 0           /SNS SW 0 SET
1764    1263  6000          JMP 0             /YES-IGNORE ERROR AND CONTINUE
1765    1264  0314          ROR 14            /ROTATE INCORRECT DIFFERENCE INTO MQ REGISTER
1766    1265  1000          LDA               /LOAD THE AC WITH THE DIFFERENCE
1767    1266  1244          FWDTST
1770    1267  0000          HLT               /HALT-UNTIL KEY CONTINUE HIT
1771    1270  6007          JMP 7             /RET TO MAIN PROG AT START OF TEST WHICH FAILED
1772
1773
1774                    /21
1775
1776                    EJECT
-
```

```
1777
2000
2001                    /ERROR HALTS FROM MAIN PROGRAM
2002                    /SENSE SWITCH Ø IGNORS ERROR AND
2003                    /RETURNS YOU TO PROGRAM SEQUENCE
2004                    /LOCATION OF NEXT EXECUTABLE INSTRUCTION
2005                    /KEY CONTINUE RETURNS YOU TO THE MAIN PROGRAM
2006                    /AT THE START OF THE TEST WHICH FAILED
2007                    /E2 CORRESPONDS TO TEST T2  ETC
2010
2011
2012    1271  0461  E2,     SNS I 1         /IS SNS Ø SET
2013    1272  6000          JMP Ø           /YES-IGNORE ERROR RET TO NEXT INST IN PROG SEQUENCE
2014    1273  0314          ROR 14          /ROTATE THE BAD CHECKSUM INTO THE MQ REGISTER
2015    1274  1000          LDA             /PUT THE CALCULATED CHECKSUM INTO THE AC REGISTER
2016    1275  1044          CKSUM
2017    1276  0000          HLT             /HALT-UNTIL KEY CONTINUE IS HIT
2020    1277  6026          JMP T2          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2021
2022    1300  0460  E3,     SNS I           /IS SENSE SW 1 SET
2023    1301  6000          JMP Ø           /YES-IGNORE ERROR RET TO NEXT INST IN PROG SEQUENCE
2024    1302  0000          HLT             /HALT-UNTIL KEY CONTINUE IS HIT
2025    1303  6051          JMP T3          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2026
2027    1304  0460  E4,     SNS I Ø         /IS SNS SW Ø SET
2030    1305  6000          JMP Ø           /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQUENCE
2031    1306  0314          ROR 14          /ROTATE THE BAD CHECKSUM INTO THE MQ REGISTER
2032    1307  1000          LDA             /LOAD THE AC WITH THE CALCULATED CHECKSUM
2033    1310  1044          CKSUM
2034    1311  0000          HLT             /HALT-UNTIL KEY CONTINUE IS HIT
2035    1312  6100          JMP T4          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2036
2037    1313  0460  E5,     SNS I Ø         /IS SNS SW Ø SET
2040    1314  6000          JMP Ø           /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQ
2041    1315  0000          HLT             /HALT-UNTIL KEY CONTINUE IS HIT
2042    1316  6111          JMP T5          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2043
2044    1317  0460  E6,     SNS I Ø         /IS SNS SW Ø SET
2045    1320  6000          JMP Ø           /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQ
2046    1321  0314          ROR 14          /ROTATE THE BAD CHECKSUM INTO THE MQ REG
2047    1322  1000          LDA             /LOAD THE AC WITH THE CALCULATED CHECKSUM
2050    1323  1044          CKSUM
2051    1324  0000          HLT             /HALT-UNTIL KEY CONTINUE IS HIT
2052    1325  6131          JMP T6          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2053
2054    1326  0460  E7,     SNS I Ø         /IS SNS SW Ø SET
2055    1327  6000          JMP Ø           /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQUENCE
2056    1330  0000          HLT             /HALT-UNTIL KEY CONTINUE IS HIT
2057    1331  6163          JMP T7          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2060
2061
2062                  /22
2063
2064                          EJECT
-
```

```
2065
2066      1332  0460  E20,    SNS I Ø          /IS SNS SW Ø SET
2067      1333  6000          JMP Ø            /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQUENCE
2070      1334  0314          ROR 14           /ROTATE THE BAD CHECKSUM INTO THE MQ REGISTER
2071      1335  1000          LDA              /LOAD THE AC WITH THE CALCULATED CHECKSUM
2072      1336  1044          CKSUM
2073      1337  0000          HLT              /HALT-UNTIL KEY CONTINUE IS HIT
2074      1340  6471          JMP T20          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2075
2076      1341  0460  E20A,   SNS I Ø          /IS SNS SW Ø SET
2077      1342  6000          JMP Ø            /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQ
2100      1343  0000          HLT              /HALT-UNTIL KEY CONTINUE IS HIT
2101      1344  6471          JMP T20          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2102
2103      1345  0460  E21A,   SNS I Ø          /IS SNS SW Ø SET
2104      1346  6000          JMP Ø            /YES-IGNORE ERROR AND RETURN TO NEXT INST IN PROG SEQ
2105      1347  0314          ROR 14           /ROTATE THE BAD CHECKSUM INTO THE MQ REGISTER
2106      1350  1000          LDA              /LOAD THE AC WITH THE CALCULATED CHECKSUM
2107      1351  1044          CKSUM
2110      1352  0000          HLT              /HALT-UNTIL KEY CONTINUE IS HIT
2111      1353  6524          JMP T21          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2112
2113      1354  0460  E23,    SNS I Ø          /IS SNS SW Ø SET
2114      1355  6000          JMP Ø            /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQ
2115      1356  0314          ROR 14           /ROTATE THE BAD CHECKSUM INTO THE MQ REGISTER
2116      1357  1000          LDA              /LOAD THE AC WITH THE CALCULATED CHECKSUM
2117      1360  1044          CKSUM
2120      1361  0000          HLT              /HALT-UNTIL KEY CONTINUE IS HIT
2121      1362  6702          JMP T23          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2122
2123      1363  0460  E24,    SNS I Ø          /IS SNS SW Ø SET
2124      1364  6000          JMP Ø            /YES-IGNORE ERROR AND RET TO NEXT INST IN PROG SEQUENCE
2125      1365  0000          HLT              /HALT-UNTIL KEY CONTINUE IS HIT
2126      1366  6725          JMP T24          /RET TO MAIN PROG AT START OF TEST WHICH FAILED
2127
2130
2131                /23
2132
2133                          EJECT
```

```
CKSU     0044
CL       0070
CLEAR    5063
CLTDAT   5075
CMPR     5141
CMPR1    5136
C1       5367
C1M      5370
C1001    5372
C3       5371
C3374    5373
C4374    5374
DATA     1400
DATSET   4752
DELAY    5130
DONE     4570
DUP      5121
DUPDAT   5107
ECMPR    5167
EFWD     5262
EREV     5224
E2       5271
E20      5332
E20A     5341
E21A     5345
E23      5354
E24      5363
E3       5300
E4       5304
E5       5313
E6       5317
E7       5326
FWDBK    5233
FWDTST   5244
FXDAT    4760
FXLOAD   4764
GCKSUM   5033
LSCOPE   4676
RADD     5026
RDAT     4776
REVBK    5202
REVTST   5213
RLOAD    5012
RNA      5024
RNB      5025
-
```

```
RWSUB   4631
RW1     4022
RW10    4460
RW11    4467
RW12    4576
RW2     4076
RW3     4124
RW4     4216
RW5     4262
RW6     4326
RW7     4377
START   4020
TBLK    0374
TDATA   2000
TST     5153
TYPE    5056
T10     4221
T11     4240
T12     4263
T13     4303
T14     4332
T15     4353
T16     4406
T17     4463
T2      4026
T20     4471
T21     4524
T23     4702
T24     4725
T3      4051
T4      4100
T5      4111
T6      4131
T7      4163
XOA     0021
X10     4227
X11     4246
X12     4271
X13     4311
X14     4340
X15     4361
X2      4034
X20     4477
X20A    4514
X21     4534
X21A    4546
X22     4657
X22A    4661
X22B    4664
X22C    4670
X23     4710
X24     4733
X3      4057
X6      4137
X7      4171
```

0000 RRORS

```
2134
2135
2136                    /CONSTANTS USED IN PROGRAM
2137
2140
2141        1367   0001   C1,      1
2142        1370   7776   C1M,     7776
2143        1371   0003   C3,      3
2144        1372   1001   C1001,   1001
2145        1373   3374   C3374,   3374
2146        1374   4374   C4374,   4374
2147                      DATA=1400
2150                      TDATA=2000
2151                      TBLK=374
2152                      XOA=21
```