IDENTIFICATION

PRODUCT CODE: MAINDEC-12-D3DB-D

PRODUCT NAME: PDP-12 TAPE DATA EXERCISER

DATE: FEB, 1,1971

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: RAYMOND SHOOP

Tape address:

TC12DAEX

1. ABSTRACT

THE PDP-12 TAPE EXERCISER PROGRAM IS A DYNAMIC TEST OF THE LINC-TAPE CONTROL AND TAPE TRANSPORTS. IT MAY BE USED TO TEST A CONTROLLER WITH FROM 1 TO 8 TAPE TRANSPORT UNITS, AND A PDP-12 WITH UP TO 32-K OF MEMORY.

2. MACHINE REQUIREMENTS

A. A STANDARD PDP-12A OR B COMPUTER.
B. PDP-12 LINC-TAPE CONTROLLER.
C. A LINC-TAPE TRANSPORT
D. AN ASR-33 TELETYPE OR EQUIVALENT

2.1 STORAGE

THIS PROGRAM MAY ONLY BE RUN IN MEMORY FIELD 0 AND OCCUPIES VIRTUALLY ALL OF THE LOWER HALF OF FIELD 0. LOCATIONS 0-3377 INCLUSIVE. LOCATIONS 3400-7777 ARE USED FOR INPUT-OUTPUT BUFFERS.

2.2 PRELIMINARY PROGRAMS

ALL PDP-8 AND LINC-MODE BASIC INSTRUCTION DIAGNOSTIC AND EXERCISERS INCLUDING TAPE CONTROL TEST MUST HAVE SUCCESSFULLY RUN PRIOR TO RUNNING TAPE EXERCISER TEST.

2.3 LOADING PROCEDURE

3.1 METHOD

THIS PROGRAM CAN BE LOADED INTO MEMORY WITH THE BINARY LOADER. IT MAY ALSO BE LOADED INTO MEMORY BY USING LAP6-DIAL.

4. STARTING PROCEDURE

THE PROCEDURE TO SETUP THE TAPE PROCESSOR FOR DIAGNOSIS IS CRITICAL, ANY ERROR IN THE STARTING PROCEDURE MAY RESULT IN AN ERROR.

A. TAPE TRANSPORT

1. MOUNT A CERTIFIED PDP-12 TAPE (WHICH HAS BEEN MARKED WITH "MARK 1600") ON ALL DRIVES TO BE TESTED.
2. SET THE UNIT SELECTOR SWITCH ON EACH TRANSPORT TO AN INCREMENTING NUMBER STARTING WITH UNIT 0.
3. SET THE LOCAL/REMOTE SWITCH TO REMOTE ON EACH DRIVE.
4. SET WRITE ENABLE SWITCHES ON EACH DRIVE.

B. SCOPE (VR 14)

1. PLACE CHANNEL SELECTOR TO 1 & 2.

C. DATA TERMINAL PANEL

1. ROTATE ANALOG CHANNEL 0 TO 4 COUNTER-CLOCKWISE TO THE END OF ROTATION. THESE ARE USED ONLY TO CONTROL THE POSITION OF THE DISPLAY.

D. COMPUTER

1. SET THE LEFT SWITCHES TO 0200.
2. SET THE RIGHT SWITCHES TO X0XX. (REFER TO SECTION 4.1)
3. SET THE MODE SWITCH TO LINC-MODE.
4. DEPRESS I/O PRESET.
5. DEPRESS START LEFT SWITCHES (LS).

THE PROGRAM IS NOW RUNNING, TAPE UNIT 0 SHOULD START MOVING IN THE REVERSE DIRECTION. WHEN THE COMPUTER IS TRANSFERRING DATA IN NO-PAUSE MODE, THE PDP-12 MAINDEC NUMBER (D3D0) WILL BE DISPLAYED ON THE DISPLAY SCREEN.

4.1 CONTROL SWITCH SETTINGS

A. RIGHT SWITCHES

RSW 0=1 DELETE RECOVERABLE ERROR HALTS, AND RESTART CURRENT PASS.
RSW 1=1 DELETE ERROR MESSAGE.
RSW 6=8= NUMBER OF EXTRA TAPE TRANSPORT UNITS.
RSW 9=11=NUMBER OF EXTRA 4K MEMORY FIELDS.

4.2 STARTING ADDRESS

LINC=MODE 0200

```
200     LINC=MODE  MOVE TOWARD BLOCK (MTB) TEST.  UPON COMPLETION OF
                   THIS TEST ON UNIT 0, EXIT TO THE DATA TEST.
201     LINC=MODE  DATA TEST ENTRY ADDRESS.
```

ONLY THESE TWO ADDRESSES ARE VALID STARTING ADDRESSES FOR THIS PROGRAM.

5. ERRORS

THE ERROR TYPE-OUT MESSAGE IS THE VALUE OF THE PROGRAM COUNTER ERROR LOCATION. THIS LISTING MUST BE CONSULTED TO FIND THE TYPE OF ERROR (I.E. ER1).
THE ERRORS ARE:

ER 1: SKIP ON TAPE DONE FAILED.
ER 2: TAC IN ERROR, AC CONTAINS THE BAD VALUE OF THE TAC.
ER 3: BAD SEARCH, AC CONTAINS THE BAD VALUE OF THE TAC.
ER 4: TAPE INTERRUPT FAILED TO CAUSE AN INTERRUPT.
ER 5: UNEXPECTED INTERRUPT, FROM AN UNWANTED SOURCE.
ER 6: MOTION ERROR.
ER 7: DATA ERROR, NON-GROUP TAPE INSTRUCTION.
ER 8: DATA ERROR, A GROUP TAPE INSTRUCTION.

WHEN A DATA ERROR IS DETECTED, LOCATIONS 3400-3777 CONTAIN THE EXPECTED DATA AND THE VALUE OF LOCATION 0015 CONTAINS THE ADDRESS OF THE DATA IN ERROR. REFER TO 10. FOR ERROR DESCRIPTIONS.

6. RESTRICTIONS

A. PROGRAM MUST BE EXECUTED IN FIELD 0.
B. STANDARD PDP-12 A OR B.
C. TAPE TRANSPORTS MUST BE SELECTED SEQUENCIALLY, STARTING WITH UNIT 0, WRITE ENABLED AND REMOTE.
D. THE RIGHT SWITCHES SET TO ONLY EXISTING TRANSPORTS AND/OR MEMORY AVAILABLE.
E. NO DEVICE WHICH CAUSES UNEXPECTED INTERRUPTS.
F. THE DATA IN BLOCKS 770 TO 1007 WILL BE DESTROYED ON ALL TRANSPORTS USED.

7. EXECUTION TIME

THE EXECUTION TIME IS VARIABLE TO THE NUMBER OF TRANSPORTS AND AMOUNT OF EXTRA MEMORY. THE MINIMUM AMOUNT OF TIME SHOULD BE CONSIDERED 15 MINUTES PER TRANSPORT.

bell rings @ 12 minute intervals using 2 transports + 8K memory

8. ERROR EXAMPLE:

PC XXXX-REFER TO LOCATION XXXX IN THE LISTING TO FIND THE TYPE OF ERROR ENCOUNTERED.

9. OVERNIGHT RUNS:

IF RSW 00 IS SET TO A ONE, THE TEST WILL TYPE OUT ANY RECOVERABLE ERROR CONDITION ENCOUNTERED AND RESTART THE CURRENT PASS. THIS IS DUE TO THE FACT THAT SEARCH AND DATA ERRORS ARE IN GENERAL NONRECOVERABLE.

10. ERROR DEFINITIONS

A. ERROR 1 SKIP ON TAPE FAILED. EXECUTED A MTB TO BLOCK 0000 IN PAUSE MODE. THE PROCESSOR WILL WAIT WHILE THE TAPE IS IN MOTION. AT THE COMPLETION OF THE INSTRUCTION THE TAPE DONE FLAG SHOULD BE SET. THE PROCESSOR DID NOT DETECT THIS FLAG AND HALTED.

B. ERROR 2 TAC IN ERROR AFTER A TAPE INSTRUCTION EXCEPT A WRI. EXECUTED A TAPE INSTRUCTION, AT IT'S COMPLETION THE TAC SHOULD CONTAIN THE VALUE 7777. THE AC CONTAINS THE VALUE READ FROM THE TAC IN ERROR.

C. ERROR 3 SEARCH ERROR OBTAIN A BLOCK NUMBER FROM A RANDOM NUMBER GENERATOR AND EXECUTE A MOVE TOWARD THAT BLOCK. DURING THE EXECUTION OF THE MTB, EACH BLOCK IS TESTED FOR PROPER SEQUENCE AND ABSOLUTE VALUE.

LOC. 0122 EXECUTED A MTB TO BLOCK 0 IN PAUSE MODE. THE AC CONTAINS THE BLOCK NUMBER READ AND LOC. 0031 CONTAINS THE EXPECTED BLOCK NUMBER.

LOC. 0150 EXECUTED A MTB TO BLOCK 777 IN NO PAUSE MODE. THE AC CONTAINS THE BLOCK NUMBER READ AND LOC. 0031 CONTAINS THE EXPECTED BLOCK NUMBER.

LOC. 1077 EXECUTED A MTB TO A BLOCK NUMBER (LOC. 1573). THE AC CONTAINS THE BLOCK NUMBER READ AND LOC. 1076 CONTAINS THE EXPECTED BLOCK NUMBER.

D. ERROR 4 TAPE INTERRUPT FAILED. AC CONTAINS THE TAPE INTERRUPT BIT AT THE XOBWD, (0100). THE PROGRAM WAITED FOR A TAPE DONE FLAG. AFTER DETECTING TAPE DONE, AN INTERRUPT SHOULD HAVE OCCURRED BUT IT DID NOT. (I.E. FALSE TAPE DONE, TAPE INTERRUPT FLIP-FLOP NOT SET)

E. ERROR 5 UNEXPECTED INTERRUPT

*0002 OBTAINED AN 8 MODE INTERRUPT, NO SUCH INTERRUPT IS LEGAL

*0041 LINC MODE INTERRUPT
THE PROGRAM DID NOT EXPECT A PROGRAM INTERRUPT, THE XOB WORD (BIT 5) WAS 0 THEREFORE NO INTERRUPT WAS EXPECTED.

*0046 LINC MODE INTERRUPT
THE PROGRAM DID EXPECT A PROGRAM INTERRUPT FROM THE TAPE CONTROL BUT IT WAS NOT FROM THE TAPE DONE FLAG. SKIP ON TAPE DONE MAY HAVE FAILED.

F. ERROR 6 MOTION ERROR EXECUTED A TAPE INSTRUCTION. WHEN COMPLETED, A TEST OF THE STATE OF THE MOTION FLIP-FLOP WAS MADE. SET THE LINK TO THE EXPECTED STATE OF THE MOTION FLIP-FLOP. IF THE LINK=0 THE MOTION SHOULD BE A 0. IF THE LINK=1 THE MOTION SHOULD EQUAL A 1. AC WILL CONTAIN EITHER A 10 OR A 0.

G. ERROR 7 DATA ERROR - RDC, RDE EXECUTED A READ OR READ AND CHECK INTO MEMORY

1. THE DATA FIELD REGISTER CONTAINS THE MEMORY FIELD IN ERROR.

2. LOCATION 0015 IS A 10 BIT ADDRESS OF THE BAD DATA LOCATION (REFER TO SECTION 1).

3. LOCATION 0016 IS A 10 BIT ADDRESS IN LDF1 WHERE THE GOOD DATA IS STORED (3400 THRU 3777 CONTAINS THE GOOD DATA).

4. THE AC CONTAINS THE GOOD DATA PATTERN. (REFER TO 11. FOR PATTERNS WRITTEN ON TAPE)

5. THE LOCATION MTINST + 1 (#1573) CONTAINS THE BLOCK NUMBER ON THE TAPE IN ERROR.

H. ERROR = 8 DATA ERROR - RCG EXECUTED A READ AND CHECK GROUP (RCG)

1. THE DATA FIELD REGISTER CONTAINS THE MEMORY FIELD IN ERROR.

2. LOCATION 0015 IS A 10 BIT ADDRESS OF THE BAD DATA LOCATION (REFER TO SECTION 1).

3. LOCATION 0016 IS A 10 BIT ADDRESS IN LDF 1 WHERE THE GOOD DATA IS STORED (3400-3777 CONTAINS THE GOOD DATA).

4. THE AC CONTAINS THE GOOD DATA PATTERN.

5. THE LOCATION MTINST +1 (#1573) CONTAINS THE GROUP COUNT AND THE BLOCK NUMBER ON THE TAPE IN ERROR.

I. TO DETERINE THE MEMORY ADDRESS OF A DATA ERROR AND IT'S VALUE AFTER THE MACHINE HAS COMPLETED TYPING THE ERROR REPORT.

A. THE GOOD DATA IS IN THE AC.

B. EXAMINE ABSOLUTE LOCATION 0015.

C. SET THE LEFT SWITCH BITS 2-11 EQUAL TO THE VALUE OF LOCATION 0015 BITS 2-11.

D. SET LEFT SWITCH BITS 0-1 EQUAL TO THAT OF BITS 3-4 OF THE DATA FIELD LIGHTS.

E. SET THE INST. FIELD SWITCHES EQUAL TO THAT OF BITS 0-2 OF THE DATA FIELD LIGHTS.

F. DEPRESS EXAM.

G. THE BAD DATA WILL NOW APPEAR IN THE MEMORY BUFFER.

11. DATA PATTERNS

A. 0000
B. 7777
C. 0000 AND 7777
D. 7777 AND 0000
E. 7070
F. 0707
G. 7070 AND 0707
H. 0707 AND 7070
I. 5252
J. 2525
K. 5252 AND 2525
L. 2645 AND 5132
M. COUNT PATTERN

APPENDIX A

PDP-8 MODE PERFORATED - TAPE LOADER

READIN MODE LOADER

THE READIN MODE (RIM) LOADER IS A MINIMUM LENGTH, BASIC, PERFORATED-TAPE PROGRAM FOR THE 33 ASR. IT IS INITIALLY STORED IN MEMORY BY MANUAL USE OF THE OPERATOR CONSOLE KEYS AND SWITCHES. THE LOADER IS PERMANENTLY STORED IN 18 LOCATIONS OF PAGE 37.

THE RIM LOADER CAN ONLY BE USED IN CONJUNCTION WITH THE 33 ASR READER (NOT THE HIGH-SPEED PERFORATED-TAPE READER). BECAUSE A TAPE IN RIM FORMAT IS, IN EFFECT, TWICE AS LONG AS IT NEED BE, IT IS SUGGESTED THAT THE RIM LOADER BE USED ONLY TO READ THE BINARY LOADER WHEN USING THE 33 ASR. (NOTE: SOME PDP-12 DIAGNOSTIC PROGRAM TAPES ARE IN RIM FORMAT).

THE COMPLETE PDP-12 RIM LOADER (SA = 7756 IS AS FOLLOWS:)

ABSOLUTE ADDRESS	OCTAL CONTENT	TAG	INSTRUCTION IS	COMMENTS
7756	6032	BEG	KCC	/CLEAR AC AND FLAG
7757	6031		KSF	/SKIP IF FLAG = 1
7760	5357		JMP-1	/LOOKING FOR CHARACTER
7761	6036		KRB	/READ BUFFER
7762	7106		CLL RTL	
7763	7006		RTL	/CHANNEL 8 IN AC0
7764	7510		SPA	/CHECKING FOR LEADER
7765	5357		JMP BEG-1	/FOUND LEADER
7766	7006		RTL	/OK, CHANNEL 7 IN LINK
7767	6031		KSF	
7770	5367		JMP-1	
7771	6034		KRS	/READ, DO NOT CLEAR
7772	7420		SNL	/CHECKING FOR ADDRESS
7773	3776		DCA I TEMP	/STORE CONTENT
7774	3376		DCA TEMP	/STORE ADDRESS
7775	5356		JMP BEG	/NEXT WORD
7776	-	TEMP	0	/TEMP STORAGE
7777	5XXX		JMP X	/JMP START OF BIN LOADER

PLACING THE RIM LOADER IN CORE MEMORY BY WAY OF THE OPERATOR CONSOLE KEYS AND SWITCHES IS ACCOMPLISHED AS FOLLOWS:

A. SET THE STARTING ADDRESS 7756 IN THE LEFT SWITCHES.

B. SET THE FIRST INSTRUCTION (6032) IN THE RIGHT SWITCHES.

C. PRESS THE FILL SWITCH.

D. SET THE NEXT INSTRUCTION (6031) IN THE RIGHT SWITCHES.

E. PRESS THE FILL STEP SWITCH.

F. REPEAT STEPS D AND E UNTIL ALL 16 INSTRUCTIONS HAVE BEEN DEPOSITED.

TO LEAD A TAPE IN RIM FORMAT, PLACE THE TAPE IN THE READER, SET THE LEFT SWITCHES TO THE STARTING ADDRESS 7756 OF THE RIM LOADER (NOT OF THE PROGRAM BEING READ), PRESS THE START LS KEY, AND START THE TELETYPE READER.

APPENDIX B

PDP-12 CONTROL WORD FORMAT

WD1 LOCATION 0021

0	NOT USED
1-2	EXTENDED UNIT GROUP
3	EXTENDED ADDRESS OPERATION
4	NOT USED
5	TAPE INTERRUPT (ONLY IF 6=0)
6	PAUSE
7	"I" BIT
8	"U" BIT
9-11	TAPE INSTRUCTION FUNCTION

WD2 LOCATION 0022

0-6	NOT USED
7-9	EXTENDED MEMORY FIELDS
10-11	LINC MEMORY FIELDS

WD3 LOCATION 0023

0	NOT USED
1-3	QUARTER NUMBER
4-7	NOT USED
8-11	BLOCK NUMBER (ADD 770)

WD4 LOCATION 0024

0-11	EXTENDED ADDRESS (USED IN XA MODE ONLY)

XOB LOCATION 0026

0-2	EXTENDED MEMORY BITS
3-4	NOT USED
5	ENABLE TAPE INTERRUPTS
6	MAINT. MODE
7	ENABLE EXTENDED ADDRESS MODE
8	DO NOT PAUSE
9	HOLD UNIT MOTION
10-11	EXTENDED UNIT GROUP

```
                    /PDP-12 TAPE DATA EXERCISER MAINDEC-12-D3DB
                    /COPYRIGHT 1971, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.


                    /STARTING ADDRESSES

                    / 200 LINC-MODE,                    MOVE TOWARD BLOCK TEST
                    / 201 LINC-MODE,                    DATA TEST


                    /RSW    0=1                         DELETE RECOVERABLE ERROR HALT, RESTART CURRENT PASS.
                    /RSW    1=1                         DELETE ERROR MESSAGE

                    /RSW    6-8=                        NUMBER OF EXTRA TRANSPORTS
                    /                                   GREATER THAN 0

                    /RSW    9-11=                       NUMBER OF EXTRA MEMORY BANKS
                    /                                   GREATER THAN 0



            0001    .1
0001    0000            LHLT                            /*** ER 5 *** IN 8 MODE
0002    7402            HLT                             /*** ER 5 *** IN 8 MODE
0003    0002    K0002,  0002
0004    0000    XXXAC,  0000                            /TYPE OUT POINTER
            0020    .20

                    /STORAGE AREA FOR SOME COMMONLY USED VARIABLES
0020    0000    MASTER, 0                               /MASTER WORD
0021    0000    WD1,    0                               /WORD1
0022    0000    WD2,    0                               /WORD2
0023    0000    WD3,    0                               /WORD3
0024    0000    WD4,    0                               /WORD4
0025    0000    UNIT,   0                               /UNIT BITS (IN 6,7,8)
0026    0000    XOBWD,  0                               /EXTENDED OPERATIONS BUFFER WORD
0027    0000    FIELDN, 0                               /FIELD NUMBER (EITHER 3 BITS OR 5)
0030    0000    AC,     0                               /AC
0031    0000    STAC,   0                               /SAVED TAPE AC
0032    0000    QNBN,   0                               /QUARTER NUMBER, BLOCK NUMBER SAVE
0033    0000    CTEM1,  0                               /QN BITS
0034    0000    CTEM3,  0                               /BN 3-11 BITS
0035    0000    CSTART, 0                               /STARTING ADDRESS OF "LITTLE PROGRAM"
0036    0100    K0100,  100
0037    0200    K0200,  200
```

```
                    /LINC INTERRUPT HANDLER

          0040      *40
0040      0000      LINTER, 0
0041      0016              LNOP                    /AN LNOP MAYBE AN
                                                    / LJMP XXX FOR INTERRUPT
                                                    / HANDLING ROUTINE
0042      0002      TSTMOR, PDP                     /CHANGE TO PDP-8 MODE
0043      7200              CLA
0044      1036              TAD     K0100
0045      6151              6151                    /SKIP IF TAPE DONE SET
0046      7402              HLT                     / *** ER 5 ***
0047      7200              CLA
0050      1037              TAD     K0200
0051      6151              6151                    /CLEAR TAPE DONE
0052      7200              CLA
0053      1036              TAD     K0100
0054      6151              6151                    /DID TAPE DONE CLEAR?
0055      7410              SKP                     /YES
0056      7402              HLT                     /NO,TAPE DONE DID NOT CLEAR
0057      7200              CLA
0060      6141              LINC                    /CHANGE BACK TO LINC MODE
0061      6061      MAGTAP, LJMP    .               /EXIT
```

```
                        /CHECK THE INSTRUCTION MTB
                        /START TAPE MOVING TOWARD BLOCK 000

0062  7011    MTBTST, CLR
0063  0001            AXO                     /CLEAR XOB
0064  0723    BKWRD,  MTB+20                  /MOVE TOWARD BLOCK 000, DON'T STOP
0065  0000            0                       /PROCESSOR WILL PAUSE UNTIL A
0066  0416            STD                     /BLOCK NUMBER IS FOUND
0067  7733            LJMP    XXX             /ERROR, TAPE DONE FLAG IS NOT SET *** ER 1 ***
0070  4030            STC     AC              /STORE AC
0071  0003            TAC                     /READ TAPE AC
0072  1040            STA                     /SAVE
0073  0031            STAC
0074  1440            SAE                     /TAC=AC?
0075  0030            AC
0076  7733            LJMP    XXX             /NO, TAC DOES NOT EQUAL AC, ERROR *** ER 2 ***
0077  0451            APO                     /SKIP IF AC POSITIVE (SHOULD BE MINUS OR 0)
0100  6104            LJMP    LOOP01          /OK SO FAR
0101  0450            AZE                     /IS AC=0?
0102  7733            LJMP    XXX             /NO, AC GREATER THAN 0 *** ER 2 ***
0103  6124            LJMP    FOBWRD          /FOUND BLOCK 0, GO ON TO SOMETHING ELSE
0104  1020    LOOP01, LDA+20                  /LOAD AC
0105  0001            1                       /WITH 1
0106  1140            ADM                     /ADD 1 TO TAC
0107  0031            STAC
0110  1460            SAE+20                  /SKIP IF 1
0111  0001            1
0112  0456            LSKP                    /NOT ONE
0113  6124            LJMP    FOBWRD          /GO TO FORWARD TEST
0114  0723            MTB+20                  /MOVE TOWARD 0 (DON'T STOP)
0115  0000            0
0116  0601            LIF     1
0117  7066            LJMP    TTDF            /TEST THE DONE FLAG
0120  1440            SAE                     /COMPARE EXPECTED DISTANCE TO 0
0121  0031            STAC
0122  7733            LJMP    XXX             /ERROR, BAD "SEARCH" COMPUTATION *** ER 3 ***
0123  6104            LJMP    LOOP01
```

```
                    /AFTER FINDING BLOCK 000, "SEARCH" FOR BLOCK 777 - FORWARD

0124  0723     FOBWRD, MTB-20                  /MOVE TOWARD BLOCK 0010
0125  0010             0010
0126  0450             AZE
0127  6124             LJMP    FOBWRD          /WAIT UNTIL IT IS FOUND
0130  1020     FORWRD, LDA-20
0131  0766             0766
0132  4031             STC     STAC            /STORE IN EXPECTED TAPE BLOCK
0133  1020             LDA-20                  /LOAD AC WITH 10
0134  0010             10
0135  0001             AXO                     /SET XOB FOR NO-PAUSE
0136  0011             CLR
0137  0723             MTB-20                  /MOVE TOWARD BLOCK 777
0140  0777             777
0141  0453             IBZ                     /WAIT FOR INTERBLOCK ZONE
0142  6141             LJMP    .-1
0143  0416             STD                     /WAIT FOR TAPE DONE FLAG
0144  6143             LJMP    .-1
0145  0003             TAC                     /READ TAPE AC
0146  1440             SAE                     /CORRECT NUMBER?
0147  0031             STAC
0150  7733             LJMP    XXX             /NO ***ER 3***
0151  0470             AZE-20                  /FOUND BLOCK 777?

0152  6155             LJMP    .-3             /YES
0153  7760             LJMP    SUBT1           /NO, SUBTRACT 1 FROM NUMBER EXPECTED
0154  6132             LJMP    FORWRD-2        /GO BACK AND DO IT AGAIN
0155  1020             LDA-20                  /SET UP TO TEST TAPE DONE
0156  6202             LJMP    DATUM
0157  4061             STC     MAGTAP          /SET UP RETURN ADDRESS
0160  6042             LJMP    TSTMOR          /TEST TAPE DONE

                    /CLEAR THE INPUT BUFFER IN FIELD 0

0161  0002     CLEAR,  PDP
0162  7300             CLA CLL
0163  1175             TAD     K4001           /SET UP A COUNTER
0164  3016             DCA     16              / LOCATION
0165  1174             TAD     M4000           /SET UP A POINTER
0166  3017             DCA     17              / LOCATION
0167  3417             DCA I   17              /
0170  2016             ISZ     16              /DONE ?
0171  5167             JMP     .-2             /NO, MORE TO DO
0172  6141             LINC
0173  6000             LJMP    0               /NOW GO DO SOMETHING

0174  4000     M4000,  -4000
0175  4001     K4001,  4001
      0200     *200
0200  6062             LJMP    MTBTST
0201  6161             LJMP    CLEAR
```

```
                /THIS SECTION BEGINS THE DATA TEST PORTION
                /OF THE PROGRAM
                /THE TEST IS BUILT AROUND 4 PARAMETER WORDS
                /THE FIRST WORD IS THE MAG TAPE "COMMAND" WORD
                /IT DEFINES THE INSTRUCTION, U, I, PAUSE, TAPE INTERRUPT
                /(ONLY IF PAUSE IS TRUE), EXTENDED OPERATION, AND EXTENDED UNITS
                /THE SECOND WORD DEFINES THE MEMORY FIELD (EITHER LINC OR 8)
                /THE THIRD WORD DEFINES QUARTER NUMBER AND BLOCK NUMBER
                /THE FOURTH WORD DEFINES THE EXTENDED ADDRESS
                /NOT ALL WORDS, OR ALL BITS OF A WORD ARE NECESSARILY USED

0202  0011    DATUM:  CLR
0203  4020            STC     MASTER          /INITIALIZE MASTER WORD TO 0
0204  0641    RESTAR, LDF 1
0205  1020            LDA+20                  /INITIALIZE
0206  6303            LJMP    PAT1            /PATTERN
0207  1040            STA                     /ROUTINE
0210  2257            PATPNT
0211  0011            CLR
0212  0066            SET+20  6               /CLEAR OUT BLOCK PATTERN TABLE
0213  3177            BLKTBL-1
0214  0067            SET+20  7
0215  7577            7577
0216  1066            STA+20  6
0217  0227            XSK+20  7
0220  6216            LJMP    .-2
0221  4021    DATLUP, STC     WD1             /SET UP WORD 1
0222  7637            LJMP    RANDOM
0223  4022            STC     WD2             /FORM WORD 2
0224  7637            LJMP    RANDOM
0225  4023            STC     WD3             /WORD 3
0226  7637            LJMP    RANDOM
0227  4024            STC     WD4             /AND WORD 4
```

```
                        /THIS SECTION OF CODING TAKES CARE OF THE EXTENDED UNITS (MORE THAN 1)
0230  1000     EXTUNT, LDA                     /GET WORD 1
0231  0021             WD1
0232  1560             BCL+20                  /MASK TO EXTENDED UNIT
0233  4777             4777
0234  0305             ROR     5               /POSITION TO NEXT TO "U" BIT
0235  4025             STC     UNIT
0236  2021             ADD     WD1             /GET WD1
0237  1560             BCL+20                  /MASK TO BIT 7
0240  7767             7767
0241  2025             ADD     UNIT            /ADD TO CURRENT UNIT
0242  4025             STC     UNIT            /RESTORE NEW UNIT
0243  0516             RSW                     /READ THE RIGHT SWITCHES
0244  1560             BCL+20                  /CLEAR ALL BUT UNITS BITS
0245  7707             7707
0246  0017             COM                     /COMPLEMENT
0247  2025             ADD     UNIT            /ADD CURRENT UNIT NUMBER
0250  0471             APO+20                  /AC MINUS
0251  6466             LJMP    INCR            /NO, BAD UNIT NUMBER, GO TO INCREMENT WD1
0252  1000             LDA
0253  0021             WD1                     /GET WORD 1
0254  0243             ROL     3               /MOVE 3 LEFT
0255  1560             BCL+20                  /CLEAR ALL BUT 2 LSB'S
0256  7774             7774
0257  4026             STC     XOBWD           /STORE IN XOB WORD
```

```
                    /THIS SECTION OF CODING SETS UP FOR EXTENDED ADDRESS OPERATIONS

0260  1000    EXTEND, LDA                     /GET WORD 1 INTO AC
0261  0021            WD1
0262  1560            BCL*20                  /MASK TO BIT 3
0263  7377            7377
0264  0470            AZE*20                  /EXTENDED ADDRESS OPERATIONS?
0265  6344            LJMP    NONEXT          /NO
0266  0304            ROR     4               /YES, MOVE TO BIT 7
0267  2026            ADD     XOBWD           /COMBINE WITH OTHER BITS
0270  4026            STC     XOBWD           /AND STORE
0271  2022    EXT0,   ADD     WD2             /GET WORD 2
0272  1560            BCL*20                  /MASK TO FIELD BITS
0273  7743            7743
0274  4027            STC     FIELDN          /SAVE
0275  2027            ADD     FIELDN          /GET FIELD
0276  0450            AZE                     /NON-ZERO?
0277  6316            LJMP    EXT3            /YES
0300  2024    EXT1,   ADD     WD4             /GET WORD 4
0301  3716            ADD     K4000
0302  0471            APO*20                  /AC POSITIVE?
0303  6307            LJMP    EXT2            /YES, OK SO FAR
0304  7637            LJMP    RANDOM          /NO, ADDRESS IS 3777 OR BELOW
0305  4024            STC     WD4
0306  6300            LJMP    EXT1
0307  1000    EXT2,   LDA                     /GET WORD 4 AGAIN
0310  0024            WD4
0311  1120            ADA*20                  /ADD -7400
0312  0377            0377
0313  0471            APO*20                  /AC MINUS?
0314  6304            LJMP    EXT2-3          /NO, ADDRESS IS ABOVE 7400
0315  6331            LJMP    EXT4            /YES, ADDRESS IS OK
0316  0516    EXT3,   RSW                     /READ RIGHT SWITCHES
0317  1560            BCL*20                  /MASK TO FIELD BITS
0320  7770            7770
0321  0242            ROL 2                   /2 LEFT
0322  0017            COM                     /MAKE NEGATIVE
0323  2027            ADD     FIELDN
0324  0451            APO                     /DO WE HAVE THE MEMORY?
0325  6307            LJMP    EXT2            /YES, CHECK THE ADDRESS
0326  7637            LJMP    RANDOM          /NO, GET A NEW FIELD NUMBER
0327  4022            STC     WD2
0330  6271            LJMP    EXT0
0331  1000    EXT4,   LDA                     /GET FIELD AGAIN
0332  0027            FIELDN
0333  0305            ROR     5               /MOVE 5 RIGHT
0334  2026            ADD     XOBWD           /COMBINE WITH OTHER BITS
0335  4026            STC     XOBWD           /AND STORE
0336  2023            ADD     WD3             /GET WORD 3
0337  1560            BCL*20                  /MASK TO BITS 8 TO 11
0340  7760            7760
0341  2411            ADD     K0770
0342  4032            STC     QNBN            /STORE IN QNBN SAVE
0343  6414            LJMP    PAUSEB
```

```
                    /THIS SECTION OF CODING SETS UP FOR NON-EXTENDED ADDRESS OPERATION

0344  1000    NONEXT, LDA                     /GET WORD 2
0345  0022            WD2
0346  1560            BCL+20                  /MASK TO LINC MEMORY FIELD
0347  7740            7740
0350  4027            STC     FIELDN          /AND SAVE
0351  2027            ADD     FIELDN          /GET LINC MEMORY FIELD
0352  1120            ADA+20
0353  7776            7776
0354  0471            APO+20                  /IS IT NOT 0 OR 1?
0355  6361            LJMP    .+4             /YES
0356  7637    NONEX1, LJMP    RANDOM          /NO, IT IS 0 OR 1, GET ANOTHER
0357  4022            STC     WD2             /STORE
0360  6344            LJMP    NONEXT          /GO BACK AND TRY AGAIN
0361  0516            RSW                     /READ RIGHT SWITCHES
0362  1560            BCL+20                  /MASK TO FIELD BITS
0363  7770            7770
0364  0242            ROL     2               /LEFT 2
0365  2377            ADD     K0003
0366  0017            COM                     /MAKE NEGATIVE
0367  2027            ADD     FIELDN
0370  0471            APO+20                  /DO WE HAVE THE MEMORY?
0371  6356            LJMP    NONEX1          /NO
0372  1000            LDA                     /GET WORD 1
0373  0021            WD1
0374  1560            BCL+20                  /CLEAR TO FUNCTION BITS
0375  7770            7770
0376  1460            SAE+20                  /SKIP IF MTB
0377  0003    K0003,  3
0400  6406            LJMP    NONEX2          /NOT MTB
0401  1000            LDA                     /GET WORD 3
0402  0023            WD3
0403  1560            BCL+20                  /MASK TO BITS 3 TO 11
0404  7000            7000
0405  6413            LJMP    NONEX3
0406  1000    NONEX2, LDA                     /GET WORD 3
0407  0023            WD3
0410  1560            BCL+20                  /CLEAR TO GN, BN (0 TO 2, 9 TO 11)
0411  0770    K0770,  0770
0412  2411            ADD     .+1             /ADD 770
0413  4032    NONEX3, STC     GNBN            /STORE IN GNBN SAVE
```

```
                        /THIS SECTION OF CODING SETS UP THE "PAUSE" BIT
                        /IF "NO PAUSE" IS SPECIFIED, CONTROL WILL THEN GO
                        /TO THE TAPE INTERRUPT ENABLE BIT HANDLER

0414  1000    PAUSEB, LDA                     /GET WORD 1
0415  0021            WD1
0416  0017            CCM                     /COMPLEMENT AC
0417  1560            BCL+20                  /MASK TO "PAUSE" BIT
0420  7737            7737
0421  0302            ROR     2               /2 RIGHT
0422  1140            ADM                     /COMBINE WITH XOB WORD
0423  0026            XOBWD
0424  1560            BCL+20                  /MASK TO "DON'T PAUSE" BIT
0425  7767            7767
0426  0470            AZE+20                  /SKIP IF SET
0427  6436            LJMP    DISPCH

                        /TAPE INTERRUPT ENABLE BIT HANDLER
                        /THIS SECTION OF CODING IS ENTERED ONLY IF
                        /THE "NO PAUSE" BIT IS TRUE

0430  1000    TPINEN, LDA                     /GET WORD 1
0431  0021            WD1
0432  1560            BCL+20                  /MASK TO BIT 5
0433  7677            7677
0434  2026            ADD     XOBWD           /COMBINE WITH OTHER BITS
0435  4026            STC     XOBWD           /AND STORE
```

```
                        /THIS SECTION OF CODING DISPATCHES THE PROGRAM
                        /TO THE APPROPRIATE SECTION OF CODING TO HANDLE
                        /THE PARTICULARS RELATING TO EACH MAG TAPE INSTRUCTION

0436  0011    DISPCH, CLR                     /GET WORD 1
0437  2021            ADD     WD1
0440  1560            BCL*20                  /MASK TO FUNCTION BITS
0441  7770            7770
0442  1120            ADA*20                  /ADD IN "MASTER JUMP"
0443  6446            LJMP    TABLE1
0444  4445            STC     .+1             /STORE
0445  6445            LJMP    .               /EXECUTE
0446  6456    TABLE1, LJMP    RDSUB           /READ AND CHECK          (0)
0447  6460            LJMP    RCKSUB          /READ AND CHECK GROUP    (1)
0450  6456            LJMP    RDSUB           /READ                    (2)
0451  6462            LJMP    MOVSUB          /MOVE TOWARD BLOCK       (3)
0452  7106            LJMP    WRITE           /WRITE AND CHECK         (4)
0453  7332            LJMP    WRCKGP          /WRITE AND CHECK GROUP   (5)
0454  7106            LJMP    WRITE           /WRITE                   (6)
0455  6464            LJMP    CHKSUB          /CHECK                   (7)
0456  6504    RDSUB,  LJMP    READ
0457  6466            LJMP    INCR
0460  6706    RCKSUB, LJMP    RDCKGP
0461  6466            LJMP    INCR
0462  7023    MOVSUB, LJMP    MOVE
0463  6466            LJMP    INCR
0464  7454    CHKSUB, LJMP    CHECK
0465  6466            LJMP    INCR

0466  1020    INCR,   LDA*20                  /INCREMENT MASTER WORD
0467  0001            1
0470  2020            ADD     MASTER
0471  0471            APO*20
0472  6501            LJMP    INCRA
0473  0601            LIF     1
0474  7131            LJMP    BELL
0475  1020            LDA*20
0476  0020            0020
0477  0004            ESF
0500  0011            CLR
0501  1040    INCRA,  STA
0502  0020            MASTER
0503  6221            LJMP    DATLUP          /GO BACK AGAIN
```

```
                        /TAPE 2
                        /THIS SECTION OF CODING HANDLES THE INSTRUCTIONS "READ"
                        /AND "READ AND CHECK BLOCK"

0504  2000    READ,   ADD     0
0505  4701            STC     REXIT           /SAVE RETURN ADDRESS
0506  1020            LDA*20                  /SET UP RETURN JUMP
0507  7610            LJMP    TDFLAG
0510  5575            STC     RJUMP           /FROM INSTRUCTION EXECUTION
0511  1020            LDA*20                  /SET UP FOR RETURN
0512  6610            LJMP    RCHK
0513  7540            LJMP    MTSET           /FROM FLAG HANDLING
0514  1560            BCL*20                  /MASK XOBWD TO EXTENDED ADDRESS MODE BIT
0515  7757            7757
0516  0470            AZE*20                  /EXTENDED ADDRESS MODE?
0517  6531            LJMP    REDNEX          /NO
0520  1000            LDA                     /YES
0521  0032            QNBN                    /GET QN=BN
0522  0601            LIF     1
0523  6722            LJMP    WRITEN          /HAS BLOCK BEEN WRITTEN?
0524  6701            LJMP    REXIT           /NO, EXIT
0525  4644            STC     PATJMP          /YES, OK, SAVE PATTERN WORD
0526  2024            ADD     WD4             /GET EXTENDED ADDRESS
0527  0023            TMA                     /LOAD TMA SETUP REGISTER
0530  7567            LJMP    MTXEQT          /EXECUTE "RDE OR RDC BN"
                        /HERE IF NOT EXTENDED ADDRESS MODE
0531  2032    REDNEX, ADD     QNBN            /GET QN=BN
0532  1560            BCL*20                  /CLEAR TO BLOCK NUMBER
0533  7000            7000
0534  0601            LIF     1
0535  6722            LJMP    WRITEN          /HAS BLOCK BEEN WRITTEN
0536  6701            LJMP    REXIT           /NO, EXIT
0537  4644            STC     PATJMP          /YES, OK, SAVE PATTERN WORD
0540  2032            ADD     QNBN            /GET QN=BN
0541  1560            BCL*20                  /MASK TO QN
0542  0777            777
0543  0451            APQ                     /DF OR IF
0544  6563            LJMP    REDDF           /DF
0545  0450            AZE                     /IF, Q0?
0546  6552            LJMP    .+4             /NOT Q0
0547  1020            LDA*20                  /Q0, INSTRUCTION
0550  0400            400                     /WILL BE STORED IN Q1
0551  0456            LSKP
0552  0011            CLR                     /NOT Q0, INSTRUCTION WILL BE STORED IN Q0
0553  4035            STC     CSTART
0554  2027            ADD     FIELDN          /GET FIELD BITS
0555  2631            ADD     LDFCON
0556  4561            STC     LDFRD1          /AND STORE

0557  2561    REDNX1, ADD     LDFRD1          /GET LDF INSTRUCTION
0560  7657            LJMP    COMON7          /SET UP "MOVLIF" AND "MOVLDF"
0561  0000    LDFRD1, 0                       /STORAGE FIELD LDF GETS STORED HERE
0562  7511            LJMP    MOVPRO          /MOVE "LITTLE PROGRAM", THEN EXECUTE IT
```

```
                        /HERE IF DATA FIELD
0563  1000    REDDF,    LDA                       /GET FIELD
0564  0027              FIELDN
0565  1120              ADA+20                    /SUBTRACT 2
0566  7775              7775
0567  0470              AZE+20                    /FIELD 2?
0570  6575              LJMP      .+5             /YES
0571  2643              ADD       LD1CON
0572  4561              STC       LDFRD1          /STORE AWAY
0573  4035              STC       CSTART          /SET UP QUARTER STORAGE ADDRESS
0574  6557              LJMP      REDNX1          /SET UP MEMORY, ETC.
0575  1020              LDA+20                    /SET UP
0576  0643              LDF       3
0577  4561              STC       LDFRD1          /STORAGE FIELD
0600  1020              LDA+20
0601  0642              LDF       2
0602  5535              STC       MOVLDF          /DATA FIELD
0603  1020              LDA+20
0604  0603              LIF       3
0605  5534              STC       MOVLIF          /INSTRUCTION FIELD
0606  4035              STC       CSTART          /SET UP QUARTER STORAGE ADDRESS
0607  6561              LJMP      LDFRD1          /GO EXECUTE LITTLE PROGRAM (EVENTUALLY)
```

```
                        /RETURN HERE IF FLAGS OK UPON INSTRUCTION COMPLETION

0610  1000    RCHK,   LDA                     /GET INSTRUCTION EXECUTED
0611  1572            MTINST
0612  1560            BCL*20                  /CLEAR ALL BUT INSTRUCTION EXECUTED
0613  7770            7770
0614  0450            AZE                     /"READ AND CHECK" INSTRUCTION?
0615  6621            LJMP    .+4             /NO
0616  0003            TAC                     /YES, READ TAPE AC
0617  0450            AZE                     /ZERO?
0620  7733            LJMP    XXX             /NO, ERROR *** ER 2 ***
0621  1000            LDA                     /GET XOB WORD
0622  0026            XOBWD
0623  1560            BCL*20                  /MASK TO EXTENDED ADDRESS BIT
0624  7757            7757
0625  0450            AZE                     /ZERO?
0626  6702            LJMP    EXTDCH          /NO
0627  2027            ADD     FIELDN          /YES, CALCULATE
0630  1120            ADA*20                  /WHERE
0631  0640    LDFCON, LDF                     /DATA
0632  4667            STC     DATCHK          /IS STORED
0633  3573            ADD     MTINST+1        /GET GN-BN
0634  1560            BCL*20                  /MASK TO 2 QUARTER BITS
0635  4777            4777
0636  0301            ROR     1               /RIGHT 1 PLACE TO FORM FIRST DATA ADDRESS
0637  4646            STC     DATADD          /STORE AWAY ADDRESS
```

```
0640  1020          LDA+20
0641  1400    K1400, 1400
0642  0601          LIF     1
0643  0641    LD1CON, LDF     1               /CHANGE DATA FIELDS
0644  6644    PATJMP, LJMP    .               /STORE NEW DATA THERE
0645  1020          LDA+20                    /SET UP ADDRESSES TO CHECK DATA
0646  0000    DATADD, 0                       /ADDRESS OF DATA READ
0647  7760          LJMP    SUBT1             /SUBTRACT 1 FROM THE AC
0650  1620          BSE+20                    /SET BIT 01 OF THE AC
0651  2000          2000
0652  4015          STC     15                /SET 0015 TO STARTING ADDRESS OF THE DATA READ
0653  0076          SET+20  16                /SET 0016 TO THE STARTING ADDRESS OF THE EXPECTED DATA
0654  3377          3377
0655  0077          SET+20  17                /SET 0017 TO A COUNT
0656  7377          7377                      / LOCATION
0657  1000          LDA                       /LOAD THE AC WITH A "CDF N" INSTRUCTION
0660  0667          DATCHK
0661  1040          STA                       /SAVE IT
0662  3013          SAVA
0663  0017          COM                       /SET THE AC TO 7777
0664  0261          ROL+20  1                 /SET THE LINK
                                              /IT WILL BE USED IN "TST1"
0665  0641          LDF     1
0666  1036          LDA+20  16                /CHECK DATA EXPECTED
0667  0640    DATCHK, LDF                     /CHANGE DATA FIELD
0670  1475          SAE+20  15                /AGAINST DATA READ
0671  7733          LJMP    XXX               /DATA ERROR *** ER 7 ***
0672  1000          LDA                       /LOAD THE AC WITH THE VALUE IN LOC 0015
0673  0015          0015
0674  0601          LIF     1
0675  6771          LJMP    TST1              /TEST THE LIMIT OF LOC 15
0676  0237    DATINC, XSK+20  17              /TEST MORE DATA TO TEST ?
0677  6665          LJMP    DATCHK-2          /YES, GO DO IT
0700  6161          LJMP    CLEAR
0701  6701    REXIT,  LJMP    .               / NO, EXIT

0702  0601    EXTDCH, LIF     1
0703  6210          LJMP    COMON4            /SET UP FOR EXTENDED ADDRESSING MODE DATA
0704  4667          STC     DATCHK            /STORE PROPER "LDF" FOR ACCESSING DATA
0705  6640          LJMP    PATJMP-4
```

```
                    /THIS SECTION OF CODING HANDLES THE INSTRUCTION "READ AND CHECK GROUP"

0706  2000    RDCKGP, ADD     0
0707  5022            STC     RGEXIT          /SAVE RETURN ADDRESS
0710  0601            LIF     1
0711  6020            LJMP    COMON1          /CHECK EXTENDED ADDRESSING TAPE/MEMORY WRAPAROUND, ETC.
0712  7022            LJMP    RGEXIT          /ILLEGAL OPERATION EXIT JUMP
0713  0601            LIF     1
0714  6137            LJMP    COMON2          /SET UP TO COUNT BLOCKS (RETURN WITH BN IN AC)
0715  0601    RGCON1, LIF     1
0716  6722            LJMP    WRITEN          /HAS BLOCK (NUMBER IN AC) BEEN WRITTEN?
0717  7022            LJMP    RGEXIT          /NO, EXIT
0720  1020            LDA*20                  /YES, SET AC TO 1
0721  0001            1
0722  1140            ADM                     /ADD TO BLOCK NUMBER
0723  0034            CTEM3
0724  0234            XSK*20  14              /DONE TESTING BLOCKS?
0725  6715            LJMP    RGCON1          /NO
0726  1020    RGCON2, LDA*20                  /SET UP RETURN JUMP
0727  7610            LJMP    TDFLAG
0730  5575            STC     RJUMP           /FROM INSTRUCTION EXECUTION
0731  1020            LDA*20                  /SET UP FOR RETURN
0732  6744            LJMP    RGCHK
0733  7540            LJMP    MTSET           /FROM FLAG HANDLING
0734  1000            LDA                     /GET FIELD
0735  0027            FIELDN
0736  2631            ADD     LDFCON
0737  1040            STA                     /AND STORE AS STORAGE FIELD
0740  0742            LDFRG1
0741  7657            LJMP    COMON7          /SET UP "MOVLIF" AND "MOVLDF"
0742  0640    LDFRG1, LDF                     /STORAGE FIELD GETS STORED HERE
0743  7511            LJMP    MOVPRO          /MOVE "LITTLE PROGRAM", THEN EXECUTE IT
```

```
                    /RETURN HERE IF FLAGS OK UPON INSTRUCTION COMPLETION

0744  0003  RGCHK,  TAC                     /READ TAPE AC
0745  0450          AZE                     /ZERO?
0746  7733          LJMP    XXX             /NO, ERROR *** ER 2 ***
0747  0601          LIF     1
0750  6137          LJMP    COMON2          /SET UP TO COUNT BLOCKS
0751  0011          CLR
0752  2034          ADD     CTEM3           /GET BLOCK NUMBER
0753  0601  RGCON3, LIF     1
0754  6722          LJMP    WRITEN          /FIND OUT BLOCK PATTERN ADDRESS
0755  0000          LHLT                    /THIS RETURN SHOULD NOT BE USED
0756  4762          STC     PATFRM          /STORE AWAY
0757  2641          ADD     K1400
0760  0601          LIF     1
0761  0641  LDFRG2, LDF     1               /STORAGE FIELD "LDF" IS STORED HERE
0762  6762  PATFRM, LJMP    .               /PATTERN JUMP IS STORED HERE, RETURN WITH DATA STORED
0763  1000          LDA                     /GET BLOCK NUMBER
0764  0034          CTEM3
0765  0601          LIF     1
0766  6162          LJMP    COMON3          /COMPUTE DATA FIELD TO ACCESS DATA
0767  5006          STC     LDFRG3          /STORE "LDF" INSTRUCTION (IN AC FROM COMON3)
0770  2034          ADD     CTEM3           /GET BLOCK NUMBER AGAIN
0771  1560          BCL*20                  /MASK TO BN 10,11
0772  7774          7774
0773  0304          ROR     4               /4 RIGHT TO FORM ADDRESS
0774  7760          LJMP    SUBT1           /SUBTRACT 1
0775  1620          BSE*20                  /SET DATA FIELD BIT
0776  2000          2000
0777  4015          STC     15              /STORE IN 15 (DATA READ)
1000  0076          SET*20  16              /SET UP 16 FOR CHECK DATA
1001  3377          3377
1002  0077          SET*20  17              /SET UP 17 FOR COUNT (400)
1003  7377          7377
1004  0641          LDF     1               /DATA EXPECTED "LDF"
1005  1036          LDA*20  16              /GET CHECK DATA
1006  0640  LDFRG3, LDF                     /DATA READ FIELD "LDF"
1007  1475          SAE*20  15              /COMPARE AGAINST DATA READ
1010  7733          LJMP    XXX             /NO, DATA ERROR *** ER 8 ***
1011  0237          XSK*20  17              /COMPARED 400?
1012  7004          LJMP    LDFRG3-2        /NO, GO BACK FOR NEXT DATA WORD
1013  1020          LDA*20                  /YES, INCREMENT
1014  0001  K0001,  1
1015  1140          ADM                     /BLOCK NUMBER
1016  0034          CTEM3
1017  0234          XSK*20  14              /DONE ALL BLOCKS?
1020  6753          LJMP    RGCON3          /NO, REPEAT FOR NEXT BLOCK
1021  6161          LJMP    CLEAR
1022  7022  RGEXIT, LJMP    .               /EXIT
```

```
                    /THIS SECTION OF CODING HANDLES THE INSTRUCTION "MOVE"

1023  2000  MOVE,   ADD     0
1024  5105          STC     MEXIT           /SAVE RETURN ADDRESS
1025  1020          LDA+20                  /SET UP RETURN JUMP
1026  7610          LJMP    TDFLAG
1027  5575          STC     RJUMP           /FROM INSTRUCTION EXECUTION
1030  3572          ADD     MTINST          /NO, GET THE LAST TAPE INST.
1031  0641          LDF     1               /FIELD 1
1032  0247          ROL     7
1033  1040          STA                     /SAVE THE VALUE IN "IBIT"
1034  3112          IBIT                    /SAVE THE PREVIOUS I BIT
1035  0471          APO+20                  / BIT 0 =1 ?
1036  7104          LJMP    MEXIT-1         / NO EXIT
1037  1020          LDA+20                  / YES EXECUTE THIS MTB
1040  7044          LJMP    MCH
1041  7540          LJMP    MTSET           /SET UP A RETURN ADD.
1042  7766          LJMP    MPAC            /SET THE "I" BIT IN THE MTB
1043  7567          LJMP    MTXEQT          /EXECUTE "MTB"
1044  0003  MCH,    TAC                     /RETURN HERE, TAC=0 ?
1045  0470          AZE+20
1046  7104          LJMP    MEXIT-1         /YES EXIT
1047  0640          LDF     0
1050  0601          LIF     1
1051  7100          LJMP    TSIGN1          /NO TEST THE LIMITS
1052  1020  MCCH,   LDA+20
1053  7061          LJMP    MCHK
1054  7540          LJMP    MTSET           /FROM FLAG HANDLING
1055  0070          SET+20  10              /SET 10 TO =1 (1'S COMP)
1056  7776          7776
1057  7766          LJMP    MPAC            /SET BIT "7"
1060  7567          LJMP    MTXEQT          /EXECUTE "MTB 8N"
                    /RETURN HERE IF FLAGS OK UPON INSTRUCTION COMPLETION

1061  0003  MCHK,   TAC                     /READ BACK THE TAPE AC
1062  0470          AZE+20                  /ZERO?
1063  7101          LJMP    MCHK1           /YES
1064  0230          XSK+20  10              /NO, FIRST NUMBER READ BACK?
1065  7075          LJMP    MCOMP           /NO
1066  0451  MBUMP,  APO                     /YES, POSITIVE AC?
1067  7072          LJMP    .+3             /NO, NEGATIVE
1070  7760          LJMP    SUBT1           /YES, DECREMENT
1071  0456          LSKP
1072  3014          ADD     K0001
1073  5076          STC     MEXPT           /SAVE
1074  7101          LJMP    MCHK1
1075  1460  MCOMP,  SAE+20                  /IS THE NUMBER READ EQUAL
1076  0000  MEXPT,  0                       /TO THE NUMBER EXPECTED?
1077  7733          LJMP    XXX             /NO, ERROR *** ER 3 ***
1100  7066          LJMP    MBUMP           /YES, SET UP FOR NEXT NUMBER
1101  0003  MCHK1,  TAC                     /READ TAC AGAIN
1102  0450          AZE                     /ZERO?
1103  7567          LJMP    MTXEQT
1104  0011          CLR
1105  7105  MEXIT,  LJMP    .               /EXIT
```

```
                        /THIS SECTION OF CODING HANDLES THE INSTRUCTIONS "WR   "
                        /AND "WRITE AND CHECK BLOCK"

1106  1020    WRITE,  LDA+20                  /SET UP RETURN JUMP
1107  7610            LJMP    TDFLAG
1110  5575            STC     RJUMP           /FROM INSTRUCTION EXECUTION
1111  1020            LDA+20                  /SETUP FOR RETURN
1112  7241            LJMP    WCHK
1113  7540            LJMP    MTSET           /FROM FLAG HANDLING
1114  1560            BCL+20                  /MASK XOBWD TO EXTENDED ADDRESS MODE BIT
1115  7757            7757
1116  0470            AZE+20                  /EXTENDED ADDRESS MODE?
1117  7143            LJMP    WRINEX          /NO
1120  0601            LIF     1
1121  6210            LJMP    COMON4          /YES, SET UP FOR EXTENDED ADDRESS MODE DATA
1122  5125            STC     LDFWR1          /STORE PROPER "LDF" FOR STORING DATA
1123  2046            ADD     DATADD          /GET ADDRESS WHERE DATA SHOULD BE STORED
1124  0601            LIF     1
1125  0640    LDFWR1, LDF                     /CHANGE DATA FIELD
1126  6254            LJMP    PATERN          /PUT DATA PATTERN IN MEMORY
1127  5317            STC     WPAT            /SAVE PATTERN TYPE (IN AC UPON RETURN)
1130  3573            ADD     MTINST+1        /GET ON-BN
1131  0601            LIF     1
1132  6236            LJMP    COMON5          /CALCULATE BLOCK STATUS WORD ADDRESS
1133  1040            STA                     /SAVE
1134  1322            UNBNSV
1135  5137            STC     .+2             /STORE FOR EXECUTION
1136  1040            STA                     /CLEAR STATUS WORD
1137  0000            0
1140  2024            ADD     WD4             /GET EXTENDED ADDRESS
1141  0023            TMA                     /LOAD TMA SETUP REGISTER
1142  7567            LJMP    MTXEQT          /EXECUTE "WRI OR WRC BN"
```

```
                        /HERE IF NOT EXTENDED ADDRESS MODE
1143  2027    WRINEX, ADD     FIELDN          /GET FIELD NUMBER
1144  2631            ADD     LDFCON
1145  1040            STA
1146  1212            LDFWR3
1147  5157            STC     LDFWR2          /STORE AWAY FOR EXECTUION AND FUTURE USE
1150  3573            ADD     MTINST*1        /GET QN-BN
1151  1560            BCL*20                  /MASK TO 2 QUARTER BITS (1 AND 2)
1152  4777            4777
1153  0301            ROR     1               /RIGHT 1 PLACE TO FORM DATA ADDRESS
1154  4646            STC     DATADD          /STORE AWAY ADDRESS
1155  2646            ADD     DATADD          /GET ADDRESS WHERE DATA SHOULD BE STORED
1156  0601            LIF     1
1157  0640    LDFWR2, LDF                     /CHANGE DATA FIELD
1160  6254            LJMP    PATERN          /PUT DATA PATTERN IN MEMORY
1161  5317            STC     WPAT            /SAVE PATTERN TYPE (IN AC UPON RETURN)
1162  3573            ADD     MTINST*1        /GET QN-BN
1163  1560            BCL*20                  /MASK TO BN BITS
1164  7000            7000
1165  0601            LIF     1
1166  6236            LJMP    COMON5          /CALCULATE BLOCK STATUS WORD ADDRESS
1167  1040            STA
1170  1322            UNBNSV                  /SAVE
1171  5173            STC     .+2             /STORE FOR EXECUTION
1172  1040            STA
1173  0000            0                       /EXECUTE TO CLEAR STATUS WORD
1174  3573            ADD     MTINST*1        /GET QN-BN
1175  1560            BCL*20                  /MASK TO QN
1176  0777            777
1177  0451            APO                     /DF OR IF?
1200  7214            LJMP    WRIDF           /DF
1201  0450            APE                     /IF,Q0?
1202  7206            LJMP    .+4             /NOT Q0
1203  1020            LDA*20                  /Q0, INSTRUCTIONS
1204  0400            400                     /WILL BE STORED IN Q1
1205  0456            LSKP
1206  0011            CLR                     /NOT Q0, INSTRUCTIONS WILL BE STORED IN Q0
1207  4035            STC     CSTART
1210  3212    WRINX1, ADD     LDFWR3          /GET LDF INSTRUCTION
1211  7657            LJMP    COMON7          /SET UP "MOVLIF" AND "MOVLDF"
1212  0640    LDFWR3, LDF                     /STORAGE FIELD LDF GETS STORED HERE
1213  7511            LJMP    MOVPRO          /MOVE "LITTLE PROGRAM", THEN EXECUTE IT
```

```
                        /HERE IF DATA FIELD
1214  1000    WRIDF,    LDA                       /GET FIELD
1215  0027              FIELDN
1216  1120              ADA+20                    /SUBTRACT 2
1217  7775              7775
1220  0470              AZE+20                    /FIELD 2?
1221  7226              LJMP      .+5             /YES
1222  2643              ADD       LD1CON
1223  5212              STC       LDFWR3          /STORE AWAY
1224  4035              STC       CSTART          /SET UP QUARTER STORAGE ADDRESS
1225  7210              LJMP      WRINX1          /SET UP MEMORY, ETC.
1226  1020              LDA+20                    /SET UP
1227  0643              LDF       3
1230  5212              STC       LDFWR3          /STORAGE FIELD
1231  1020              LDA+20
1232  0642              LDF       2
1233  5535              STC       MOVLDF          /DATA FIELD
1234  1020              LDA+20
1235  0603              LIF       3
1236  5534              STC       MOVLIF          /INSTRUCTION FIELD
1237  4035              STC       CSTART          /SET UP QUARTER STORAGE ADDRESS
1240  7212              LJMP      LDFWR3          /GO EXECUTE LITTLE PROGRAM (EVENTUALLY)
```

```
                        /RETURN HERE IF FLAGS OK UPON INSTRUCTION COMPLETION

1241  1000    WCHK,   LDA                     /GET INSTRUCTION EXECUTED
1242  1572            MTINST
1243  1560            BCL*20                  /CLEAR I AND U
1244  0030            0030
1245  1040            STA                     /SAVE FOR FUTURE REFERENCE
1246  1305            WINST
1247  1460            SAE*20                  /WRITE AND CHECK INSTRUCTION?
1250  0704            WRC
1251  7255            LJMP    .+4             /NO, WRITE INSTRUCTION
1252  0003            TAC                     /READ TAPE AC
1253  0450            AZE                     /ZERO?
1254  7733            LJMP    XXX             /NO,ERROR *** ER 2 ***
1255  1000            LDA                     /GET XOB WORD
1256  0026            XOBWD
1257  0325            ROR*20  5               /MOVE EXTENDED ADDRESS BIT INTO LINK
1260  1000            LDA                     /GET GN-BN
1261  1573            MTINST+1
1262  1060            STA*20                  /SAVE FOR FUTURE REFERENCE
1263  0000    WINST1, 0
1264  3014            ADD     K0001
1265  0472            LZE*20                  /EXTENDED ADDRESS MODE?
1266  7301            LJMP    WCONT1          /NO, GO DIRECTLY TO EXECUTE "MTB BN+1"
1267  1040            STA                     /YES, SAVE
1270  1300            WTEMP
1271  1120            ADA*20                  /SUBTRACT 1007
1272  6770            6770
1273  0451            APO                     /LEGITIMATE NEXT BLOCK?
1274  7277            LJMP    .+3             /YES
1275  0011            CLR                     /NO, NEXT BLOCK IS 0
1276  7301            LJMP    WCONT1
```

```
1277  1020            LDA+20                  /GET NEXT BLOCK

1300  0000  WTEMP,    0
1301  7576  WCONT1,   LJMP    COMON6          /SET UP AND EXECUTE "MTB BN+1"
1302  3263            ADD     WINST1          /MOVE "BN" BACK
1303  4032            STC     ONBN
1304  1020            LDA+20                  /GET ORIGINAL INSTRUCTION EXECUTED
1305  0000  WINST,    0
1306  1460            SAE+20                  /WRITE INSTRUCTION?
1307  0706            WRI
1310  7316            LJMP    WCONT2          /NO
1311  1020            LDA+20                  /YES, ADD 4 TO
1312  0004            4
1313  2021            ADD     WD1             /WORD 1
1314  4021            STC     WD1
1315  7454            LJMP    CHECK           /EXECUTE A "CHECK BN"
1316  1020  WCONT2,   LDA+20                  /GET PATTERN TYPE WRITTEN IN BLOCK
1317  0000  WPAT,     0
1320  0641            LDF     1
1321  1040            STA
1322  0000  UNBNSV,   0                       /STORE IN BLOCK PATTERN INDICATOR
1323  1000            LDA                     /GET WORD1
1324  0021            WD1
1325  1560            BCL+20                  /CLEAR FUNCTION BITS TO
1326  0007            7                       /CREATE "RDC"
1327  4021            STC     WD1             /STORE BACK
1330  6504            LJMP    READ            /GO TO SUBROUTINE TO EXECUTE "RDC BN"
1331  6466  WEXIT,    LJMP    INCR            /EXIT
```

```
                        /TAPE 3
                        /THIS SECTION OF CODING HANDLES THE INSTRUCTION "WRITE AND CHECK GROUP"

1332  0601    WRCKGP, LIF     1
1333  6020            LJMP    COMON1          /CHECK EXTENDED ADDRESSING, TAPE/MEMORY WRAPAROUND, ETC.
1334  7453            LJMP    WGEXIT          /ILLEGAL OPERATION EXIT JUMP
1335  1000            LDA                     /SET UP INSTRUCTION FIELD
1336  0027            FIELDN
1337  2631            ADD     LDFCON
1340  7657            LJMP    COMON7          /SET UP "MOVLIF" AND "MOVLDF"
1341  0601            LIF     1
1342  6137            LJMP    COMON2          /SET UP TO COUNT BLOCKS.(RETURN WITH BLOCK NUMBER[ IN AC)
1343  0601    WGCON1, LIF     1
1344  6162            LJMP    COMON3          /COMPUTE DATA FIELD TO STORE DATA
1345  5353            STC     LDFWG1          /STORE "LDF" INSTRUCTION [IN AC FROM COMON2)
1346  2034            ADD     CTEM3           /GET BLOCK NUMBER
1347  1560            BCL*20                  /MASK TO BN 10,11
1350  7774            7774
1351  0304            ROR     4               /4 RIGHT TO FORM ADDRESS
1352  0601            LIF     1
1353  0640    LDFWG1, LDF                     /CHANGE DATA FIELD TO STORE DATA
1354  6254            LJMP    PATERN          /PUT PATTERN IN MEMORY
1355  5363            STC     WGPAT           /SAVE PATTERN ADDRESS
1356  2034            ADD     CTEM3           /GET BLOCK NUMBER
1357  0601            LIF     1
1360  6236            LJMP    COMON5          /COMPUTE BLOCK STATUS WORD ADDRESS
1361  5366            STC     .+5             /STORE
1362  1120            ADA*20                  /GET PATTERN ADDRESS
1363  0000    WGPAT,  0
1364  0641            LDF     1
1365  1040            STA                     /STORE AWAY
1366  0000            0
1367  1020            LDA*20                  /INCREMENT
1370  0001            1
1371  1140            ADM                     /BLOCK NUMBER
1372  0034            CTEM3
1373  0234            XSK*20  14              /DONE ALL BLOCKS (AND QUARTERS)?
1374  7343            LJMP    WGCON1          /NO, GO BACK TO DO NEXT BLOCK(QUARTER)
1375  1020            LDA*20                  /SET UP RETURN JUMP
1376  7412            LJMP    WGRET
1377  5375            STC     RJUMP           /FROM INSTRUCTION EXECUTION
1400  1020            LDA*20                  /SET UP FOR RETURN
1401  7430            LJMP    WGCHK
1402  7540            LJMP    MTSET           /FROM FLAG HANDLING
1403  1000            LDA                     /GET "LITTLE" PROGRAM INSTRUCTION FIELD
1404  1534            MOVLIF
1405  1120            ADA*20                  /MAKE AN
1406  0040            40                      /"LDF" INSTRUCTION
1407  5410            STC     LDFWG2          /STORE AWAY
1410  0640    LDFWG2, LDF                     /EXECUTE THE LDF
1411  7511            LJMP    MOVPRO          /MOVE "LITTLE PROGRAM", THEN EXECUTE IT
```

```
                    /RETURN HERE AFTER EXECUTING THE "LITTLE PROGRAM"

1412  1020  WGRET,  LDA+20
1413  0773           773
1414  0601           LIF     1
1415  6722           LJMP    WRITEN          /GET BLOCK PATTERN ADDRESS FOR Q3
1416  7427           LJMP    Q3PAT+1         /Q3 NEED NOT BE RELOADED
1417  5426           STC     Q3PAT           /SAVE
1420  3410           ADD     LDFWC2          /GET "INSTRUCTION FIELD"
1421  5425           STC     Q3PAT+1         /STORE FOR EXECUTION
1422  1020           LDA+20                  /GET STORAGE ADDRESS
1423  1400           1400
1424  0601           LIF     1
1425  0640           LDF                     /EXECUTE THE LDF
1426  6000  Q3PAT,   LJMP                    /FILL Q3 AGAIN ("LITTLE" PROGRAM WAS STORED
                                              /ON THE DATA ORIGINALLY STORED THERE)
1427  7610           LJMP    TDFLAG          /GO TO TAPE DONE ROUTINE AFTER LOADING MEMORY

                    /RETURN HERE IF FLAGS OK UPON INSTRUCTION COMPLETION

1430  0003  WGCHK,  TAC                     /READ TAPE AC
1431  0450           AZE                     /ZERO?
1432  7733           LJMP    XXX             /NO, ERROR *** ER 2 ***
1433  3573           ADD     MTINST+1        /GET QN-BN
1434  1040           STA                     /SAVE
1435  1450           WGQNBN
1436  2033           ADD     CTEM1           /ADD QN
1437  2003           ADD     K0002
1440  7576           LJMP    COMON6          /SET UP AND EXECUTE "MTB BN+QN+1"
1441  1000           LDA                     /GET WORD1
1442  0021           WD1
1443  1560           BCL+20                  /CLEAR OUT FUNCTION BITS
1444  0007           7
1445  3014           ADD     K0001
1446  4021           STC     WD1             /PUT BACK
1447  1020           LDA+20                  /GET
1450  0000  WGQNBN,  0                       /QN-BN OF WRG INSTRUCTION
1451  4032           STC     QNBN            /PLACE IN QN-BN
1452  6706           LJMP    RDCKGP          /EXECUTE A "RCG QNBN"
1453  6466  WGEXIT,  LJMP    INCR            /EXIT
```

```
                        /THIS SECTION OF CODING HANDLES THE INSTRUCTION "CHECK"

1454  2000    CHECK,  ADD     0
1455  5510            STC     CEXIT
1456  1020            LDA+20                  /SET UP RETURN JUMP
1457  7610            LJMP    TDFLAG
1460  5575            STC     RJUMP           /FROM INSTRUCTION EXECUTION
1461  1020            LDA+20                  /SET UP FOR RETURN
1462  7502            LJMP    CCHK
1463  7540            LJMP    MTSET           /FROM FLAG HANDLING
1464  0325            ROR+20  5               /MOVE EXT. ADDRESS BIT INTO THE LINK
1465  1000            LDA                     /GET THE BLOCK NUMBER
1466  0032            QNBN
1467  0452            LEE                     /EXTENDED ADDRESSING ?
1470  7474            LJMP    .+4             /YES
1471  1560            BCL+20                  /NO, MASK TO BITS 3-11
1472  7000            7000
1473  7476            LJMP    .+3             /THEN CHECK IT
1474  1560            BCL+20                  /EXTENDED ADDRESSING, MASK TO BITS 2-11
1475  6000            6000
1476  0001            LIF     1               /THEN CHECK IT IF IT HAS BEEN WRITTEN IN
1477  6722            LJMP    WRITEN
1500  7506            LJMP    CEXIT-2         /NO, THE BLOCK HAS NOT BEEN WRITTEN IN, EXIT
1501  7567            LJMP    MTXEQT          /EXECUTE "CHECK BN"

                        /RETURN HERE IF FLAGS OK UPON INSTRUCTION COMPLETION
1502  0003    CCHK,   TAC                     /READ TAPE AC
1503  0450            AZE                     /ZERO?
1504  7733    CCHKA,  LJMP    XXX             /NO, ERROR *** ER 2 ***
1505  7701            LJMP    CHECKI          /CHECK TAPE MOTION
1506  0011            CLR
1507  5572            STC     MTINST          /CLEAR MTINST
1510  7510    CEXIT,  LJMP    .               /EXIT
```

```
                /ROUTINE TO MOVE "LITTLE PROGRAM" TO APPROPRIATE PLAC  MEMORY
                /THEN EXECUTE IT
                /ENTER WITH DATA FIELD SET FOR STORAGE
                /"LITTLE PROGRAM" WILL BE MOVED FROM MINST, THEN EXECUTED

1511  0011      MOVPRO, CLR
1512  2035              ADD     CSTART
1513  7760              LJMP    SUBT1           /SUBTRACT 1 FROM THE AC
1514  1620              BSE+20                  /SET DATA FIELD BIT
1515  2000              2000
1516  4011              STC     11              /STORE DESTINATION ADDRESS IN 11
1517  0072              SET+20  12              /SET ORIGIN ADDRESS INTO 12
1520  1566              MTXEQT-1
1521  0073              SET+20  13              /SET COUNT (-7) INTO 13
1522  7770              7770
1523  1032              LDA+20  12              /MOVE THE PROGRAM
1524  1071              STA+20  11
1525  0233              XSK+20  13
1526  7523              LJMP    .-3
1527  1000              LDA                     /GET STARTING ADDRESS OF THE PROGRAM
1530  0035              CSTART
1531  1620              BSE+20                  /FORM LJMP INSTRUCTION
1532  6000              LJMP
1533  5537              STC     MOVJMP          /STORE FOR EXECUTION
1534  0000  MOVLIF,     0                       /CHANGE INSTRUCTION FIELD
1535  0000  MOVLDF,     0                       /CHANGE DATA FIELD
1536  0006              DJR
1537  7537  MOVJMP,     LJMP    .               /JUMP TO "LITTLE PROGRAM"
```

```
                        /SUBROUTINE TO SET UP MAGTAPE INSTRUCTIONS
                        /SUBROUTINE IS ENTERED WITH "WHERE TO GO IF INTERRUPT OCCURS AS EXPECTED" IN AC
                        /SUBROUTINE EXITS WITH CONTENTS OF XOB WORD IN AC AND IN XOB

1540  4061    MTSET,  STC     MAGTAP          /SAVE INSTRUCTION WHERE WE HOPE IT WILL STAY
1541  2000            ADD     0
1542  5566            STC     MTEXIT          /SAVE RETURN ADDRESS
1543  2026            ADD     XOBWD           /GET XOB WORD
1544  1560            BCL*20                  /MASK TO TAPE INTERRUPT BIT
1545  7677            7677
1546  0450            AZE                     /BIT SET?
1547  7552            LJMP    .+3             /YES, SET LOCATION TO A LNOP
1550  3504            ADD     CCHKA
1551  7554            LJMP    .+3             / IN CASE INTERRUPT OCCURS
1552  1020            LDA*20                  /ERRONEOUSLY
1553  0016            LNOP
1554  4041            STC     TSTMOR-1
1555  2021            ADD     WD1
1556  1560            BCL*20                  /MASK TO INSTRUCTION BITS
1557  7740            7740
1560  3673            ADD     RDCCON
1561  5572            STC     MTINST          /STORE
1562  2032            ADD     QNBN
1563  5573            STC     MTINST+1        /MOVE QN-BN INDICATOR
1564  2026            ADD     XOBWD           /GET XOB WORD
1565  0001            AXO                     /LOAD XOB
1566  7566    MTEXIT, LJMP    .               /EXIT

                        /THIS IS THE "LITTLE PROGRAM"
                        /EXECUTE THE FOLLOWING MAGTAPE INSTRUCTIONS BY JUMPING HERE

1567  0011    MTXEQT, CLR
1570  0500            IOB
1571  6001            ION
1572  0000    MTINST, 0                       /MAGTAPE INSTRUCTION
1573  0000            0                       /QN-BN
1574  0600            LIF                     /SET INSTRUCTION FIELD BACK TO 0
1575  7610    RJUMP,  LJMP    TDFLAG          /NORMALLY THIS LOCATION WILL CONTAIN
                                              /AN "LJMP TDFLAG" TO PROCESS TAPE DONE FLAG
                                              /HOWEVER, THIS LOCATION WILL CONTAIN
                                              /"LJMP WGRET" IF A "WRG" INSTRUCTION
                                              /IS BEING EXECUTED
```

```
                /THIS SECTION OF CODING HANDLES SOME OF THE CALCULA  'S
                /COMMON TO THE "WRITE" AND "WRCKCP" SUBROUTINES
                /IN PARTICULAR, THIS ROUTINE SETS UP AND EXECUTES A "MO.  " INSTRUCTION
                /ENTER WITH BLOCK NUMBER OF BLOCK TO BE "MOVED TO" IN AC

1576  4032      COMON6, STC     QNBN            /STORE BN IN QN-BN LOCATION
1577  2000              ADD     0
1600  5607              STC     C6EXIT
1601  2021              ADD     WD1
1602  1560              BCL*20
1603  0007              7
1604  2377              ADD     K0003
1605  4021              STC     WD1
1606  7023              LJMP    MOVE
1607  7607      C6EXIT, LJMP    .

                /ROUTINE TO HANDLE "TAPE DONE" FLAG IF NO INTERRUPT OCCURS

1610  1000      TDFLAG, LDA                     /GET XOB WORD
1611  0026              XOBWD
1612  1560              BCL*20                  /MASK TO PAUSE BIT
1613  7767              7767
1614  0450              AZE                     /PAUSE?
1615  7621              LJMP    .+4             /NO, NOT PAUSE
1616  0416              STD                     /YES, PAUSE, IS TAPE DONE SET?
1617  7733              LJMP    XXX             /NO, NOT SET, ERROR *** ER 1 ***
1620  7625              LJMP    TLAG
1621  0436      TFLAG,  STD*20                  /HERE IF NO-PAUSE MODE
1622  7625              LJMP    TLAG
1623  0601              LIF     1
1624  7145              LJMP    DDISP
1625  0016      TLAG,   LNOP                    /WAIT 1 MORE CYCLE TO ALLOW PI TO OCCUR
1626  0500              IOB
1627  6002              IOF                     /TURN OFF PI
1630  1000              LDA                     /GET XOB WORD
1631  0026              XOBWD
1632  1560              BCL*20                  /MASK TO TAPE INTERRUPT BIT
1633  7677              7677
1634  0450              AZE                     /IS TAPE INTERRUPT BIT SET?
1635  7733              LJMP    XXX             /YES, ERROR, NO INTERRUPT OCCURRED *** ER 4 ***
1636  6042              LJMP    TSTMOR          /ALL OK, SO FAR, CHECK "TAPE DONE" IN 8-MODE
```

```
                        /RANDOM NUMBER GENERATOR - EXIT WITH RANDOM NUMBER IN AC

1637  1000    RANDOM, LDA
1640  0000            0
1641  5656            STC     RANXIT
1642  3654            ADD     HALFX
1643  3655            ADD     HALFY
1644  0261            ROL*20  1
1645  5655            STC     HALFY
1646  3655            ADD     HALFY
1647  3654            ADD     HALFX
1650  0261            ROL*20  1
1651  5654            STC     HALFX
1652  3655            ADD     HALFY
1653  7656            LJMP    .+3
1654  0001    HALFX,  0001
1655  0001    HALFY,  0001
1656  5256    RANXIT, JMP     .               /EXIT

                        /THIS SECTION OF CODING HANDLES SOME OF THE CALCULATIONS
                        /COMMON TO "READ", "RDCKGP", "WRITE", "WRCKGP" SUBROUTINES
                        /IN PARTICULAR, THIS SUBROUTINE SETS UP LOCATIONS "MOVLIF" AND "MOVLDF"
                        /ENTER WITH FIELD WHERE PROGRAM IS STORED IN AC
1657  5663    COMON7, STC     C7TEMP          /SAVE AC
1660  2000            ADD     0
1661  5700            STC     C7EXIT          /SAVE RETURN
1662  1020            LDA*20                  /GET LDF FOR PROGRAM STORAGE
1663  0000    C7TEMP, 0
1664  1120            ADA*20                  /SUBTRACT 40
1665  7737            7737
1666  1040            STA                     /STORE INSTRUCTION FIELD INSTRUCTION
1667  1534            MOVLIF
1670  1120            ADA*20                  /ADD 41
1671  0041            41
1672  1460            SAE*20                  /TOO FAR?
1673  0700    RDCCON, 700
1674  7677            LJMP    .+3             /NO
1675  1020            LDA*20                  /YES, FORM LDF2
1676  0042            LDF     2
1677  5535            STC     MOVLDF          /STORE DATA FIELD INSTRUCTION
1700  7700    C7EXIT, LJMP    .               /EXIT
```

```
                /SUBROUTINE TO HANDLE "I" BIT OF MAG TAPE INSTRUCTION
                /(CHECKS TAPE MOTION AFTER INSTRUCTION EXECUTION)
                /RETURNS TO LOC+1 IF ALL OK, OTHERWISE...

1701  1000      CHECKI, LDA                     /GET CONTENTS OF 0
1702  0000              0
1703  5732              STC     CIEXIT          /SET UP EXIT LOCATION
1704  2641              ADD     K1400
1705  4003              STC     3
1706  0223              XSK+20  3               /SHORT DELAY
1707  7706              LJMP    .-1
1710  3572              ADD     MTINST
1711  0325              ROR+20  5               /MOVE "I" BIT INTO LINK
1712  1020              LDA+20                  /SET UP AC
1713  5000              5000
1714  0500              IOB                     /TO
1715  6151              6151                    /LOAD THE TAPE MAINT. REG
1716  4000      K4000,  STC     0               /CLEAR AC.
1717  0500              IOB
1720  6154              6154                    /READ-UNITS AND MOTION INTO AC
1721  1560              BCL+20                  /MASK TO MOTION BIT
1722  7767              7767
1723  0452              LZE                     /LINK=0?
1724  7730              LJMP    .+4             /NO. EXPECT MOTION TO EQUAL 1
1725  0450              AZE                     /YES, DOES MOTION=0?
1726  7733              LJMP    XXX             /NO, ERROR *** ER 6 ***
1727  7732              LJMP    CIEXIT          /YES, OK
1730  0470              AZE+20                  /DOES MOTION=1
1731  7733              LJMP    XXX             /NO ERROR *** ER 6 ***
1732  7732      CIEXIT, LJMP    .               /YES, EXIT ROUTNE

                /COMMON ERROR HALT SUBROUTINE

1733  4004      XXX,    STC     XXXAC
1734  0500              IOB
1735  6002              IOF                     /DISABLE INTERRUPTS
1736  2000              ADD     0
1737  5757              STC     XXXPC
1740  0516              RSW                     /READ RIGHT SWITCHES
1741  0241              ROL     1
1742  0451              APO                     /RSW 1=1
1743  7750              LJMP    XXR             /YES DELETE TYPE OUT
1744  4000              STC                     /NO, TYPE OUT THE MESSAGE
1745  3757              ADD     XXXPC
1746  0601              LIF     1
1747  7024              LJMP    XX
1750  0516      XXR,    RSW                     /READ RIGHT SWITCHES
1751  0451              APO
1752  6204              LJMP    RESTAR          /ESCAPE TO RESTART ADDRESS
1753  1000              LDA
1754  0004              XXXAC
1755  0000              LHLT
1756  4000              STC
17    7757      XXXPC,  LJMP    .
```

```
                    /COMMON ROUTINE TO SUBTRACT
                    / 1 FROM THE NUMBER IN THE AC

1760  5764    SUBT1,  STC     .+4
1761  0011            CLR
1762  0017            COM
1763  1220            LAM+20
1764  0000            0
1765  6000            LJMP    0


                    /A ROUTINE TO SET BIT "7" OF THE
                    /TAPE INSTRUCTION

1766  1000    MPAC,   LDA
1767  1572            MTINST
1770  1620            BSE+20
1771  0020            0020
1772  5572            STC     MTINST
1773  6000            LJMP    0



      2007    .2007
2007  0000    XAC,    0000                    /STORAGE
```

```
                2020    *2020

                        /THIS SECTION OF CODING HANDLES SOME OF THE CALCULATIONS
                        /COMMON TO THE "READ AND CHECK GROUP" AND "WRITE AND CHECK GROUP" INSTRUCTIONS
                        /IN PARTICULAR, THIS SECTION CHECKS FOR:
                        /       1)  EXTENDED ADDRESSING MODE
                        /       2)  TAPE OR MEMORY WRAP-AROUND
                        /COMPUTES STARTING ADDRESS OF "LITTLE PROGRAM"
                        /AND CHECKS FOR TRANSFER INTO NON-EXISTANT MEMORY
2020  1000      COMON1, LDA
2021  0000              0
2022  4135              STC     C1EXIT-2000
2023  0640              LDF     0
2024  1100              ADA                     /GET XOB WORD
2025  2026              XOBWD-2000
2026  1560              BCL-20                  /MASK TO EXTENDED ADDRESS BIT
2027  7757              7757
2030  0450              AZE                     /BIT SET?
2031  6133              LJMP    C1EXIT-2        /YES, ILLEGAL OPERATION
2032  1000              LDA                     /GET QN-BN
2033  2032              QNBN-2000
2034  1560              BCL-20                  /CLEAR TO QUARTER BITS
2035  0777              777
2036  0243              ROL     3               /3 LEFT
2037  1040              STA                     /SAVE
2040  2033              CTEM1-2000
2041  1000              LDA                     /GET QN-BN
2042  2032              QNBN-2000
2043  1560              BCL-20                  /CLEAR TO BLOCK BITS
2044  7770              7770
2045  4102              STC     CTEM2-2000      /SAVE
2046  2102              ADD     CTEM2
2047  0470              AZE-20                  /NON-ZERO?
2050  6061              LJMP    CCON1           /NO, ZERO
2051  1100              ADA                     /YES, COMBINE WITH QUARTER BITS
2052  2033              CTEM1-2000
2053  1120              ADA-20                  /ADD-7
2054  7770              7770
2055  0471              APO-20                  /QN+BN<10?
2056  6133              LJMP    C1EXIT-2        /NO, EXIT
2057  0011              CLR
2060  6077              LJMP    CCON2-2         /SET UP STARTING ADDRESS OF PROGRAM OF 0
2061  1100      CCON1,  ADA                     /GET QN
2062  2033              CTEM1-2000
2063  1120              ADA-20                  /ADD-2
2064  7775              7775
2065  0451              APO                     /QN<3
2066  6075              LJMP    CCON2           /YES, SET UP STARTING ADDRESS OF PROGRAM OF 1400
2067  1000              LDA                     /NO, QN-3 OR MORE
2070  2026              XOBWD-2000              /GET XOB WORD
2071  1620              BSE-20                  /SET "DO NOT PAUSE" BIT
2072  0010              10
2073  1040              STA                     /STORE BACK
2074  2026              XOBWD-2000
```

```
2075  1020    CCON2,  LDA+20
2076  1400            1400
2077  1040            STA                     /SET UP STARTING ADDRESS OF PROGRAM
2100  2035            CSTART+2000
2101  1020            LDA+20                  /GET BN BITS 9-11
2102  0000    CTEM2,  0
2103  1100            ADA                     /ADD ON BITS
2104  2033            CTEM1+2000
2105  1120            ADA+20                  /ADD-3
2106  7774            7774
2107  0451            APO                     /4 OR GREATER?
2110  6127            LJMP    CCON3           /NO, ALL OK
2111  1000            LDA                     /GET CONTENTS OF FIELD
2112  2027            FIELDN+2000
2113  2131            ADD     K0001A-2000     ADD 1
2114  4136            STC     CTEM4-2000      /STORE
2115  0516            RSW                     /READ RIGHT SWITCHES
2116  1560            BCL+20                  /MASK TO EXTENDED MEMORY SWITCHES
2117  7770            7770
2120  0242            ROL     2               /2 LEFT
2121  1120            ADA+20                  /ADD 3
2122  0003    K0003A, 3
2123  0017            COM                     /MAKE MINUS
2124  2136            ADD     CTEM4           /COMBINE WITH "FIELD+1"
2125  0471            APO+20                  /DOES NEXT FIELD EXIST?
2126  6133            LJMP    C1EXIT-2        /NO, EXIT
2127  1020    CCON3,  LDA+20                  /INCREMENT EXIT LOCATION
2130  0001    K0001A, 1
2131  1140            ADM
2132  0135            C1EXIT+2000
2133  0041            LDF     1
2134  0000            LIF     0
2135  6135    C1EXIT, LJMP    .               /EXIT
2136  0000    CTEM4,  0                       /TEMP STORAGE OF UPPER MEMORY BANK NUMBERS
```

```
                /THIS SECTION OF CODING HANDLES SOME OF THE CALCULATI
                /COMMON TO THE "READ AND CHECK GROUP" AND "WRITE AND CHE  GROUP" INSTRUCTIONS
                /IN PARTICULAR, THIS SECTION SETS UP TO COUNT BLOCKS BY
                /SETTING UP 14 TO COUNT, CTEM3 TO BN 3 TO 11, AND EXITS WITH BN3 TO 11 IN AC
2137  1000      COMON2, LDA
2140  0000              0
2141  4161              STC     C2EXIT-2000
2142  0640              LDF     0
2143  1100              ADA                     /GET QN BITS
2144  2033              CTEM1-2000
2145  0017              COM
2146  6757              LJMP    SUBT1A          /SUBTRACT 1
2147  1040              STA                     /STORE IN 14
2150  2014              2014
2151  1000              LDA                     /GET QN-BN
2152  2032              QNBN-2000
2153  1560              BCL-20                  /MASK TO BN BITS 3 TO 11
2154  7000              7000
2155  1040              STA                     /STORE
2156  2034              CTEM3-2000
2157  0641              LDF     1
2160  0600              LIF     0
2161  6161      C2EXIT, LJMP    .               /EXIT

                /THIS SECTION OF CODING HANDLES SOME OF THE CALCULATIONS
                /COMMON TO THE "READ AND CHECK GROUP" AND "WRITE AND CHECK GROUP" INSTRUCTIONS
                /IN PARTICULAR, THIS SECTION DETERMINES THE DATA FIELD INSTRUCTION
                /TO ACCESS DATA IN MEMORY (FOR EITHER STORAGE OR CHECKING)
                /ENTER WITH BLOCK NUMBER IN AC
                /EXIT WITH "LDF" INSTRUCTION IN AC

2162  4207      COMON3, STC     C3TEMA-2000     /SAVE AC
2163  2000              ADD     0
2164  4206              STC     C3EXIT-2000     /SAVE EXIT ADDRESS
2165  0640              LDF     0
2166  2207              ADD     C3TEMA          /GET BLOCK NUMBER
2167  1560              BCL-20                  /MASK TO BN 9-11
2170  7770              7770
2171  1120              ADA-20                  /ADD -3
2172  7774              7774
2173  0451              APO                     /BN<4
2174  6200              LJMP    .+4             /YES
2175  1000              LDA                     /NO, GET LDF INSTRUCTION
2176  3535              MOVLDF-2000
2177  6204              LJMP    C3EXIT-2
2200  1000              LDA                     /GET LIF INSTRUCTION
2201  3534              MOVLIF-2000
2202  1120              ADA-20                  /ADD 40 TO MAKE LDF
2203  0040              40
2204  0641              LDF     1
2205  0600              LIF     0
2206  6206      C3EXIT, LJMP    .               /EXIT
2207  0000      C3TEMA, 0                       /TEMP STORAGE
```

```
                /THIS SECTION OF CODING HANDLES SOME OF THE CALCULATIONS
                /COMMON TO THE "READ" AND "WRITE" SUBROUTINES
                /IN PARTICULAR, THIS ROUTINE:
                /       1)  SETS UP THE DATA ADDRESS FOR EXTENDED ADDRESS MODE ADDRESSING
                /       2)  CALCULATES THE DATA FIELD INSTRUCTION FOR ACCESSING IT

2210  1000      COMON4, LDA
2211  0000              0
2212  4235              STC     C4EXIT-2000
2213  0640              LDF     0
2214  1000              LDA                     /GET WORD4
2215  2024              WD4+2000
2216  1560              BCL+20                  /MASK TO "ADDRESS BITS"
2217  6000              6000
2220  1040              STA                     /SAVE DATA ADDRESS
2221  2046              DATADD+2000
2222  1000              LDA                     /GET WORD 4 AGAIN
2223  2024              WD4+2000
2224  1560              BCL+20                  /MASK TO "EXT FIELD" BITS (0,1)
2225  1777              1777
2226  0242              ROL     2               /2 LEFT
2227  1100              ADA                     /COMBINE WITH OTHER FIELD BITS
2230  2027              FIELDN+2000
2231  1120              ADA+20                  /COMBINE WITH BASIC "LDF"
2232  0640              LDF
2233  0641              LDF     1
2234  0600              LIF     0
2235  6235      C4EXIT, LJMP    .               /EXIT

                /THIS SECTION OF CODING HANDLES SOME OF THE CALCULATIONS
                /COMMON TO THE "WRITE" AND "WRCKGP" SUBROUTINES
                /IN PARTICULAR, THIS ROUTINE COMPUTES AN ADDRESS FOR AN STA
                /FOR PATTERN WORD STORAGE FOR A PARTICULAR BLOCK ON A PARTICULAR TAPE
                /ENTER WITH BLOCK NUMBER IN AC
                /EXIT WITH ADDRESS IN AC

2236  1120      COMON9, ADA+20                  /SUBTRACT 770
2237  7007              7007
2240  4246              STC     C4TEMA-2000     /SAVE
2241  0640              LDF     0
2242  1000              LDA                     /GET UNIT NUMBER
2243  2025              UNIT+2000
2244  0241              ROL     1               /1 LEFT
2245  1120              ADA+20                  /ADD IN "TRIMMED BLOCK NUMBER"
2246  0000      C4TEMA, 0
2247  1120              ADA+20                  /ADD IN TABLE ENTRY ADDRESS
2250  3200              BLKTBL
2251  0641              LDF     1
2252  0600              LIF     0
2253  6000              LJMP    0               /EXIT
```

```
                        /TAPE 4
                        /SUBROUTINE TO PUT A PATTERN IN MEMORY
                        /SUBROUTINE IS ENTERED WITH ADDRESS FOR STORAGE IN THE AC
                        /SUBROUTINE EXITS WITH "PATTERN ADDRESS" IN AC AS A "LJMP ZZZ"
                        /DATA FIELD IS SET PREVIOUS TO ENTERING THIS ROUTINE

2254  4274  PATERN,  STC     PSAVE-2000      /SAVE STORAGE ADDRESS
2255  0006            DJR
2256  1020            LDA+20                  /GET NEXT PATTERN ADDRESS
2257  6303  PATPNT,  LJMP    PAT1
2260  4276            STC     PJMP-2000       /STORE IN JUMP LOCATION
2261  1020            LDA+20                  /INCREMENT PATTERN POINTER
2262  0002            2
2263  1140            ADM
2264  0257            PATPNT-2000
2265  1460            SAE+20                  /GONE TOO FAR?
2266  6335            LJMP    ZEROES
2267  6273            LJMP    .+4             /NO
2270  1020            LDA+20                  /YES, RESET
2271  6303            LJMP    PAT1
2272  4257            STC     PATPNT-2000
2273  1020            LDA+20                  /GET STORAGE ADDRESS
2274  0000  PSAVE,    0                       /SAVED ADDRESS
2275  0006            DJR
2276  6276  PJMP,     LJMP    .               /JMP THERE
2277  2276            ADD     PJMP            /PICKUP THE JUMP
2300  0641            LDF     1
2301  0600            LIF     0
2302  6302  PEXIT,    LJMP    .               /EXIT
```

```
2303  0006    PAT1,   DJR
2304  6335            LJMP    ZEROES          /ZEROS STORED
2305  0006            DJR
2306  6355            LJMP    ONES            /ONES STORED
2307  0006            DJR
2310  6377            LJMP    ZERONE          /ZEROES AND ONES STORED
2311  0006            DJR
2312  6422            LJMP    ONEZER          /ONES AND ZEROES STORED
2313  0006            DJR
2314  6443            LJMP    SEVZER          /7070 STORED
2315  0006            DJR
2316  6466            LJMP    ZERSEV          /0707 STORED
2317  0006            DJR
2320  6511            LJMP    SEVALT          /7070, 0707 ALTERNATING STORED
2321  0006            DJR
2322  6535            LJMP    ZERALT          /0707, 7070 ALTERNATING STORED
2323  0006            DJR
2324  6561            LJMP    FIVTWO          /5252 STORED
2325  0006            DJR
2326  6604            LJMP    TWOFIV          /2525 STORED
2327  0006            DJR
2330  6627            LJMP    FIVALT          /5252, 2525 ALTERNATING STORED
2331  0006            DJR
2332  6653            LJMP    TWOALT          /2645, 5132 ALTERNATING STORED
2333  0006            DJR
2334  6677            LJMP    COUNT           /COUNT PATTERN STORED

                /STORE ZEROES

2335  4341    ZEROES, STC     .+4-2000        /SUBTRACT 1
2336  0011            CLR
2337  0017            COM
2340  1220            LAM+20
2341  0000            0
2342  1620            BSE+20                  /SET DATA FIELD BIT
2343  2000            2000
2344  4006            STC     6               /SET POINTER
2345  2000            ADD     0               /SAVE RETURN ADDRESS
2346  4302            STC     PEXIT-2000
2347  0067            SET+20  7               /SET 7 TO -400
2350  7377            7377
2351  6765            LJMP    TST             /STORE
2352  0227            XSK+20  7               /COUNT
2353  6351            LJMP    .-2             /LOOP
2354  6277            LJMP    PJMP+1          /EXIT
```

```
                    /STORE ONES

2355  4361  ONES,   STC     .+4-2000        /SUBTRACT 1
2356  0011          CLR
2357  0017          COM
2360  1220          LAM+20
2361  0000          0
2362  1620          BSE+20                  /SET DATA FIELD BIT
2363  2000          2000
2364  4006          STC     6               /SET POINTER
2365  2000          ADD     0               /SAVE RETURN ADDRESS
2366  4302          STC     PEXIT-2000
2367  0067          SET+20  7               /SET 7 TO -400
2370  7377          7377
2371  0017          COM                     /SET AC TO 7777
2372  6765          LJMP    TST             /STORE
2373  0227          XSK+20  7               /COUNT
2374  6372          LJMP    .-2             /LOOP
2375  0011          CLR                     /CLEAR AC
2376  6277          LJMP    PJMP+1          /EXIT

                    /STORE ZEROES AND ONES ALTERNATELY

2377  4403  ZERONE, STC     .+4-2000        /SUBTRACT 1
2400  0011          CLR
2401  0017          COM
2402  1220          LAM+20
2403  0000          0
2404  1620          BSE+20                  /SET DATA FIELD BIT
2405  2000          2000
2406  4006          STC     6               /SET POINTER
2407  2000          ADD     0               /SAVE RETURN ADDRESS
2410  4302          STC     PEXIT-2000
2411  0067          SET+20  7               /SET 7 TO -400
2412  7377          7377
2413  0456          LSKP                    /SKIP WITH 0000 AC
2414  0017          COM                     /COMPLEMENT AC
2415  6765          LJMP    TST             /STORE
2416  0227          XSK+20  7               /COUNT
2417  6414          LJMP    .-3             /LOOP
2420  0011          CLR                     /CLEAR AC
2421  6277          LJMP    PJMP+1          /EXIT
```

```
                    /STORE ONES AND ZEROES ALTERNATELY

2422  4426  ONEZER, STC     .+4-2000        /SUBTRACT 1
2423  0011          CLR
2424  0017          COM
2425  1220          LAM+20
2426  0000          0
2427  1620          BSE+20                  /SET DATA FIELD BIT
2430  2000          2000
2431  4006          STC     6               /SET POINTER
2432  2000          ADD     0               /SAVE RETURN ADDRESS
2433  4302          STC     PEXIT-2000
2434  0067          SET+20  7               /SET 7 TO -400
2435  7377          7377
2436  0017          COM                     /COMPLEMENT AC
2437  6765          LJMP    TST             /STORE
2440  0227          XSK+20  7               /COUNT
2441  6436          LJMP    .-3             /LOOP
2442  6277          LJMP    PJMP+1

                    /STORE 7070

2443  4447  SEVZER, STC     .+4-2000        /SUBTRACT 1
2444  0011          CLR
2445  0017          COM
2446  1220          LAM+20
2447  0000          0
2450  1620          BSE+20                  /SET DATA FIELD BIT
2451  2000          2000
2452  4006          STC     6               /SET POINTER
2453  2000          ADD     0               /SAVE RETURN ADDRESS
2454  4302          STC     PEXIT-2000
2455  0067          SET+20  7               /SET 7 TO -400
2456  7377          7377
2457  1020          LDA+20                  /SET AC TO 7070
2460  7070          7070
2461  6765          LJMP    TST             /STORE
2462  0227          XSK+20  7               /COUNT
2463  6461          LJMP    .-2             /LOOP
2464  0011          CLR                     /CLEAR AC
2465  6277          LJMP    PJMP+1          /EXIT
```

```
                    /STORE 0707

2466  4472  ZERSEV, STC     .+4-2000        /SUBTRACT 1
2467  0011          CLR
2470  0017          COM
2471  1220          LAM+20
2472  0000          0
2473  1620          BSE+20                  /SET DATA FIELD BIT
2474  2000          2000
2475  4006          STC     6               /SET POINTER
2476  2000          ADD     0               /SAVE RETURN ADDRESS
2477  4302          STC     PEXIT-2000
2500  0067          SET+20  7               /SET 7 TO -400
2501  7377          7377
2502  1020          LDA+20                  /SET AC TO 0707
2503  0707          0707
2504  6765          LJMP    TST             /STORE
2505  0227          XSK+20  7               /COUNT
2506  6504          LJMP    .-2             /LOOP
2507  0011          CLR                     /CLEAR AC
2510  6277          LJMP    PJMP+1          /EXIT

                    /STORE 7070,0707 ALTERNATING

2511  4515  SEVALT, STC     .+4-2000        /SUBTRACT 1
2512  0011          CLR
2513  0017          COM
2514  1220          LAM+20
2515  0000          0
2516  1620          BSE+20                  /SET DATA FIELD BIT
2517  2000          2000
2520  4006          STC     6               /SET POINTER
2521  2000          ADD     0               /SAVE RETURN ADDRESS
2522  4302          STC     PEXIT-2000
2523  0067          SET+20  7               /SET 7 TO -400
2524  7377          7377
2525  1020          LDA+20                  /SET AC TO 0707
2526  0707          0707
2527  0017          COM                     /COMPLEMENT AC
2530  6765          LJMP    TST             /STORE
2531  0227          XSK+20  7               /COUNT
2532  6527          LJMP    .-3             /LOOP
2533  0011          CLR                     /CLEAR AC
2534  6277          LJMP    PJMP+1          /EXIT
```

```
                        /STORE 0707,7070 ALTERNATING

2535  4541      ZERALT, STC     .+4-2000        /SUBTRACT 1
2536  0011              CLR
2537  0017              COM
2540  1220              LAM+20
2541  0000              0
2542  1620              BSE+20                  /SET DATA FIELD BIT
2543  2000              2000
2544  4006              STC     6               /SET POINTER
2545  2000              ADD     0               /SAVE RETURN ADDRESS
2546  4302              STC     PEXIT-2000
2547  0067              SET+20  7               /SET 7 TO -400
2550  7377              7377
2551  1020              LDA+20                  /SET AC TO 7070
2552  7070              7070
2553  0017              COM                     /COMPLEMENT AC
2554  6765              LJMP    TST             /STORE
2555  0227              XSK+20  7               /COUNT
2556  6553              LJMP    .-3             /LOOP
2557  0011              CLR                     /CLEAR AC
2560  6277              LJMP    PJMP+1          /EXIT

                        /STORE 5252

2561  4565      FIVTWO, STC     .+4-2000        /SUBTRACT 1
2562  0011              CLR
2563  0017              COM
2564  1220              LAM+20
2565  0000              0
2566  1620              BSE+20                  /SET DATA FIELD BIT
2567  2000              2000
2570  4006              STC     6               /SET POINTER
2571  2000              ADD     0               /SAVE RETURN ADDRESS
2572  4302              STC     PEXIT-2000
2573  0067              SET+20  7               /SET 7 TO -400
2574  7377              7377
2575  1020              LDA+20                  /SET AC TO 5252
2576  5252              5252
2577  6765              LJMP    TST             /STORE
2600  0227              XSK+20  7               /COUNT
2601  6577              LJMP    .-2             /LOOP
2602  0011              CLR                     /CLEAR AC
2603  6277              LJMP    PJMP+1          /EXIT
```

```
                    /STORE 2525

2604  4610    TWOFIV, STC     .+4-2000        /SUBTRACT 1
2605  0011            CLR
2606  0017            COM
2607  1220            LAM+20
2610  0000            0
2611  1620            BSE+20                  /SET DATA FIELD BIT
2612  2000            2000
2613  4006            STC     6               /SET POINTER
2614  2000            ADD     0               /SAVE RETURN ADDRESS
2615  4302            STC     PEXIT-2000
2616  0067            SET+20  7               /SET 7 TO -400
2617  7377            7377
2620  1020            LDA+20                  /SET AC TO 2525
2621  2525            2525
2622  6765            LJMP    TST             /STORE
2623  0227            XSK+20  7               /COUNT
2624  6622            LJMP    .-2             /LOOP
2625  0011            CLR                     /CLEAR AC
2626  6277            LJMP    PJMP+1          /EXIT

                    /STORE 5252,2525 ALTERNATING

2627  4633    FIVALT, STC     .+4-2000        /SUBTRACT 1
2630  0011            CLR
2631  0017            COM
2632  1220            LAM+20
2633  0000            0
2634  1620            BSE+20                  /SET DATA FIELD BIT
2635  2000            2000
2636  4006            STC     6               /SET POINTER
2637  2000            ADD     0               /SAVE RETURN ADDRESS
2640  4302            STC     PEXIT-2000
2641  0067            SET+20  7               /SET 7 TO -400
2642  7377            7377
2643  1020            LDA+20                  /SET AC TO 2525
2644  2525            2525
2645  0017            COM                     /COMPLEMENT AC
2646  6765            LJMP    TST             /STORE
2647  0227            XSK+20  7               /COUNT
2650  6645            LJMP    .-3             /LOOP
2651  0011            CLR                     /CLEAR AC
2652  6277            LJMP    PJMP+1          /EXIT
```

```
                        /STORE 2645, 5132 ALTERNATING

2653  4657    TWOALT, STC     .+4-2000        /SUBTRACT 1
2654  2011            CLR
2655  0017            COM
2656  1220            LAM+20
2657  0000            0
2660  1620            BSE+20                  /SET DATA FILED BIT
2661  2000            2000
2662  4006            STC     6               /SET POINTER
2663  2000            ADD     0               /SAVE RETURN ADDRESS
2664  4302            STC     PEXIT-2000
2665  0067            SET+20  7               /SET 7 TO -400
2666  7377            7377
2667  1020            LDA+20                  /SET AC TO 5132
2670  5132            5132
2671  0017            COM                     /COMPLEMENT AC
2672  6765            LJMP    TST             /STORE
2673  0227            XSK+20  7               /COUNT
2674  6671            LJMP    .-3             /LOOP
2675  0011            CLR                     /CLEAR AC
2676  6277            LJMP    PJMP+1          /EXIT

                        /STORE COUNT PATTERN

2677  4703    COUNT,  STC     .+4-2000        /SUBTRACT 1
2700  0011            CLR
2701  0017            COM
2702  1220            LAM+20
2703  0000            0
2704  1620            BSE+20                  /SET DATA FIELD BIT
2705  2000            2000
2706  4006            STC     6               /SET POINTER
2707  2000            ADD     0               /SAVE RETURN ADDRESS
2710  4302            STC     PEXIT-2000
2711  0067            SET+20  7               /SET 7 TO -400
2712  7377            7377
2713  1120            ADA+20                  /INCREMENT AC
2714  0001            1
2715  6765            LJMP    TST             /STORE
2716  0227            XSK+20  7               /COUNT
2717  6713            LJMP    .-4             /LOOP
2720  0011            CLR                     /CLEAR AC
2721  6277            LJMP    PJMP+1          /EXIT
```

```
                /SUBROUTINE TO CHECK TO SEE IF BLOCK "N" HAS BEEN WRI  N INTO
                /"N" IS IN AC, TAPE DRIVE NUMBER IS IN LOCATION "UNIT"
                /ROUTINE EXITS TO LJMP+1 IF UNWRITTEN, LJMP+2 IF WRITTEN

2722  4756      WRITEN, STC     WSAVE-2000      /SAVE AC
2723  2000              ADD     0               /GET CONTENTS OF 0
2724  4755              STC     WNEXIT-2000     /AND SAVE
2725  0640              LDF     0
2726  2756              ADD     WSAVE           /GET BLOCK NUMBER
2727  1120              ADA+20                  /SUBTRACT 770
2730  7007              7007
2731  4756              STC     WSAVE-2000      /SAVE
2732  1000              LDA                     /GET UNIT NUMBER
2733  2025              UNIT+2000
2734  0241              ROL     1               /ROTATE 1 LEFT
2735  2756              ADD     WSAVE           /ADD IN "TRIMMED" BLOCK NUMBER
2736  1120              ADA+20                  /ADD IN TABLE ENTRY ADDRESS
2737  3200              ADD     BLKTBL
2740  4741              STC     GET-2000        /STORE AWAY
2741  2741  GET,        ADD     .               /GET CONTENTS OF BLOCK STATUS WORD
2742  4756              STC     WSAVE-2000
2743  2756              ADD     WSAVE
2744  0470              AZE+20                  /NON-ZERO?
2745  6753              LJMP    WNEXIT-2        /NO, ZERO, EXIT
2746  1020              LDA+20                  /YES, INCREMENT EXIT POINT
2747  0001              1
2750  2755              ADD     WNEXIT          /THEN
2751  4755              STC     WNEXIT-2000
2752  2756              ADD     WSAVE           /GET STATUS WORD
2753  0641              LDF     1
2754  0600              LIF     0
2755  6755  WNEXIT,     LJMP    .               /EXIT
2756  0000  WSAVE,      0


                /SUBROUTINE TO SUBTRACT 1
2757  4763      SUBT1A, STC     SUBT1B-2000
2760  0011              CLR
2761  0017              COM
2762  1220              LAM+20
2763  0000      SUBT1B, 0
2764  6000              LJMP    0
```

```
                    /ROUTINE TO CHECK ACROSS LINK MEMORY PAGE BOUNDRY

2765  1066    TST,    STA+20  6
2766  5015            STC     SAV-2000        /SAVE A,C,
2767  0011            CLR
2770  2006            ADD     6
2771  1560    TST1,   BCL+20
2772  6000            6000
2773  0006            DJR
2774  1460            SAE+20                  /TEST FOR 17777
2775  1777            1777
2776  7014            LJMP    SAV-1           /NO, EXIT
2777  4001            STC     1
3000  0500            IOB                     /YES, CHANGE LDF ROUTINE
3001  6214            RDF
3002  0301            ROR     1
3003  1120            ADA+20
3004  0641            641
3005  0472            LEE+20                  /TEST LINK
3006  7012            LJMP    .+4             /READING RESTORE
3007  0640            LDF     0               /DATCHK LOCATION
3010  1040            STA
3011  2667            DATCHK+2000
3012  5013            STC     .+1-2000
3013  0000    SAVA,   0                       /CHANGE DATA FIELD
3014  1020            LDA+20                  /RESTORE A,C,
3015  0000    SAV,    0
3016  0006            DJR
3017  0472            LEE+20                  /READING OR WRITING
3020  6000            LJMP    0               /WRITING, EXIT TO THIS FIELD
3021  0600            LIF     0               /READING, EXIT TO FIELD 0
3022  6676            LJMP    DATINC
```

```
                3200    *3200
                        /BLOCK PATTERN TABLE
3200  0000      BLKTBL, 0
                3400    *BLKTBL+200
                        /DATA BUFFER = 400 LOCATIONS



                        /LINC INSTRUCTION DEFINITIONS

                2000    ADD=2000
                1100    ADA=1100
                1140    ADM=1140
                1200    LAM=1200
                1240    MUL=1240
                1000    LDA=1000
                1300    LDH=1300
                4000    STC=4000
                1040    STA=1040
                1340    STH=1340
                0240    ROL=0240
                0300    ROR=0300
                0340    SCR=0340
                0000    LHLT=0000
                0016    LNOP=0016
                0011    CLR=0011
                0040    SET=0040
                6000    LJMP=6000
                0006    DJR=0006
                0004    ESF=0004
                0024    SFA=0024
```

```
0005    QAC=0005
1540    BCL=1540
1600    BSE=1600
1640    BCO=1640
0017    COM=0017
1440    SAE=1440
1400    SHD=1400
0440    SNS=0440
0456    LSKP=0456
0450    AZE=0450
0451    APO=0451
0452    LZE=0452
0453    IBZ=0453
0454    FLO=0454
0455    QLZ=0455
0400    SXL=0400
0415    KST=0415
1500    SRO=1500
0200    XSK=0200
0014    ATR=0014
0015    RTA=0015
0100    SAM=0100
0140    DIS=0140
1740    DSC=1740
0516    RSW=0516
0517    LSW=0517
0500    IOB=0500
0600    LIF=0600
0640    LDF=0640
0702    RDE=0702
0700    RDC=0700
0701    RCG=0701
0706    WRI=0706
0704    WRC=0704
0705    WCG=0705
0707    CHK=0707
0703    MTB=0703
0001    AXO=0001
0021    XOA=0021
0023    TMA=0023
0416    STD=0416
0417    TWC=0417
0002    PDP=0002
6141    LINC=6141
0003    TAC=0003
```

```
                3145    *3145
                        /ROUTINE TO DISPLAY A MESSAGE
                        /ON THE VR14 DISPLAY

3145  0067      DDISP,  SET*20  7               /SET 7 TO
3146  1161              DDTABL-2000-1           /TABLE ENTRY ADDRESS
3147  0104              SAM     4               /SAMPLE CHANNEL 4
3150  4001              STC     1               /SAVE IN LOC 1
3151  0100              SAM     0               /SAMPLE CHANNEL 0
3152  1767              DSC*20  7               /DSC DISPLAY
3153  1000              LDA
3154  0007              7                       /LOAD THE A,C,
3155  1460              SAE*20                  /IS IT THE END?
3156  1177              TAG-2000
3157  7151              LJMP    .-6             /NO, RE-EXECUTE
3160  0600              LIF     0               /YES, EXIT
3161  7621              LJMP    TFLAG

                        /TABLE OF CURRENT VERSION OF THIS
                        /PROGRAM TO BE DISPLAYED

3162  4177      DDTABL, 4177                    /D
3163  3641              3641
3164  0000              0000                    /SPACE
3165  0000              0000
3166  4122              4122                    /3
3167  2651              2651
3170  0000              0000                    /SPACE
3171  0000              0000
3172  4177              4177                    /D
3173  3641              3641
3174  0000              0000                    /SPACE
3175  0000              0000
3176  5177              5177                    /B
3177  2651      TAG,    2651                    /END OF THE MESSAGE
```

```
                3024    .3024
                        /PDP-12 LINK MODE ERROR
                        /HANDLER

3024  0077      XX,     SET.20  17
3025  7773              7773
3026  1560              BCL.20
3027  6000              6000
3030  5045              STC     TEMP-2000
3031  1020              LDA.20
3032  0320              0320
3033  7121              LJMP    PRINTR
3034  1020              LDA.20
3035  0303              0303
3036  7121              LJMP    PRINTR
3037  3120              ADD     K240
3040  7121              LJMP    PRINTR
3041  3045              ADD     TEMP
3042  6757              LJMP    SUBT1A
3043  0243              ROL     3
3044  1060              STA.20
3045  0000      TEMP,   0000
3046  1560              BCL.20
3047  7770              7770
3050  1120              ADA.20
3051  0260              0260
3052  7121              LJMP    PRINTR
3053  3045              ADD     TEMP
3054  0237              XSK.20  17
3055  7043              LJMP    TEMP-2

3056  1020      CRLF,   LDA.20                  /LOAD THE A,C,
3057  0215              0215                    /WITH 0215
3060  7121              LJMP    PRINTR          /PRINT IT
3061  1020              LDA.20                  /LOAD THE A,C,
3062  0212              0212                    /WITH 0212
3063  7121              LJMP    PRINTR          /PRINT IT
3064  0600              LIF     0
3065  7750              LJMP    XXR
```

```
                        /TEST THE DONE FLAG IN 0 MODE

3066  5075    TTDF,   STC     .+9-2000
3067  1020            LDA+20
3070  0100            0100
3071  6500            IOB
3072  6151            6151
3073  0000            LHLT                    / *** ER 1 ***
3074  1020            LDA+20
3075  0000            0000
3076  0600            LIF     0
3077  6000            LJMP    0


                        /A ROUTINE TO BUFFER THE MTB BY 3 BLOCKS

3100  1060    TSIGN1, STA+20
3101  0000    TSIGN,  0
3102  0471            APO+20                  /TAC = ?
3103  0017            COM                     /NO COMPLEMENT IT
3104  2122            ADD     K0003A          /ADD 3
3105  0451            APO                     /WITHIN 3
3106  7113            LJMP    IBIT+1          /NO, ALL OK
3107  1000            LDA
3110  1101            TSIGN-2000              /XOR TSIGN
3111  1660            BCO+20                  /AND
3112  0000    IBIT,   0                       /IBIT
3113  0641            LDF     1
3114  0600            LIF     0
3115  0451            APO                     /BEYOND THE BLOCK ?
3116  7052            LJMP    MCCH            /NO, ALL OK, DO THE NEXT BLOCK
3117  7104            LJMP    MEXIT-1         /YES, FORGET IT
3120  0240    K240,   0240


3121  0002    PRINTR, PDP
3122  6046            6046
3123  6041            6041
3124  5323            JMP     .-1
3125  6042            6042
3126  7200            CLA
3127  6141            LINC
3130  6000            LJMP    0

3131  1020    BELL,   LDA+20
3132  0207            0207
3133  7121            LJMP    PRINTR
3134  0600            LIF     0
3135  6475            LJMP    INCRA-4
                      $
```

```
0000  01111000  00000000  11111111  11111111  11111111  11111111  11111111  11111111
0100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111100

0200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

0400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

0600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
0700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1000  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

1600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
1700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11110000

2000  00000001  00000000  11111111  11111111  11111111  11111111  11111111  11111111
2100  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

2200  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2300  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

2400  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2500  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

2600  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111
2700  11111111  11111111  11111111  11111111  11111111  11111111  11111111  11111111

3000  11111111  11111111  11101111  11111111  11111111  11111111  11111111  11111111
3100  11111111  11111111  11111111  11111100  00000111  11111111  11111111  11111111

3200  10000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000
3300  00000000  00000000  00000000  00000000  00000000  00000000  00000000  00000000

3400
3500

3600
3700
```

```
4000
4100

4200
4300

4400
4500

4600
4700


5000
5100

5200
5300

5400
5500

5600
5700


6000
6100

6200
6300

6400
6500

6600
6700


7000
7100

7200
7300

7400
7500

7600
7700
```

```
AC      0030    DATCHK  0667    LDFRC1  0742    PATFRM  0762
ADA     1100    DATINC  0676    LDFRC2  0761    PATJMP  0644
ADD     2000    DATLUP  0221    LDFRC3  1006    PATPNT  2257
ADM     1140    DATUM   0202    LDFWC1  1353    PAUSEB  0414
APO     0451    DDISP   3145    LDFWC2  1410    PDP     0002
ATR     0014    DDTABL  3162    LDFWR1  1125    PEXIT   2302
AXO     0001    DIS     0140    LDFWR2  1157    PJMP    2276
AZE     0450    DISPCH  0436    LDFWR3  1212    PRINTR  3121
BCL     1540    DJR     0006    LDH     1300    PSAVE   2274
BCO     1640    DSC     1740    LHLT    0000    Q3PAT   1426
BELL    3131    ESF     0004    LIF     0600    QAC     0005
BKWRD   0064    EXT0    0271    LINC    6141    QLZ     0455
BLKTBL  3200    EXT1    0300    LINTER  0040    QNBN    0032
BSE     1600    EXT2    0307    LJMP    6000    RANDOM  1637
C1EXIT  2135    EXT3    0316    LNOP    0016    RANXIT  1656
C2EXIT  2161    EXT4    0331    LOOP01  0104    RCB     0701
C3EXIT  2206    EXTDCH  0702    LSKP    0456    RCHK    0610
C3TEMA  2207    EXTEND  0260    LSW     0517    RCKSUB  0460
C4EXIT  2235    EXTUNT  0230    LZE     0452    RDC     0700
C4TEMA  2246    FIELDN  0027    M4000   0174    RDCCON  1673
C6EXIT  1607    FIVALT  2627    MAGTAP  0061    RDCKCP  0706
C7EXIT  1700    FIVTWO  2561    MASTER  0020    RDE     0702
C7TEMP  1663    FLO     0454    MBUMP   1066    RDSUB   0456
CCHK    1502    FOBWRD  0124    MCOM    1052    READ    0504
CCHKA   1504    FORWRD  0130    MCM     1044    REDDF   0563
CCON1   2061    GET     2741    MCMK    1061    REDNEX  0531
CCON2   2075    HALFX   1654    MCMK1   1101    REDNX1  0597
CCON3   2127    HALFY   1655    MCOMP   1075    RESTAR  0204
CEXIT   1510    IBIT    3112    MEXIT   1105    REXIT   0701
CHECK   1454    IBZE    0453    MEXPT   1076    RGCHK   0744
CHECKI  1701    INCR    0466    MOVE    1023    RGCON1  0715
CHK     0707    INCRA   0501    MOVJMP  1537    RGCON2  0726
CHKSUB  0464    IOB     0500    MOVLDF  1535    RGCON3  0753
CIEXIT  1732    K0001   1014    MOVLIF  1534    RGEXIT  1022
CLEAR   0161    K0001A  2130    MOVPRO  1511    RJUMP   1575
CLR     0011    K0002   0003    MOVSUB  0462    ROL     0240
COM     0017    K0003   0377    MPAC    1766    ROR     0300
COMON1  2020    K0003A  2122    MTB     0703    RSW     0516
COMON2  2137    K0100   0036    MTBTST  0062    RTA     0010
COMON3  2162    K0200   0037    MTEXIT  1566    SAE     1440
COMON4  2210    K0770   0411    MTINST  1572    SAM     0100
COMON5  2236    K1400   0641    MTSET   1540    SAV     3015
COMON6  1576    K240    3120    MTXEQT  1567    SAVA    3013
COMON7  1657    K4000   1716    MUL     1240    SCR     0340
COUNT   2677    K4001   0175    NONEX1  0356    SET     0040
CRLF    3056    KST     0415    NONEX2  0406    SEVALT  2511
CSTART  0035    LAM     1200    NONEX3  0413    SEVZER  2443
CTEM1   0033    LD1CON  0643    NONEXT  0344    SFA     0024
CTEM2   2102    LDA     1000    ONES    2355    SHD     1400
CTEM3   0034    LDF     0640    ONEZER  2422    SNS     0440
CTEM4   2136    LDFCON  0631    PAT1    2303    SRO     1500
DATADD  0646    LDFRD1  0561    PATERN  2254    STA     1040
```

```
STAC    0031            WRINX1  1210
STC     4000            WRITE   1106
STD     0416            WRITEN  2722
STH     1340            WSAVE   2756
SUBT1   1760            WTEMP   1300
SUBT1A  2757            XAC     2007
SUBT1B  2763            XOA     0021
SXL     0400            XOBWD   0026
TABLE1  0446            XSK     0200
TAC     0003            XX      3024
TAG     3177            XXR     1750
TDFLAG  1610            XXX     1733
TEMP    3045            XXXAC   0004
TFLAG   1621            XXXPC   1757
TLAG    1625            ZERALT  2335
TMA     0023            ZEROES  2335
TPINEN  0430            ZERONE  2377
TSIGN   3101            ZERSEV  2466
TSIGN1  3100
TST     2765
TST1    2771
TSTMOR  0042
TTDF    3066
TWC     0417
TWOALT  2653
TWOFIV  2604
UNBNSV  1322
UNIT    0025
WCG     0705
WCHK    1241
WCONT1  1301
WCONT2  1316
WD1     0021
WD2     0022
WD3     0023
WD4     0024
WEXIT   1331
WGCHK   1430
WGCON1  1343
WGEXIT  1453
WGPAT   1363
WGQNBN  1450
WGRET   1412
WINST   1305
WINST1  1263
WNEXIT  2755
WPAT    1317
WRC     0704
WRCKGP  1332
WRI     0706
WRIDF   1214
WRINEX  1143
```

ERRORS DETECTED: 0

LINKS GENERATED: 0

RUN-TIME: 18 SECONDS

3K CORE USED