.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-D2BMA-B-D
PRODUCT NAME:	MEMORY I/O EXERCISER
DATE CREATED:	1-DEC-72
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	BOB BRAIN

CONTENTS

1. ABSTRACT

THIS PROGRAM CHECKS BANK SELECTION, EA BITS, AND MEMORY USING ANY NPR DEVICE WITH EA BITS. IT RUNS STAND ALONE OR WITH KT11C OR KT11D TO ACCESS EXTENDED MEMORY. WORST CASE NOISE PATTERNS ARE USED WITH THE NPR DEVICE TO TEST THE MEMORY. IF THE PROGRAM IS STARTED BY ACT11, IT WILL CYCLE THROUGH ALL THE EXISTING DEVICES. IF IT IS STARTED BY DDP2 IN CHAIN MODE, IT WILL CYCLE THROUGH ALL THE DEVICES EXCEPT THE DDP2 DEVICE. IN ALL CASES, DRIVE 1 ON THE TC11 WILL BE USED.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OF 8K OF MEMORY
KT11C/D FOR MEMORY EXPANSION (OPTIONAL)

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

MEMORY DIAGNOSTICS
DEVICE DIAGNOSTICS

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

START AT 200.

4.3 PROGRAM AND/OR OPERATOR ACTION

1) START AT 200
2) TYPE DEVICE (RF11, RK11, RP11, RC11, TC11, TM11, DM11, OR DR11(B)) AND RETURN.
3) TEST WILL START (↑C WILL RETURN TO STEP 2 AND RESTORE THE LOADER)
OR
1) START AT 200
2) TYPE A RETURN
3) TYPE THE LOWER BANK TO BE TESTED ... 1=20000-37777 ETC.
4) TYPE THE UPPER BANK TO BE TESTED
5) TYPE DEVICE (RF11, RK11, RP11, RC11, TC11, TM11, DM11, OR DR11(B)) AND RETURN.
6) TEST WILL START (↑C WILL RETURN TO STEP 2 AND RESTORE THE LOADER)

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 ... ALL SWITCHES DOWN IS WORST CASE TESTING. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

SW(15) = 1 ..... HALT ON ERROR
SW(14) = 1 ..... HANG ON CURRENT BANK
SW(13) = 1 ..... INHIBIT PRINTOUT
SW(12) = 1 ..... INHIBIT BELL ON PASS COMPLETE
SW(09) = 1 ..... INHIBIT USE OF MEMORY EXPANSION DEVICE
SW(08) = 1 ..... TRACE BANK UNDER TEST

5.2 SUBROUTINE ABSTRACTS

5.2.1 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERUPTS WILL HALT AT THE VECTOR + 2.

5.2.2 TRAP HANDLER

MOST OF THE SUBROUTINE CALLS ARE DONE VIA A TRAP+N CALL. TO FIND THE SUBROUTINE BEING CALLED, LOOK IN THE COMMENT

```
200                                 SECTION OF THE TRAP DEFINATION TABLE FOR THE NAME OF THE
201                                 TRAP,  THEN SCAN TO THE LEFT MARGIN FOR THE STARTING ADDRESS
```

OF THE SUBROUTINE.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

RF11 ERROR: CS = 100305 ER = 001000

RF11 = DEVICE UNDER TEST
CS = CONTROL AND STATUS REGISTER
ER = ERROR REGISTER

DATA ERROR AT 401220 TRUE = 177777 RECEIVED = 177757

401120 = ADDRESS OF BAD DATA (18 BIT)
TRUE = DATA SENT
RECEIVED = DATA FOUND

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

TO RUN RP02, CHANGE LOC=5456 TO 000002 (RTI).

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE EXECUTATION TIME IS DEPENDANT ON THE AMOUNT OF MEMORY AND THE DEVICE USED. THE BELL SHOULD RING WITHIN 20 MINUTES (USING DRII AT 110 BAUD IN 124K.)

8.2 STACK POINTER

STACK IS INITALLY SET TO 500

8.3 POWER FAIL

AFTER THE PROGRAM IS STARTED, POWER DOWN THEN UP. THERE

```
  258                                   SHOULD BE NO ERROR TYPEOUTS.
```

9. PROGRAM DESCRIPTION

THE PROGRAM SIZES CORE AND TEST SEQUENTIALLY ALL CORE OR ANY SECTION OF CORE IN 4K CHUNKS USING THE DEVICE SPECIFIED AND MEMORY EXPANSION DEVICE. WORST CASE NOISE PATTERNS ARE USED FOR EACH MEMORY TYPE ON EACH PASS.

```
                        .
                        .
                        .
                        X
.TITLE  MAINDEC-11-DZQMA-B      MEMORY I/O EXERCISER
.REM%
Abstract

        This program checks bank selection, EA bits, and memory using
        any  NPR  device  with  EA  bits.  It runs stand alone or with
        KT11C or KT11D to access extended memory.  Worst  case  noise
        patterns are used with the NPR device to test the memory.


Requirements

        PDP-11 with  at  least  8K  of  memory  (KT11C  or  KT11D  are
        optional.)


Storage - First 4K

Loading - Absolute Loader


Execution time

        The time is dependent on the amount of memory and  the  device
        which  is used.  A bell will signify end of pass.  Maximum run
        time is no greater than 5 minutes.


Starting procedure

        1) Start at 200
        2) Type device (RF11, RK11, RP11, RC11, TC11, TM11,  DM11,  or
        DR11B) and RETURN.
        3) Test will start (^C will return to step 2 and  restore  the
        loader)
          or
        1) Start at 200
        2) Type a RETURN
        3) Type the lower bank to be tested ..  1 = 20000-37777 etc.
        4) Type the upper bank to be tested
        5) Type device (RF11, RK11, RP11, RC11, TC11, TM11,  DM11,  or
        DR11B) and RETURN.
        6) Test will start (^C will return to step 2 and  restore  the
        loader)


Switch register options - yes

        SW15 - HALT ON ERROR
        SW14 - LOOP ON CURRENT BANK
        SW13 - INHIBIT TYPEOUTS
        SW10 - INHIBIT BELL
        SW9  - INHIBIT USE OF MEMORY EXPANTION DEVICE
        SW8  - TYPE BANK UNDER TEST#
```

```
                                ;COPYRIGHT 1972,DIGITAL EQUIPMENT CORP., MAYNARD MASS.
                                ;PROGRAM BY BOB BRAIN

                 104400         SCOPE=  TRAP
                 104000         HLT=    EMT
                 000004         TYPE=   IOT
                 177776         PS=     177776
                 177570         SWR=    177570
                 177570         DISPLAY=SWR
                 000007         BELL=   7
                 000000         R0=     %0
                 000001         R1=     %1
                 000002         R2=     %2
                 000003         R3=     %3
                 000004         R4=     %4
                 000005         R5=     %5
                 000006         TTY=    %6
                 000006         SP=     %6
                 000007         PC=     %7
                 000000         .=      0
                 040000         SWI4=   40000
                 020000         SWI3=   20000
                 010000         SWI2=   10000
                 004000         SWI1=   4000
                 002000         SWI0=   2000
                 001000         SWR=    1000
                 000400         SWB=    400

                 000000         .=      0                       ;TRAP CATCHER FROM 0 - 776

                 000200         .=      200

000200  000137  001024          JMP     @#BEGIN                 ;JUMP TO BEGINING ADDRESS OF PROGRAM

                 001000         .=      1000

001000  000000                 ICNT:   0                       ;LH = ITERATION COUNT  ;RH = TEST NO.
001002  000000                 ERRORS: 0                       ;ERROR COUNT
001004  000000  000000          PCNT:   0,0                     ;2 WORD PASS COUNT
001010  000000                  LAD:    0                       ;LOOP ADDRESS FOR SCOPE
001012  000000                  HLTADR: 0                       ;LAST HLT INSTRUCTION ADD. EXECUTED
001014  000000                  MX:     0                       ;BANK UNDER TEST
001016  000000                  MIN:    0                       ;FIRST BANK
001020  000000                  LIMIT:  0                       ;LAST BANK
001022  000000                  OPTION: 0                       ;OPTION - MI = MX11 - GT = KT11
```

```
001024  000005                         BEGIN:  RESET                           ;ZAP THE WORLD
001026  105737  177564                         TSTB    @#177564                ;TTY READY
001032  100375                                 BPL     .-4                     ;NO - WAIT
001034  012706  000500                         MOV     #500,SP                 ;SET STACK TO  ## 500 ##
001040  012737  012104  000020                 MOV     #IOTS,@#20              ;LOAD IOT VECTOR
001046  012737  000340  000022                 MOV     #340,@#22               ;LOCK UP
001054  012737  012612  000024                 MOV     #PDOWNS,@#24            ;LOAD PF VECTOR
001062  012737  000340  000026                 MOV     #340,@#26               ;LOCK UP
001070  012737  002716  000034                 MOV     #TRAPS,@#34             ;SET FOR TRAP
001076  012737  000340  000036                 MOV     #340,@#36               ;LOCK UP
001104  012777  000240  010430                 MOV     #240,@RFTRAP+2
001112  012777  000240  010426                 MOV     #240,@RCTRAP+2
001120  012777  000240  010424                 MOV     #240,@RPTRAP+2
001126  012777  000240  010422                 MOV     #240,@RKTRAP+2
001134  012777  000300  010420                 MOV     #300,@TCTRAP+2
001142  012777  000240  010422                 MOV     #240,@TMTRAP+2
001150  012777  000240  010410                 MOV     #240,@DRTRAP+2
001156  012777  004232  010354                 MOV     #TRP.RF,@RFTRAP
001164  012777  004614  010352                 MOV     #TRP.RC,@RCTRAP
001172  012777  005324  010350                 MOV     #TRP.RP,@RPTRAP
001200  012777  005654  010346                 MOV     #TRP.RK,@RKTRAP
001206  012777  006230  010344                 MOV     #TRP.TC,@TCTRAP
001214  012777  007136  010346                 MOV     #TRP.TM,@TMTRAP
001222  012777  010514  010334                 MOV     #TRP.DR,@DRTRAP
001230  012767  012752  176622                 MOV     #READR,60               ;KEYBOARD VECTOR
001236  012767  140000  010474                 MOV     #140000,LOLIM           ;INIT LOLIM
001244  012767  160000  010470                 MOV     #160000,HILIM           ;INIT HILIM
001252  005067  177540                         CLR     MIN
001256  005067  177532                         CLR     MX
001262  012767  000003  010244                 MOV     #3,PSCNT                ;INIT PASS COUNT
001270  005067  002450                         CLR     MEMORY                  ;WHICH MEMORY BEING CHECKED
001274  005067  177522                         CLR     OPTION                  ;TYPE THE OPTION
001300  032737  001000  177570                 BIT     #SW9,@#SWR              ;INHIBIT USE OF MEMORY EXTENTION DEVICE
001306  001030                                 BNE     SIZEIT                  ;SKIP
001310  012737  001330  000004                 MOV     #1$,@#4                 ;SET FOR TIMEOUT
001316  005777  010376                         TST     @SR0                    ;CHECK FOR KT11
001322  105167  177474                         COMB    OPTION                  ;OPTION IS KT11
001326  000415                                 BR      2$

001330  012737  001362  000004         1$:     MOV     #2$,@#4                 ;SET FOR TIMEOUT
001336  005777  010374                         TST     @MXC3                   ;CHECK FOR MX11
001342  105167  177455                         COMB    OPTION+1                ;OPTION IS MX11
001346  012767  000003  177442                 MOV     #3,MIN
001354  012767  000003  177432                 MOV     #3,MX
001362  012737  000006  000004         2$:     MOV     #6,@#4                  ;RESET FOR NEM

001370  005767  177426                 SIZEIT: TST     OPTION                  ;WHICH OPTION AM I
001374  100417                                 BMI     DOMX                    ;MUST HAVE MX11
001376  001037                                 BNE     DOSEG                   ;MUST HAVE MEMORY MANAGEMENT
```

```
                001400  012737  001432  000004  DOCORE: MOV     #2$,2#4         ;SET FOR NEM
                001406  012701  017776                  MOV     #17776,R1       ;SET UP ADDRESS
                001412  005000                          CLR     R0              ;SET UP BANK COUNT
                001414  062701  020000          1$:     ADD     #20000,R1       ;MOVE TO NEXT BANK
                001420  005200                          INC     R0              ;INC THE BANK COUNT
                001422  005711                          TST     (1)             ;TIMEOUT?
                001424  022701  177776                  CMP     #177776,R1      ;END?
                001430  001371                          BNE     1$              ;LOOP IF NOT AT THE END
                001432  000446                  2$:     BR      TYPEIT          ;TYPE IT

                001434  012737  001466  000004  DOMX:   MOV     #2$,2#4         ;SET UP FOR NEM
                001442  012700  000003                  MOV     #3,R0           ;SET BANK COUNT
                001446  005200                  1$:     INC     R0              ;MOVE TO NEXT BANK
                001450  010077  010262                  MOV     R0,2MXC3        ;SET UP MXC3
                001454  005737  157776                  TST     @#157776        ;TIMEOUT?
                001460  022700  000036                  CMP     #36,R0          ;END?
                001464  001370                          BNE     1$              ;LOOP IF NOT END
                001466  012777  000006  010242  2$:     MOV     #6,2MXC3        ;MAP INTO SELF
                001474  000425                          BR      TYPEIT          ;TYPE IT

                001476  104444          DOSEG:  MAPIT                           ;SETUP MEMORY MANAGEMENT REGISTERS
                001500  005277  010214                  INC     @SR0            ;TURN ON MEMORY MANAGEMENT
                001504  012737  001534  000004          MOV     #2$,2#4         ;SET TIMEOUT ADDRESS FOR CORE CALCULATIONS
                001512  005737  157776          1$:     TST     @#157776        ;TRAP ON NON EX MEM
                001516  062777  000200  010202          ADD     #200,2KISAR6    ;GO TO NEXT BANK
                001524  022777  007600  010174          CMP     #7600,2KISAR6   ;LAST ONE?
                001532  003367                          BGT     1$              ;TRY NEXT
                001534  017700  010166          2$:     MOV     2KISAR6,R0      ;SAVE ASR6 IN R0
                001540  006300                          ASL     R0              ; #ASH #-7,R0#
                001542  000300                          SWAB    R0              ;KLUDGY ISN'T IT
                001544  042700  177740                  BIC     #177740,R0      ;CLEAR JUNK
```

```
                                        TYPEIT: DEC     R0              ;DROP BACK
                                                MOV     #6,@#4          ;SET FOR NEM
                                                MOV     #500,SP         ;CLEAR STACK ##500##
                                                INC     @#-1            ;TYPE THE OPTION ONLY ONCE
                                                BNE     REDO            ;FIRST TIME?
                                                JSR     7,LODGET        ;GET THE LOADER ONLY THE FIRST TIME
                                                TYPE    ,+2             ;.ASCIZ <15><12><12>"MAINDEC-11-DZQMA-B"<15><12><12>"BAN
                                                MOV     R0,TTY          ;TYPE R0 IN OCTAL
                                                JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                                TYPE    ,+2             ;.ASCIZ " EXIST"
                                                TST     OPTION          ;WHICH OPTION?
                                                BEQ     REDO            ;NOTHING
                                                BMI     1$              ;MXII
                                                TYPE    ,WITHSG         ;MEMORY MANAGEMENT
                                                BR      REDO            ;SKIP
                                        1$:     TYPE    ,WITHMX         ;MXII
                                        REDO:   MOV     R0,LIMIT        ;SAVE UPPER BANK
                                                JSR     7,LODRES        ;RESTORE LOADER
                                                TST     @#42            ;UNDER A MONITOR?
                                                BEQ     TD              ;NO
                                                CLR     @#PS            ;CLEAR PS
                                        RSIT:   MOV     #DEVTAB-2,DEVADR;SET FOR DEVICE SCAN
                                        ABRT:   ADD     #2,DEVADR       ;GO TO NEXT ONE
                                        NEMDEV: TST     @DEVADR         ;IS IT THE END?
                                                BNE     1$              ;YES
                                                JSR     PC,GETMON       ;GET THE MONITOR
                                                BR      RSIT            ;LOOP
                                        1$:     MOV     DEVADR,R3       ;GET ADDRESS
                                                SUB     #DEVTAB,R3      ;MAKE IT AN OFFSET
                                                CMPB    TAB1(3),@#41    ;SAME DEVICE?
                                                BEQ     ABRT            ;YES - ABORT DEVICE
                                                MOV     @DEVADR,INPUT   ;SET INPUT
                                                MOV     #'11,INPUT+2    ;FOR TYPING
                                                CLR     INPUT+4         ;MAKE TERMINATOR
                                                TYPE    ,+2             ;.ASCIZ <15><12>
                                                TYPE    ,INPUT          ;TYPE THE DEVICE
                                                MOV     #BADDEV,@#4     ;SET FOR TIMEOUT
                                                MOV     #TRP.BF,@#BFTRAP
                                                MOV     #TRP.BC,@#BCTRAP
                                                MOV     #TRP.BP,@#BPTRAP
                                                MOV     #TRP.BK,@#BKTRAP
                                                MOV     #TRP.TC,@#TCTRAP
                                                MOV     #TRP.TM,@#TMTRAP
                                                MOV     #TRP.DR,@#DRTRAP
                                                BR      RTPF            ;START IT

                                        BADDEV:
                                                TYPE    ,+2             ;.ASCIZ " - NONE"
                                                JMP     ABRT            ;GO TO NEXT DEVICE
```

B02


```
002142  012737  002560  000004  T0:    MOV     #NODEV,@#4          ;SET UP TRAP IF NO DEVICE
002150  005037  177776                 CLR     @#177776            ;DROP PRIORITY
[illegible] [illegible] 002160         TYPE    ,.+2                ;.ASCIZ <15><12>"#"
[illegible] [illegible] 010066         JSR     PC,READS            ;INPUT DATA FROM TTY
002170  012706  000500          RTPF:  MOV     #500,SP             ;RESET THE STACK   ## 500 ##
002174  [illegible] 000100  177560     BIS     #100,@#177560       ;IE THE TTY
[illegible] [illegible] 041522  010106 CMP     #"RC,INPUT          ;CHECK FOR RC11
[illegible] 001004                     BNE     T1
[illegible] [illegible] 000100  007376 BIS     #100,@RCCS          ;TURN ON INTERUPT
[illegible] [illegible]                BR      WAIT
[illegible] [illegible] 050122  010066  T1:    CMP     #"RP,INPUT   ;CHECK FOR RP11
[illegible] 001004                     BNE     T2
[illegible] [illegible] 000100  007370 BIS     #100,@RPCS          ;TURN ON INTERUPT
[illegible] [illegible]                BR      WAIT
[illegible] [illegible] 043122  010046  T2:    CMP     #"RF,INPUT   ;CHECK FOR RF11
[illegible] 001004                     BNE     T3
[illegible] [illegible] 000100  007324 BIS     #100,@RFCS          ;TURN ON INTERUPT
[illegible] [illegible]                BR      WAIT
[illegible] [illegible] 045522  010026  T3:    CMP     #"RK,INPUT   ;CHECK FOR RK11
[illegible] 001004                     BNE     .+4
[illegible] [illegible] 000100  007344 BIS     #100,@RKCS          ;TURN ON INTERUPT
[illegible] [illegible]                BR      WAIT
[illegible] [illegible] 046524  010006  T4:    CMP     #"TM,INPUT   ;CHECK FOR TM11
[illegible] 001004                     BNE     T5
[illegible] [illegible] 000100  007360 BIS     #100,@TMCS          ;TURN ON INTERUPT
[illegible] [illegible]                BR      WAIT
[illegible] [illegible] 041524  007766  T5:    CMP     #"TC,INPUT   ;CHECK FOR TC11
[illegible] 001004                     BNE     T6
[illegible] [illegible] 000100  007316 BIS     #100,@TCCS          ;TURN ON INTERUPT
[illegible] [illegible]                BR      WAIT
[illegible] [illegible] 046504  007746  T6:    CMP     #"DM,INPUT   ;CHECK FOR DM11
[illegible] 001007                     BNE     T7
[illegible] [illegible] 007322         TST     @DMCS               ;IS IT THERE?
[illegible] 012737  000006  000004     MOV     #6,@#4              ;RESTORE 4
[illegible] 000167  005126             JMP     TMP.DM
[illegible] [illegible] 051104  007720  T7:    CMP     #"DR,INPUT   ;CHECK FOR DR11
[illegible] 001004                     BNE     T8
[illegible] [illegible] 000100  007306 BIS     #100,@DRCS          ;TURN ON INTERUPT
[illegible] 000410                     BR      WAIT
[illegible] 105767  007702      T8:    TSTB    INPUT               ;CR ALONE
[illegible] 001420                     BEQ     CRLF
[illegible] 000004  010716             TYPE    ,.HI
[illegible] 005037  177560             CLR     @#177560            ;ID THE TTY
[illegible] 000646                     BR      T0

002420  012737  000006  000004  WAIT:  MOV     #6,@#4              ;SET NEW TRAP
[illegible] 016704  007276      1$:    MOV     LOLIM,R4            ;SET LOW LIMIT
[illegible] 026704  007274      2$:    CMP     HILIM,R4            ;END OF RANGE
[illegible] 001773                     BEQ     1$                  ;YES - RESET
[illegible] 000241                     CLC                         ;CLEAR CARRY BIT
[illegible] 005524                     ADC     (4)+                ;NOP WITH READ/WRITE
002464  000772                         BR      2$                  ;LOOP
```

```
##############
#            #
# BACKGROUND #
#    TEST    #
#            #
##############
```

```
002456  012737  000006  000004  CALF:   MOV     #6,@#4
[illegible]  005037  177560                  CLR     @#177560        ;TO THE TTY
002470  000004  011103                          TYPE    ,M11            ;ASK FOR LOWER LIMIT
002474  004767  007512                          JSR     PC,READS        ;GET IT
002500  005067  176312                          CLR     MIN
002504  012701  012316                          MOV     #INPUT,R1
002510  105711                          CR1:    TSTB    (1)             ;CR?
002512  001414                                  BEQ     CR2
002514  106011                                  RORB    (1)
002516  106011                                  RORB    (1)
002520  106011                                  RORB    (1)
[illegible]  106167  176270                      ROLB    MIN
[illegible]  106111                              ROLB    (1)
[illegible]  106167  176262                      ROLB    MIN
[illegible]  106121                              ROLB    (1)+
[illegible]  106167  176254                      ROLB    MIN
[illegible]  000762                              BR      CR1
[illegible]  005767  176246              CR2:    TST     MIN             ;MIN MUST BE >0
[illegible]  003742                              BLE     CALF            ;NOPE!
[illegible]  005367  176240                      DEC     MIN
[illegible]  016767  176234  176230              MOV     MIN,MX          ;SAVE LOWER LIMIT
[illegible]  000004  011121              MXTCR:  TYPE    ,M12            ;ASK FOR UPPER LIMIT
[illegible]  004767  007416                      JSR     PC,READS        ;GET IT
[illegible]  005000                              CLR     R0
[illegible]  012701  012316                      MOV     #INPUT,R1
[illegible]  105711                      CR3:    TSTB    (1)             ;CR?
[illegible]  001411                              BEQ     CR4             ;TEST FOR END
[illegible]  106011                              RORB    (1)
[illegible]  106011                              RORB    (1)
[illegible]  106011                              RORB    (1)
[illegible]  106100                              ROLB    R0
[illegible]  106111                              ROLB    (1)
[illegible]  106100                              ROLB    R0
[illegible]  106121                              ROLB    (1)+
[illegible]  106100                              ROLB    R0
[illegible]  [illegible]                         BR      CR3
[illegible]  020067  176164              CR4:    CMP     R0,LIMIT        ;GREATER THAN THE MAX?
[illegible]  [illegible]                         BGT     MXTCR           ;YES - TRY AGAIN
[illegible]  020067  176154                      CMP     R0,MIN          ;R0 MUST BE > MIN
[illegible]  002705                              BLT     CALF            ;IT'S NOT
[illegible]  000167  177272                      JMP     TO              ;GET ANOTHER

[illegible]  000004  011127              NODEV:  TYPE    ,M13            ;TYPES NO DEVICE
[illegible]  000004  011137                      TYPE    ,M13
[illegible]  000004  012316                      TYPE    ,INPUT
[illegible]  000004  002670                      TYPE    ,.+2            ;.ASCIZ " DOES NOT EXIST"<15><12>
002712  000167  176106                          JMP     BEGIN           ;RETURN TO TTY INPUT MODE
```

```
                                        ;       $TRAP           ;TRAP HANDLER

                                        ;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APPROPRIATE
                                        ;SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
                                        ;THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
                                        ;FOLLOW THIS MACRO.

        002716  011605                  TRAPS:  MOV     (6),R5          ;GET TRAP INSTRUCTION
        002720  162705  000002                  SUB     #2,R5           ;BACK UP BY 2
        002724  111505                          MOVB    (5),R5          ;GET THE RIGHT BYTE OF TRAP
        002726  062705  002750                  ADD     #ADRTAB,R5      ;INDEX TO TABLE
        002732  022705  003020                  CMP     #ENDTAB,R5      ;CHECK FOR END OF TABLE
        002736  103401                          BHI     .+4             ;OUT OF BOUNDS
        002740  000175  000000                  JMP     @(5)            ;GO TO ROUTINE
        002744  000000                          HALT                    ;TRAP OUT OF BOUNDS
        002746  000776                          BR      .-2             ;HANG UP

        002750                          ADRTAB:
        [illegible] 002716                      WRADR           ;CORE   =  TRAP+0      (104400)
        [illegible] [illegible]                 WRCADR          ;COMCOR =  TRAP+2      (104402)
        [illegible] [illegible]                 CH.RF           ;CHK.RF =  TRAP+4      (104404)
        [illegible] [illegible]                 CH.RC           ;CHK.RC =  TRAP+6      (104406)
        [illegible] [illegible]                 CH.RP           ;CHK.RP =  TRAP+10     (104410)
        [illegible] [illegible]                 CH.RK           ;CHK.RK =  TRAP+12     (104412)
        [illegible] [illegible]                 CH.TC           ;CHK.TC =  TRAP+14     (104414)
        [illegible] [illegible]                 CH.TM           ;CHK.TM =  TRAP+16     (104416)
        [illegible] 010616                      CH.DR           ;CHK.DR =  TRAP+20     (104420)
        [illegible] [illegible]                 S.TC            ;SEARCH =  TRAP+22     (104422)
        [illegible] 007174                      RW.TM           ;REWIND =  TRAP+24     (104424)
        [illegible] [illegible]                 CDATA           ;CKDATA =  TRAP+26     (104426)
        [illegible] [illegible]                 CH.TC           ;ERR2   =  TRAP+30     (104430)
        [illegible] [illegible]                 POTS            ;STOP   =  TRAP+32     (104432)
        [illegible] 002774                      ER3             ;ERR3   =  TRAP+34     (104434)
        [illegible] [illegible]                 .BEGIN          ;.START =  TRAP+36     (104436)
        003010  [illegible]                     SETBRK          ;SETBRN =  TRAP+40     (104440)
        003012  [illegible]                     NXT             ;NEXT   =  TRAP+42     (104442)
        003014  004160                          MAP             ;MAPIT  =  TRAP+44     (104444)
        003016  [illegible]                     PATCH1          ;DUM1   =  TRAP+46     (104446)
        003020  [illegible]                     PATCH2          ;DUM2   =  TRAP+50     (104450)
                003020                  ENDTAB=.-2


        003022  004767  007006          GETMON: JSR     PC,LODRES       ;RESTORE THE LOADER
        [illegible] 000004  000177              TYPE    ,177            ;TYPE A NULL
        [illegible] 000005                      RESET                   ;KILL THE WORLD
        [illegible] 010046                      MOV     R0,-(6)         ;PUSH R0 ON STACK
        [illegible] 017703  000042              MOV     @#42,R3         ;GET THE ADDRESS
        [illegible] 004713                      JSR     PC,(3)          ;GO TO MONITOR
        [illegible] 000240  000240  000240      240,240,240             ;SAVE ROOM FOR ACT11
        [illegible] 012600                      MOV     (6)+,R0         ;POP STACK INTO R0
        [illegible] 005003                      CLR     R3              ;INIT COUNT
        [illegible] 005203              17$:    INC     R3              ;COUNT
        [illegible] 001376                      BNE     17$             ;DOWN
        [illegible] 000207                      RTS     PC              ;RETURN
```

```
003064  005067  006446           NXT:    CLR     FLG             ;SET A FLAG
003070  032737  040000  177570           BIT     #SW14,@#SWR     ;LOOP ON BANK?
003076  001126                           BNE     2$              ;YES - GET OUT
003100  126700  175710                   CMPB    MX,MO           ;LAST PAGE?
003104  002450                           BLT     1$              ;NO - SKIP
003106  032737  002000  177570           BIT     #SW10,@#SWR     ;INHIBIT BELL
003114  001002                           BNE     5$              ;YES!
003116  000004  000007                   TYPE    ,BELL           ;RING THE BELL
003122  005367  006406           5$:     DEC     PSCNT           ;3 YET
003126  001020                           BNE     12$             ;NO - SKIP
003130  005267  006402                   INC     FLG             ;RECALL MONITER
003134  013703  000042                   MOV     @#42,R3         ;GET MONITOR ADDRESS
003140  001410                           BEQ     10$             ;SKIP IF 0
003142  062767  000002  006362           ADD     #2,DEVADR       ;GO TO NEXT ONE
003150  005777  006356                   TST     @DEVADR         ;END OF TABLE?
003154  001002                           BNE     10$             ;SKIP MONITOR CALL
003156  004767  177640                   JSR     PC,GETMON       ;GET THE MONITOR
003162  012767  000003  006344   10$:    MOV     #3,PSCNT        ;RESET COUNT
003170  016767  175622  175616   12$:    MOV     MIN,MX          ;RESET MX
003176  005767  000512                   TST     MEMORY          ;CHECK STATE OF FLAG
[illegible]  100407                      BMI     4$              ;SWAB IT
[illegible]  001403                      BEQ     3$              ;COM IT
[illegible]  005067  000502              CLR     MEMORY          ;INIT IT
[illegible]  000405                      BR      1$              ;CONTINUE
[illegible]  105167  000475      3$:     COMB    MEMORY+1        ;MAKE IT 177400
[illegible]  000402                      BR      1$              ;SKIP
[illegible]  000367  000466      4$:     SWAB    MEMORY          ;MAKE IT 377
[illegible]  026767  [illegible]  175560   1$:  CMP  MIN,MX      ;FIRST?
[illegible]  001022                      BNE     6$              ;SKIP CALF
[illegible]  005737  000042              TST     @#42            ;MONITOR?
[illegible]  001407                      BEQ     14$             ;SKIP LOOPING
[illegible]  005767  006266              TST     FLG             ;UNDER MONITOR?
[illegible]  001404                      BEQ     14$             ;SKIP EXIT
[illegible]  005037  177776              CLR     @#PS            ;CLEAR PS
[illegible]  000167  [illegible]         JMP     MEMDEV          ;LOOP
[illegible]  032737  000400  177570  14$: BIT    #SW8,@#SWR      ;TRACE?
003270  001404                           BEQ     6$              ;NO - SKIP
003272  000004  003276                   TYPE    ,.+2            ;.ASCIZ <15><12>
[illegible]  005267  [illegible]     6$:  INC     MX              ;GO TO NEXT BANK
[illegible]  016737  [illegible]  177570  MOV     MX,@#DISPLAY    ;DISPLAY THE BANK IN USE
[illegible]  032737  000400  177570      BIT     #SW8,@#SWR      ;TYPE BANK NUMBER?!
[illegible]  001414                      BEQ     2$              ;NO!
[illegible]  000004  003330              TYPE    ,.+2            ;.ASCIZ " "
[illegible]  016705  [illegible]         MOV     MX,TTY          ;SET FOR TYPING
[illegible]  032705  000007              BIT     #7,TTY          ;MULTIPLE OF 10?
[illegible]  001402                      BEQ     7$              ;YES
[illegible]  042705  177770              BIC     #177770,TTY     ;CLEAR OUT UPPER BITS
[illegible]  004767  007022      7$:     JSR     PC,PRINTS       ;TYPE MX AND SUPPRESS LEADING ZEROES
[illegible]  104440              2$:     SETBANK                 ;SET MXC
[illegible]  016704  006366              MOV     LOLIM,R4        ;SET UP FOR BACKGROUND
003362  000002                           RTI                     ;RETURN
```

```
                                CDATA:  SETBNK                  ;CHECK DATA
                                        TST     EX
                                        BEQ     .+4
                                        RTI
                                        CLR     DATA
                                        BTT
                                        MOV     LOLIM,R2        ;SET UP BEGINNING OF BUFFER
                                1$:     JSR     PC,GDATA
                                        CMP     DATA,(2)
                                        BNE     .+4
                                        EMT3
                                        TST     (2)+
                                        CMP     HILIM,R2        ;CHECK FOR END
                                        BNE     1$
                                        RTI

                                GDATA:  TST     MEMORY          ;CHECK FLAG
                                        BMI     T1XOR8          ;MM11G
                                        BCT     T8XOR13         ;MM11F

                                T1XOR8: MOV     R2,R3           ;MM11E MEMORY
                                        SUB
                                        ROR
                                        ROR
                                        R2,R3
                                        ROR
                                ALLOR:  BIT     T,R3
                                        BNE
                                        BIT
                                        DATA
                                1$:     RTS     PC

                                T8XOR13:MOV     R2,R3           ;MM11F MEMORY
                                        MOV     ,R5
                                        SUB
                                        ROR
                                        ROR
                                        ROR
                                        ROR
                                T3X9:   ROR     R5,R3
                                        BR      ALLOR

                                T3XOR9: MOV     R2,R3           ;MM11G MEMORY
                                        ROR
                                        MOV     ,R5
                                        SUB
                                        ROR
                                        ROR
                                        BR      T3X9
```

```
003544	005037	177776		POTS:	CLR	@#177776		;DROP PRIORITY
003550	005737	177570			TST	@#SWR			;HALT ON ERROR?
003554	100001				BPL	.+4
003556	000000				HALT
003560	000002				RTI

003562	016705	175226		.BEGIN:	MOV	MX,TTY
003566	000241				CLC				; #ASH #13,TTY#
003570	006005				ROR	TTY			;TO GET
003572	006005				ROR	TTY			;THE THREE
003574	006005				ROR	TTY			;UPPER BITS
003576	006005				ROR	TTY			;TO MAKE
003600	042705	017777			BIC	#17777,TTY		;THE ADDRESS
003604	017646	000000			MOV	@(6),-(6)		;GET ADDRESS
003610	017646	000000			MOV	@(6),-(6)		;GET ADDRESS
003614	010576	000000			MOV	TTY,@(6)		;LOAD MEMORY ADDRESS
003620	022626				CMP	(6)+,(6)+		;RESTORE STACK
003622	062716	000002			ADD	#2,(6)			;INCREMENT STACK
003626	016705	175162			MOV	MX,TTY			;GET UPPER 2 BITS OF MXC
003632	006105				ROL	TTY			;GET INTO POSITION
003634	042705	177717			BIC	#177717,TTY		;CLEAR JUNK
003640	022767	041524	006450		CMP	#"TC,INPUT		;CHECK FOR DTA
003646	001002				BNE	1$			;SKIP IF NOT
003650	052705	000400			BIS	#400,TTY		;SET DTA1
003654	057605	000000		1$:	BIS	@(6),TTY		;GET REST OF COMMAND
003660	062716	000002			ADD	#2,(6)			;INCREMENT STACK
003664	017646	000000			MOV	@(6),-(6)		;GET ADDRESS
003670	017646	000000			MOV	@(6),-(6)		;GET ADDRESS
003674	010576	000000			MOV	TTY,@(6)		;LOAD MEMORY ADDRESS
003700	022626				CMP	(6)+,(6)+		;RESTORE STACK
003702	062716	000002			ADD	#2,(6)			;INCREMENT STACK
003706	000002				RTI
003710	000000			DATA:	0
003712	000000			BIT:	0
003714	000000			MEMORY:	0

003716	104442			WRROR:	NEXT
003720	005067	177764			CLR	DATA
003724	005067	177762			CLR	BIT
003730	016702	006004			MOV	LOLIM,R2		;SET THE STARTING ADDRESS
003734	004767	177600		1$:	JSR	PC,GDATA
003740	016722	177744			MOV	DATA,(2)+
003744	026702	006772			CMP	HILIM,R2		;END?
003750	001371				BNE	1$
003752	000002				RTI

003754	[illegible]			WRCADR:	SETBANK
003756	[illegible]	006756			MOV	LOLIM,R2		;SET UP BEGINNING OF BUFFER
003762	[illegible]		1$:	COM	(2)+			;COMPLIMENT IT
003764	[illegible]	006752			CMP	HILIM,R2
003770	001374				BNE	1$			;LOOP TIL END OF CORE
003772	000002				RTI
```

```
004 003774 032737 020000 177570 ER3:    BIT     @SW13,@#SWR     ;INHIBIT TYPEOUTS
005 004002 001401                       BEQ     .+4
006 004004 000002                       RTI
007 004006 022767 041524 006202         CMP     #"TC,INPUT
008 004014 001003                       BNE     1$
009 004016 012777 000001 006622         MOV     #1,@TCCS
010 004024 000004 011032                TYPE    M7
011 004030 010205               1$:     MOV     R2,TTY          ;TYPE R2 WITH MX AS 18 BIT ADDRESS
012 004032 004767 006264                JSR     PC,PRINTA       ;GO TO ADDRESS PRINTER
013 004036 000004 011053                TYPE    M8
014 004042 016705 177442                MOV     DATA,TTY        ;TYPE DATA IN OCTAL
015 004046 004767 006314                JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
016 004052 000004 011065                TYPE    M9
017 004056 011205                       MOV     (2),TTY         ;TYPE (2) IN OCTAL
018 004060 004767 006302                JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
019 004064 104402                       STOP
020 004066 000002                       RTI
021
022 004070 016705 174730        SETBAK: MOV     MX,TTY
023 004074 005767 174722                TST     OPTION          ;WHAT AM I
024 004100 001411                       BEQ     2$              ;I'M NOTHING
025 004102 100406                       BMI     1$              ;I'M AN MX11
026 004104 000306                       SWAB    TTY             ; #ASH #7,TTY#
027 004106 006305                       ASR     TTY
030 004110 010577 005612                MOV     TTY,@#KISAR6
031 004114 000002                       RTI
032 004116 010577 005614        1$:     MOV     TTY,@#DCC3      ;LOAD THE MCC
033 004122 000002                       RTI
034 004124 000241               2$:     CLC                     ; #ASH #13,TTY#
035 004126 006005                       ROR     TTY             ;ROTATE
036 004130 006005                       ROR     TTY             ;THE UPPER
037 004132 006005                       ROR     TTY             ;THREE BITS
038 004134 006005                       ROR     TTY             ;TO MAKE
039 004136 042705 017777                BIC     #17777,TTY      ;CLEAR JUNK
040 004142 010577 005572                MOV     TTY,LOLIM       ;ADDRESS
041 004146 062705 020000                ADD     #20000,TTY      ;MAKE UPPER
042 004152 010577 005564                MOV     TTY,HILIM       ;LIMIT
043 004156 000002                       RTI
044
045
046 004160 012777 000000 005624 MAP:    MOV     #0,@#KISAR0
047 004166 012777 077406 005620         MOV     #77406,@#KISDR0
050 004174 012777 000200 005614         MOV     #200,@#KISAR6
051 004202 012777 077406 005620         MOV     #77406,@#KISDR6
052 004210 012777 007600 005514         MOV     #7600,@#KISAR7
053 004216 012777 077406 005510         MOV     #77406,@#KISDR7
054 004224 000002                       RTI
```

```
                                M.RF:   CHK.RF                  ;CHECK FOR ERRORS
                                        CKDATA                  ;CHECK CORE
                                        CORE                    ;LOAD CORE WITH DATA
                                TRP.RF: MOV     #C.RF,@#ETRAP   ;'CHECK' TRAP ADDRESS
                                        MOV     #BUFCNT,@#RFWC  ;WORD COUNT
                                        CLR     @#RFDA          ;DSK ADDRESS
                                        .START                  ;LOAD CURRENT ADDRESS & IE!3 AND GO
                                        #RFCA,100!3,#RFCS
                                        RTI

                                C.RF:   CHK.RF                  ;CHECK FOR ERRORS
                                        MOV     #R.RF,@#ETRAP   ;'READ' TRAP ADDRESS
                                        MOV     #BUFCNT,@#RFWC  ;WORD COUNT
                                        CLR     @#RFDA          ;DSK ADDRESS
                                        .START                  ;LOAD CURRENT ADDRESS & IE!7 AND GO
                                        #RFCA,100!7,#RFCS
                                        RTI

                                R.RF:   CHK.RF                  ;CHECK FOR ERRORS
                                        COMCOR                  ;COMPLIMENT CORE
                                        MOV     #W.RF,@#ETRAP   ;'WRITE' TRAP ADDRESS
                                        MOV     #BUFCNT,@#RFWC  ;WORD COUNT
                                        CLR     @#RFDA          ;DSK ADDRESS
                                        .START                  ;LOAD CURRENT ADDRESS & IE!5 AND GO
                                        #RFCA,100!5,#RFCS
                                        RTI
```

```
004360  005067  004330          CH.RF:  CLR     EX              ;CLEAR ERROR FLAG
004364  005777  005214                  TST     @RFCS           ;ANY ERRORS?
004370  100026                          BPL     1$              ;BRANCH IF NO ERRORS
004372  005267  004316                  INC     EX              ;SET ERROR FLAG
004376  032737  020000  177570          BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
004404  001020                          BNE     1$              ;INHIBIT TYPEOUTS
004406  017705  005172                  MOV     @RFCS,TTY       ;SET TTY FOR TYPING
004412  000004  011151                  TYPE    ,RF.M1          ;TYPE DEVICE MESSAGE
004416  000004  011012                  TYPE    ,M3             ;TYPE CS
004422  004767  005740                  JSR     PC,PRINTR       ;TYPE RFCS IN OCTAL
004426  000004  011022                  TYPE    ,M4             ;TYPE ER
004432  017705  005150                  MOV     @RFER,TTY       ;TYPE @RFER IN OCTAL
004436  004767  005724                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
004442  104432                          STOP                    ;HALT ON ERROR
004444  000002                          RTI                     ;RETURN
004446  032737  020000  177570  1$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
004454  001054                          BNE     2$              ;YES!
004456  117767  005122  004226          MOVB    @RFCS,CHK       ;GET EA BITS
004464  116767  174324  004221          MOVB    MX,CHK+1        ;GET MX BITS
004472  105267  004215                  INCB    CHK+1           ;INCREMENT INTO EA BITS
004476  106067  004210                  RORB    CHK             ;MOVE OVER BY 1
004502  006067  004204                  ROR     CHK             ;MOVE IT
004506  006067  004200                  ROR     CHK             ;INTO
004512  006067  004174                  ROR     CHK             ;POSITION
004516  042767  176374  004166          BIC     #176374,CHK     ;CLEAR JUNK
004524  126767  004162  004161          CMPB    CHK,CHK+1       ;MAKE SURE EA BITS INCREMENT
004532  001425                          BEQ     2$              ;RETURN IF EQUAL
004534  000004  011151                  TYPE    ,RF.M1          ;TYPE DEVICE MESSAGE
004540  000004  011142                  TYPE    ,M15            ;TYPE "BANK"
004544  016705  174244                  MOV     MX,TTY          ;TYPE MX IN OCTAL
004550  004767  005622                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
004554  000004  011063                  TYPE    ,M8             ;TRUE
004560  116705  004127                  MOVB    CHK+1,TTY       ;GET BYTE
004564  004767  005606                  JSR     7,PRINTS        ;TYPE TRUE EA BITS
004570  000004  011066                  TYPE    ,M9             ;RECEIVED
004574  116705  004112                  MOVB    CHK,TTY         ;GET BYTE
004600  004767  005672                  JSR     7,PRINTS        ;TYPE RECEIVED EA BITS
004604  104432                          STOP                    ;WAIT
004606  000002                  2$:     RTI                     ;RETURN
```

```
 916  004610  104406                  M.RC:   CHK.RC                  ;CHECK FOR ERRORS
 917  004612  104455                          CKDATA                  ;CHECK CORE
 918  004614  104400                  TRP.RC: CORE                    ;LOAD CORE WITH DATA
 919  004616  012777  004650  004720          MOV     #C.RC,@RCTRAP   ;'CHECK' TRAP ADDRESS
 920  004624  016777  005114  004770          MOV     #NBDCNT,@RCWC   ;WORD COUNT
 921  004632  005077  004770                  CLR     @RCDA           ;DSK ADDRESS
 922  004636  104436                          .START                  ;LOAD CURRENT ADDRESS & IE!3 AND GO
 923  004640  011677  000103  011616          RCCA,100!3,RCCS
 924  004646  000002                          RTI
 925
 926  004650  104406                  C.RC:   CHK.RC                  ;CHECK FOR ERRORS
 927  004652  012777  004704  004664          MOV     #R.RC,@RCTRAP   ;'READ' TRAP ADDRESS
 928  004660  016777  005050  004734          MOV     #NBDCNT,@RCWC   ;WORD COUNT
 929  004666  005077  004734                  CLR     @RCDA           ;DSK ADDRESS
 930  004672  104436                          .START                  ;LOAD CURRENT ADDRESS & IE!7 AND GO
 931  004674  011677  000107  011616          RCCA,100!7,RCCS
 932  004702  000002                          RTI
 933
 934  004704  104406                  R.RC:   CHK.RC                  ;CHECK FOR ERRORS
 935  004706  104402                          COMCOR                  ;COMPLIMENT CORE
 936  004710  012777  004610  004626          MOV     #W.RC,@RCTRAP   ;'WRITE' TRAP ADDRESS
 937  004716  016777  005022  004676          MOV     #NBDCNT,@RCWC   ;WORD COUNT
 938  004724  005077  004676                  CLR     @RCDA           ;DSK ADDRESS
 939  004730  104436                          .START                  ;LOAD CURRENT ADDRESS & IE!5 AND GO
 940  004732  011677  000105  011616          RCCA,100!5,RCCS
 941  004740  000002                          RTI
```

L02


```
004742  005067  003746            CH.RC:  CLR     EX              ;CLEAR ERROR FLAG
004746  005777  004644                    TST     @RCCS           ;ANY ERRORS?
004752  100026                            BPL     1$              ;BRANCH IF NO ERRORS
004754  005267  003734                    INC     EX              ;SET ERROR FLAG
004760  032737  020000  177570            BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
004766  001020                            BNE     1$              ;INHIBIT TYPEOUTS
004770  017705  004622                    MOV     @RCCS,TTY       ;SET TTY FOR TYPING
004774  000004  011174                    TYPE    ,RC.M1          ;TYPE DEVICE MESSAGE
005000  000004  011012                    TYPE    ,M3             ;TYPE CS
005004  004767  005356                    JSR     PC,PRINTR       ;TYPE RCCS IN OCTAL
005010  000004  011022                    TYPE    ,M4             ;TYPE ER
005014  017705  004600                    MOV     @RCER,TTY       ;TYPE @RCER IN OCTAL
005020  004767  005342                    JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
005024  104432                            STOP                    ;HALT ON ERROR
005026  000002                            RTI                     ;RETURN
005030  032737  020000  177570    1$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
005036  001054                            BNE     2$              ;YES!
005040  117767  004552  003644            MOVB    @RCCS,CHK       ;GET EA BITS
005046  116767  173742  003637            MOVB    MX,CHK+1        ;GET MX BITS
005054  105267  003633                    INCB    CHK+1           ;INCREMENT INTO EA BITS
005060  106067  003626                    RORB    CHK             ;MOVE OVER BY 1
005064  006067  003622                    ROR     CHK             ;MOVE IT
005070  006067  003616                    ROR     CHK             ;INTO
005074  006067  003612                    ROR     CHK             ;POSITION
005100  042767  176374  003604            BIC     #176374,CHK     ;CLEAR JUNK
005106  126767  003600  003577            CMPB    CHK,CHK+1       ;MAKE SURE EA BITS INCREMENT
005114  001425                            BEQ     2$              ;RETURN IF EQUAL
005116  000004  011174                    TYPE    ,RC.M1          ;TYPE DEVICE MESSAGE
005122  000004  011142                    TYPE    ,M15            ;TYPE "BANK"
005126  016705  173642                    MOV     MX,TTY          ;TYPE MX IN OCTAL
005132  004767  005240                    JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
005136  000004  011053                    TYPE    ,M8             ;TRUE
005142  116705  003545                    MOVB    CHK+1,TTY       ;GET BYTE
005146  004767  005224                    JSR     7,PRINTS        ;TYPE TRUE EA BITS
005152  000004  011065                    TYPE    ,M9             ;RECEIVED
005156  116705  003530                    MOVB    CHK,TTY         ;GET BYTE
005162  004767  005210                    JSR     7,PRINTS        ;TYPE RECEIVED EA BITS
005166  104432                            STOP                    ;WAIT
005170  000002                    2$:     RTI                     ;RETURN
```

```
                                M.RP:   CHK.RP                  ;CHECK FOR ERRORS
                                        CKDATA                  ;CHECK CORE
                                TP.RP:  CORE                    ;LOAD CORE WITH DATA
005172  104410                          MOV     #C.RP,@RPTRAP   ;'CHECK' TRAP ADDRESS
005174  012777  000232  004342          MOV     WRDCNT,@RPWC    ;WORD COUNT
005200  016777  000532  004316          CLR     @RPDA           ;DSK ADDRESS
005204  005077  004316                  .START                  ;LOAD CURRENT ADDRESS & IE!3 AND GO
005210  104436                          RPCA,100!3,RPCS
005212  011634  000103  011630          RTI
005216  000002
005220
                                C.RP:   CHK.RP                  ;CHECK FOR ERRORS
005222  104410                          MOV     #R.RP,@RPTRAP   ;'READ' TRAP ADDRESS
005224  012777  000256  004306          MOV     WRDCNT,@RPWC    ;WORD COUNT
005230  016777  004476  004362          CLR     @RPDA           ;DSK ADDRESS
005234  005077  004362                  .START                  ;LOAD CURRENT ADDRESS & IE!7 AND GO
005240  104436                          RPCA,100!7,RPCS
005242  011634  000107  011630          RTI
005246  000002

                                R.RP:   CHK.RP                  ;CHECK FOR ERRORS
005250  104410                          COMCOR                  ;COMPLIMENT CORE
005252  104402                          MOV     #W.RP,@RPTRAP   ;'WRITE' TRAP ADDRESS
005254  012777  005172  004260          MOV     WRDCNT,@RPWC    ;WORD COUNT
005260  016777  004440  004324          CLR     @RPDA           ;DSK ADDRESS
005264  005077  004324                  .START                  ;LOAD CURRENT ADDRESS & IE!5 AND GO
005270  104436                          RPCA,100!5,RPCS
005272  011634  000105  011630          RTI
005276  000002
005300  012777  000011  004276  TRP.RP: MOV     #11,@RPCS
005304  105777  004272                  TSTB    @RPCS
005310  100375                          BPL     .-4
005312  005777  004276                  TST     @RPDS
005316  100375                          BPL     .-4
005320  005077  004256                  CLR     @RPCS
005324  000711                          BR      TP.RP
```

```
1014  005354  005067  003334          CH.RP:  CLR     EX              ;CLEAR ERROR FLAG
1015  005360  005777  004244                  TST     @RPCS           ;ANY ERRORS?
1016  005364  100034                          BPL     1$              ;BRANCH IF NO ERRORS
1017  005366  005267  003322                  INC     EX              ;SET ERROR FLAG
1018  005372  032737  020000  177570          BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
1019  005400  001026                          BNE     1$              ;INHIBIT TYPEOUTS
1020  005402  017705  004222                  MOV     @RPCS,TTY       ;SET TTY FOR TYPING
1021  005406  000004  011242                  TYPE    ,RP.M1          ;TYPE DEVICE MESSAGE
1022  005412  000004  011012                  TYPE    ,M3             ;TYPE CS
1023  005416  004767  004744                  JSR     PC,PRINTR       ;TYPE RPCS IN OCTAL
1024  005422  000004  011032                  TYPE    ,M4             ;TYPE ER
1025  005426  017705  004206                  MOV     @RPER,TTY       ;TYPE @RPER IN OCTAL
1026  005432  004767  004730                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1027  005436  104432                          STOP                    ;HALT ON ERROR
1028  005440  012777  000001  004162          MOV     #1,@RPCS        ;LOAD A CLEAR
1029  005446  105777  004156                  TSTB    @RPCS           ;WAIT FOR
1030  005452  100375                          BPL     .-4             ;DONE
1031  005454  000002                          RTI                     ;RETURN
1032  005456  032737  020000  177570  1$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
1033  005464  001054                          BNE     2$              ;YES!
1034  005466  117767  004136  002216          MOVB    @RPCS,CHK       ;GET EA BITS
1035  005474  116767  172314  003211          MOVB    MX,CHK+1        ;GET MX BITS
1036  005502  105267  003205                  INCB    CHK+1           ;INCREMENT INTO EA BITS
1037  005506  105267  003200                  INCB    CHK             ;MOVE OVER BY 1
1038  005512  006067  003174                  ROR     CHK             ;MOVE IT
1039  005516  006067  003170                  ROR     CHK             ;INTO
1040  005522  006067  003164                  ROR     CHK             ;POSITION
1041  005526  042767  176374  003156          BIC     #176374,CHK     ;CLEAR JUNK
1042  005534  126767  003152  003151          CMPB    CHK,CHK+1       ;MAKE SURE EA BITS INCREMENT
1043  005542  001465                          BEQ     2$              ;RETURN IF EQUAL
1044  005544  000004  011242                  TYPE    ,RP.M1          ;TYPE DEVICE MESSAGE
1045  005550  000004  011142                  TYPE    ,M15            ;TYPE "BANK"
1046  005554  016705  172334                  MOV     MX,TTY          ;TYPE MX IN OCTAL
1047  005560  004767  004512                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
1048  005564  000004  011053                  TYPE    ,M8             ;TRUE
1049  005570  116705  003117                  MOVB    CHK+1,TTY       ;GET BYTE
1050  005574  004767  004576                  JSR     7,PRINTS        ;TYPE TRUE EA BITS
1051  005600  000004  011066                  TYPE    ,M9             ;RECEIVED
1052  005604  116705  003102                  MOVB    CHK,TTY         ;GET BYTE
1053  005610  004767  004562                  JSR     7,PRINTS        ;TYPE RECEIVED EA BITS
1054  005614  104432                          STOP                    ;WAIT
1055  005616  000002                  2$:     RTI                     ;RETURN
```

```
                                M.RK:   CHK.RK                  ;CHECK FOR ERRORS
                                        CKDATA                  ;CHECK CORE
                                TRP.RK: CORE                    ;LOAD CORE WITH DATA
005640  012777  005640  002720            MOV     @C.RK,@RKTRAP   ;'CHECK' TRAP ADDRESS
005644  016777  004104  004006            MOV     WRDCNT,@RKDC    ;WORD COUNT
005652  005077  004006                    CLR     @RKDA           ;DISK ADDRESS
005656  104436                            .START                  ;LOAD CURRENT ADDRESS & IE!3 AND GO
005660  011677  000103  011644            MOVB    R0,100!3,RKCS
005666  000002                            RTI

                                C.RK:   CHK.RK                  ;CHECK FOR ERRORS
005670  012777  005714  002664            MOV     @R.RK,@RKTRAP   ;'READ' TRAP ADDRESS
005674  016777  004060  002752            MOV     WRDCNT,@RKDC    ;WORD COUNT
005702  005077  002752                    CLR     @RKDA           ;DISK ADDRESS
005706  104436                            .START                  ;LOAD CURRENT ADDRESS & IE!7 AND GO
005710  011677  000107  011644            MOVB    R0,100!7,RKCS
005712  000002                            RTI

                                R.RK:   CHK.RK                  ;CHECK FOR ERRORS
005714  104412                            COMCOR                  ;COMPLIMENT CORE
005716  104406                            MOV     @W.RK,@RKTRAP   ;'WRITE' TRAP ADDRESS
005720  012777  005620  002636            MOV     WRDCNT,@RKDC    ;WORD COUNT
005726  016777  004012  002714            CLR     @RKDA           ;DISK ADDRESS
005734  005077  002714                    .START                  ;LOAD CURRENT ADDRESS & IE!5 AND GO
005740  104436                            MOVB    R0,100!5,RKCS
005742  011677  000105  011644            RTI
005750  000002
```

```
                                CH.RK:  CLR     EX              ;CLEAR ERROR FLAG
                                        TST     @RKCS           ;ANY ERRORS?
                                        BPL     1$              ;BRANCH IF NO ERRORS
                                        INC     EX              ;SET ERROR FLAG
                                        BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
                                        BNE     1$              ;INHIBIT TYPEOUTS
                                        MOV     @RKCS,TTY       ;SET TTY FOR TYPING
                                        TYPE    ,RK.MI          ;TYPE DEVICE MESSAGE
                                        TYPE    ,M3             ;TYPE CS
                                        JSR     PC,PRINTR       ;TYPE RKCS IN OCTAL
                                        TYPE    ,M4             ;TYPE ER
                                        MOV     @RKER,TTY       ;TYPE @RKER IN OCTAL
                                        JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
                                        STOP                    ;HALT ON ERROR
                                        MOV     #1,@RKCS        ;LOAD A CLEAR
                                        TSTB    @RKCS           ;WAIT FOR
                                        BPL     .-4             ;DONE
                                        RTI                     ;RETURN
                                1$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
                                        BNE     2$              ;YES.
                                        MOVB    @RKCS,CHK       ;GET EA BITS
                                        MOVB    RK,CHK+1        ;GET RK BITS
                                        INCB    CHK+1           ;INCREMENT INTO EA BITS
                                        ROLB    CHK             ;MOVE OVER BY 1
                                        ROLB    CHK             ;MOVE IT
                                        ROLB    CHK             ;INTO
                                        ROLB    CHK             ;POSITION
                                        BIC     #176374,CHK     ;CLEAR JUNK
                                        CMPB    CHK,CHK+1       ;MAKE SURE EA BITS INCREMENT
                                        BEQ     2$              ;RETURN IF EQUAL
                                        TYPE    ,RK.MI          ;TYPE DEVICE MESSAGE
                                        TYPE    ,M15            ;TYPE 'BANK'
                                        MOV     RK,TTY          ;TYPE RK IN OCTAL
                                        JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                        TYPE    ,M8             ;TRUE
                                        MOVB    CHK+1,TTY       ;GET BYTE
                                        JSR     7,PRINTS        ;TYPE TRUE EA BITS
                                        TYPE    ,M9             ;RECEIVED
                                        MOVB    CHK,TTY         ;GET BYTE
                                        JSR     7,PRINTS        ;TYPE RECEIVED EA BITS
                                        STOP                    ;WAIT
                                2$:     RTI                     ;RETURN
```

```
                                M.TC:   CHK.TC                  ;CHECK FOR ERRORS
                                        CKDATA                  ;CHECK CORE
                                        MOV     #1,@TCCS        ;STOP ALL TRANSPORTS
                                TRP.TC: CORE                    ;LOAD WORST CASE NOISE
                                        SEARCH  .10             ;SEARCH FOR BLOCK 10
                                        MOV     #R.TC,@TCTRAP   ;TRAP ADDRESS
                                        MOV     #WDCNT,@TCWC    ;WORD COUNT
                                        .START                  ;LOAD CURRENT ADDRESS & IE!15 AND GO
                                        TCCA,100!15,TCCS
                                        RTI

                                R.TC:   CHK.TC                  ;CHECK FOR ERROR
                                        CONCOR                  ;COMPLIMENT CORE
                                        SEARCH  .10             ;SEARCH FOR BLOCK 10
                                        MOV     #W.TC,@TCTRAP   ;TRAP ADDRESS
                                        MOV     #WDCNT,@TCWC    ;WORD COUNT
                                        .START                  ;LOAD CURRENT ADDRESS & IE!5 AND GO
                                        TCCA,100!5,TCCS
                                        RTI
```

```
                                    CH.TC:  CLR     EX              ;CLEAR ERROR FLAG
                                            TST     @TCCS           ;ANY ERRORS?
                                            BPL     1$              ;BRANCH IF NO ERRORS
                                            INC     EX              ;SET ERROR FLAG
                                            BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
                                            BNE     1$              ;INHIBIT TYPEOUTS
                                            MOV     @TCCS,TTY       ;SET TTY FOR TYPING
                                            TYPE    ,TC.M1          ;TYPE DEVICE MESSAGE
                                            TYPE    ,M3             ;TYPE CS
                                            JSR     PC,PRINTR       ;TYPE TCCS IN OCTAL
                                            TYPE    ,M4             ;TYPE ER
                                            MOV     @TCER,TTY       ;TYPE @TCER IN OCTAL
                                            JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
                                            STOP                    ;HALT ON ERROR
                                            MOV     #1,@TCCS        ;LOAD A CLEAR
                                            TSTB    @TCCS           ;WAIT FOR
                                            BPL     .-4             ;DONE
                                    1$:     RTI                     ;RETURN
                                            BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
                                            BNE     2$              ;YES!
                                            MOVB    @TCCS,CHK       ;GET EA BITS
                                            MOVB    MX,CHK+1        ;GET MX BITS
                                            INCB    CHK+1           ;INCREMENT INTO EA BITS
                                            ROLB    CHK             ;MOVE OVER BY 1
                                            ROLB    CHK             ;MOVE IT
                                            ROLB    CHK             ;INTO
                                            ROLB    CHK             ;POSITION
                                            BIC     #177674,CHK     ;CLEAR JUNK
                                            CMPB    CHK,CHK+1       ;MAKE SURE EA BITS INCREMENT
                                            BEQ     2$              ;RETURN IF EQUAL
                                            TYPE    ,TC.M1          ;TYPE DEVICE MESSAGE
                                            TYPE    ,M15            ;TYPE  BANK
                                            MOV     MX,TTY          ;TYPE MX IN OCTAL
                                            JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
                                            TYPE    ,M8             ;TRUE
                                            MOVB    CHK+1,TTY       ;GET BYTE
                                            JSR     7,PRINTS        ;TYPE TRUE EA BITS
                                            TYPE    ,M9             ;RECEIVED
                                            MOVB    CHK,TTY         ;GET BYTE
                                            JSR     7,PRINTS        ;TYPE RECEIVED EA BITS
                                            STOP                    ;WAIT
                                    2$:     RTI                     ;RETURN
```

```
1185  006566  012667  000000  000330  S.TC:   MOV     @(6),TC.TA
1186  006572  062716  000002                  ADD     #2,(6)
1187  006576  005767  000320                  TST     TC.TA           ;FOWARD OR REVERSE
1188  006602  100004                          BPL     FOWSER          ;LEAVE IT IN REVERSE
1189  006604  005467  000312                  NEG     TC.TA           ;GET PLUS BLOCK NUMBER
1190  006610  012777  006662  002740          MOV     #SR1,@TCTRAP    ;TRAP ADDRESS
1191  006616  016767  000300  000274          MOV     TC.TA,WANT
1192  006624  062767  000002  000266          ADD     #2,WANT         ;2 PAST BLOCK
1193  006632  012667  000270                  MOV     (6)+,RETURN     ;SAVE RETURN ADDRESS
1194  006636  005726                          TST     (6)+
1195  006640  012777  000503  003006  RETI:   MOV     #503,@TCCS      ;SEARCH FOWARD
1196  006646  000002                          RTI
1197
1198  006650  005777  003000          SR1:    TST     @TCCS           ;CHECK FOR ERROR
1199  006654  100006                          BPL     SR1A            ;NO ERRORS
1200  006656  005777  002774                  TST     @TCER           ;CHECK FOR END OF TAPE
1201  006662  100401                          BMI     .+4
1202  006664  104434                          EMT     34
1203  006666  000405                          BR      SF1
1204
1205  006670  026777  000224  002766  SR1A:   CMP     WANT,@TCDB      ;ON BLOCK?
1206  006676  003401                          BLT     SF1             ;GO FOWARD
1207  006700  000757                          BR      RETI            ;CONTINUE AS YOU WERE
1208
1209  006702  012777  006726  002646  SF1:    MOV     #SF2,@TCTRAP    ;TRAP ADDRESS
1210  006710  012777  004503  002736  RETR:   MOV     #4503,@TCCS     ;SEARCH REVERSE
1211  006716  000002                          RTI
1212
1213  006720  005777  002730          SF2:    TST     @TCCS           ;ERROR?
1214  006724  100006                          BPL     SF1A            ;NO ERRORS
1215  006726  005777  002724                  TST     @TCER           ;END OF TAPE?
1216  006732  100401                          BMI     .+4
1217  006734  104434                          EMT     34              ;NO
1218  006736  000740                          BR      RETI            ;SWITCH DIRECTIONS
1219
1220  006740  026777  000156  002716  SF1A:   CMP     TC.TA,@TCDB     ;RIGHT BLOCK?
1221  006746  001360                          BNE     RETR            ;NO
1222  006750  000177  000152                  JMP     @RETURN
```

```
                                FOWSER: MOV     #SC1,@TCTRAP    ;TRAP ADDRESS
                                        MOV     TC.TA,WANT
                                        SUB     #2,WANT         ;2 PAST BLOCK
                                        MOV     (6)+,RETURN     ;SAVE RETURN ADDRESS
                                        TST     (6)+
007006 012777 004503 002642     RETI1:  MOV     #4503,@TCCS     ;SEARCH REVERSE
007014 000002                           RTI

007016 005777 002634            SC1:    TST     @TCCS           ;CHECK FOR ERROR
007022 100005                           BPL     SC1A            ;NO ERRORS
007024 005777 002630                    TST     @TCER           ;CHECK FOR END OF TAPE
007030 100401                           BMI     .+4
007032 104434                           EMT3
007034 000405                           BR      SCF1

007036 026777 000060 002622     SC1A:   CMP     WANT,@TCDB      ;ON BLOCK?
007044 001001                           BNE     SCF1            ;GO REVERSE
007046 000757                           BR      RETI1           ;CONTINUE AS YOU WERE

007050 012777 007066 002602     SCF1:   MOV     #SCF2,@TCTRAP   ;TRAP ADDRESS
007056 012777 000003 002572     RETRI:  MOV     #3,@TCCS        ;SEARCH FOWARD
007064 000002                           RTI

007066 005777 002564            SCF2:   TST     @TCCS           ;ERROR?
007072 100005                           BPL     SCF1A           ;NO ERRORS
007074 005777 002560                    TST     @TCER           ;END OF TAPE?
007100 100401                           BMI     .+4
007102 104430                           EMT2                    ;NO
007104 000740                           BR      RETI1           ;SWITCH DIRECTIONS

007106 026777 000012 002552     SCF1A:  CMP     TC.TA,@TCDB     ;RIGHT BLOCK?
007114 001360                           BNE     RETRI           ;NO
007116 000177 000006                    JMP     @RETURN

007122 000000                   WANT:   0
007124 000000                   TC.TA:  0
007126 000000                   TC.MC:  0
007130 000000                   RETURN: 0
```

```
                                M.TM:   CHK.TM                  ;CHECK FOR ERROR
                                        CKOBTA
                                TRP.TM: CORE                    ;SET UP CORE
                                        REWIND
007124 012777 007224 002420             MOV     #R.TM,@TMTRAP   ;TRAP ADDRESS
007130 016777 002570 002516             MOV     WRDCNT,@TMBC    ;BYTE COUNT
007136 006377 002512                    ASL     @TMBC           ;MAKE BYTE COUNT
007142 104436                           .START                  ;LOAD CURRENT ADDRESS & IE!60005 AND GO
007144 011676 060105 011670             TMCA,100!60005,TMCS
007172 000002                           RTI

007174 012777 007220 002366     RW.TM:  MOV     #X.TM,@TMTRAP   ;TRAP ADDRESS
007202 010667 000056                    MOV     (6)+,TMRET      ;SAVE ADDRESS
007206 005726                           TST     (6)+            ;RESTORE STACK
007210 012777 060117 002452             MOV     #60117,@TMCS    ;REWIND
007216 000002                           RTI

007220 000177 000040            X.TM:   JMP     @TMRET

                                R.TM:   CHK.TM                  ;ANY ERRORS?
                                        COMCOR                  ;COMPLIMENT CORE
                                        REWIND
007232 012777 007132 002330             MOV     #W.TM,@TMTRAP   ;TRAP ADDRESS
007240 016777 002460 002426             MOV     WRDCNT,@TMBC    ;BYTE COUNT
007246 006377 002422                    ASL     @TMBC           ;MAKE BYTE COUNT
007252 104436                           .START                  ;LOAD CURRENT ADDRESS & IE!60003 AND GO
007254 011676 060103 011670             TMCA,100!60003,TMCS
007262 000002                           RTI

007264 000000                   TMRET:  0
```

```
007266  005067  001422          CH.TM:  CLR     EX              ;CLEAR ERROR FLAG
007272  005777  002372                  TST     @TMCS           ;ANY ERRORS?
007276  100026                          BPL     1$              ;BRANCH IF NO ERRORS
007300  005267  001410                  INC     EX              ;SET ERROR FLAG
007304  032737  020000  177570          BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
007312  001020                          BNE     1$              ;INHIBIT TYPEOUTS
007314  017705  002350                  MOV     @TMCS,TTY       ;SET TTY FOR TYPING
007320  000004  011306                  TYPE    ,TM.M1          ;TYPE DEVICE MESSAGE
007324  000004  011012                  TYPE    ,M3             ;TYPE CS
007330  004767  003032                  JSR     PC,PRINTR       ;TYPE TMCS IN OCTAL
007334  000004  011022                  TYPE    ,M4             ;TYPE ER
007340  017705  002336                  MOV     @TMER,TTY       ;TYPE @TMER IN OCTAL
007344  004767  003016                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
007350  104432                          STOP                    ;HALT ON ERROR
007352  000002                          RTI                     ;RETURN
007354  032737  020000  177570  1$:     BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
007362  001054                          BNE     2$              ;YES!
007364  117767  002300  001320          MOVB    @TMCS,CHK       ;GET EA BITS
007372  116767  171416  001313          MOVB    MX,CHK+1        ;GET MX BITS
007400  105267  001307                  INCB    CHK+1           ;INCREMENT INTO EA BITS
007404  106067  001302                  RORB    CHK             ;MOVE OVER BY 1
007410  006067  001276                  ROR     CHK             ;MOVE IT
007414  006067  001272                  ROR     CHK             ;INTO
007420  006067  001266                  ROR     CHK             ;POSITION
007424  042767  176374  001260          BIC     #176374,CHK     ;CLEAR JUNK
007432  126767  001254  001253          CMPB    CHK,CHK+1       ;MAKE SURE EA BITS INCREMENT
007440  001425                          BEQ     2$              ;RETURN IF EQUAL
007442  000004  011206                  TYPE    ,TM.M1          ;TYPE DEVICE MESSAGE
007446  000004  011142                  TYPE    ,M15            ;TYPE "BANK"
007452  016705  171336                  MOV     MX,TTY          ;TYPE MX IN OCTAL
007456  004767  002714                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
007462  000004  011063                  TYPE    ,M8             ;TRUE
007466  116705  001221                  MOVB    CHK+1,TTY       ;GET BYTE
007472  004767  002700                  JSR     7,PRINTS        ;TYPE TRUE EA BITS
007476  000004  011066                  TYPE    ,M9             ;RECEIVED
007502  116705  001204                  MOVB    CHK,TTY         ;GET BYTE
007506  004767  002664                  JSR     7,PRINTS        ;TYPE RECEIVED EA BITS
007512  104432                          STOP                    ;WAIT
007514  000002                  2$:     RTI                     ;RETURN
```

```
                                        ;DM11 PRIMING ROUTINE.  THIS ROUTINE SETS UP THE DM11 TO TRANSMIT
                                        ;A 400(8) BINARY COUNT PATTERN ON ALL 16 LINES.

                                TRP.DM: NEXT
                                        MOV     MX,TTY          ;GET MX
                                        BIS     TTY             ;WASH 813.,TTY#
                                        BIS     TTY
                                        BIS     TTY
                                        BIS     TTY
                                        BIC     #17777,TTY      ;CLEAR JUNK
                                        MOV     TTY,DMCAT       ;ADDRESS OF CAT
                                1$:     MOV     DMCAT,DMACT     ;SET UP WORD COUNT
                                        ADD     #32.,DMACT
                                        MOV     DMACT,DMBAT     ;SET BASE ADDRESS
                                        ADD     #32.,DMBAT
                                        MOV     DMBAT,TUNTAB    ;SET UP TUMBLE TABLE
                                        ADD     #64.,TUNTAB
                                        MOV     TUNTAB,BINCNT   ;SET UP BINARY COUNT
                                        ADD     #128.,BINCNT
                                        MOV     R2,-(6)         ;SAVE R2 & R1
                                        MOV     R1,-(6)         ;ON THE STACK
                                        MOV     LOLIM,R2
                                        MOV     #16.,R1
                                DMPRIA: MOV     BINCNT,(2)+     ;LOAD CURRENT ADDRESS TABLE
                                        DEC     R1
                                        BNE     DMPRIA
                                        MOV     #16.,R1
                                DMPRIB: MOV     #-400,(2)+      ;LOAD WORD COUNT TABLE
                                        DEC     R1
                                        BNE     DMPRIB
                                        ADD     #2.,R2          ;BUMP TO TUNTAB-2
                                        CLR     (2)+
                                        MOV     #64.,R1
                                DMPRIC: CLR     (2)+            ;CLEAR TUMBLE TABLE
                                        DEC     R1
                                        BNE     DMPRIC
                                        CLR     R1
```

```
                                        DMPRID: MOVB    R1,(2)+         ;GENERATE BINARY COUNT
                                                INCB    R1              ;THIS WILL BE THE TRANSMITTED DATA
                                                BNE     DMPRID
                                                CLR     @DMCS
                                                CLR     @DMACT
                                                CLR     @DMBRK
                                                CLR     DMTDAT
                                                MOV     DMCAT,@DMRDR
                                                MOV     LOLIM,DMPTR
                                                ADD     #200,DMPTR
                                                MOV     DMPTR,ENDPTR
                                                ADD     #200,ENDPTR
                                                MOV     DMVEC,R1
                                                MOV     @DMRINT,(1)+
                                                MOV     #240,(1)+
                                                MOV     @DMTINT,(1)+
                                                MOV     #240,(1)
                                                MOV     RX,TTY
                                                ADD     TTY
                                                BIC     #177717,TTY
                                                MOV     TTY,@DMCS
                                                BIS     #10105,@DMCS
                                                MOV     DMBAIN,@DMACT   ;START TRANSMITTING
                                                MOV     (6)+,R1
                                                MOV     (6)+,R2
                                                CLR     @#177776        ;CLEAR PS
                                                JMP     WAIT

                                        ;DH11 TRANSMITTER INTERRUPT SERVICE ROUTINE

                                        DMTINT: BIT     #60000,@DMCS    ;TEST FOR ERROR FLAGS
                                                BEQ     DMTINC          ;BRANCH IF NO ERROR FLAGS
                                                TYPE    ,DM.MI
                                                TYPE    ,M3
                                                MOV     @DMCS,TTY       ;TYPE @DMCS IN OCTAL
                                                JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
                                                STOP
                                                BIC     #60000,@DMCS    ;CLEAR ERROR FLAGS
                                        DMTINC: TST     @DMCS           ;TEST FOR READY FLAG
                                                BPL     1$
                                                BIC     #100000,@DMCS   ;CLEAR READY
                                        1$:     RTI                     ;EXIT

                                        DMTDAT: 0                       ;STORAGE LOCATION FOR LAST TRANSMITTED CHAR
                                        DMPTR:  0                       ;DH11 SOFTWARE TUMBLE TABLE POINTER
                                        ENDPTR: 0
                                        DMBAIN: 177777                  ;INITIALIZE TO TRANSMIT ON ALL LINES.
                                        MASK:   177400
                                        DMCAT:  0
                                        DMACT:  0
                                        DMBRT:  0
                                        TUMTAB: 0
                                        BINCNT: 0
```

```
                                        ;DM11 RECEIVER INTERRUPT SERVICE ROUTINE
                                
010166 010246                   DMRI.T: MOV     R2,-(6)
010170 016702 177750                    MOV     DMPTR,R2
010174 104406                           SETBANK
010176 005712                           TST     (2)             ;TEST FOR VALID DATA ENTRY
010200 100446                           BMI     DMRINC          ;BRANCH IF VALID DATA ENTRY
010202 000004 011327                    TYPE    ,DM.M1
010206 000004 010212                    TYPE    ,.+2            ;.ASCIZ "  NO DATA RECEIVED"
010212 104432                           STOP
010214 000457                           BR      DMRINX          ;GO TO EXIT
                                
010216 005742                   DMRINB: TST     -(2)            ;CHECK LAST ENTRY
010220 001421                           BEQ     1$              ;OK IF 0
010222 000004 011327                    TYPE    ,DM.M1
010226 000004 010256                    TYPE    ,.+2            ;.ASCIZ "  TABLE OVERFLOW"
010232 012642                           MOV     (6)+,R2         ;RESTORE R2
010234 022626                           CMP     (6)+,(6)+       ;CLEAR RETURN
010236 000167 177206                    JMP     TRP.DM          ;LOOP
010242 005722                   1$:     TST     (2)+            ;MOD POINTER
010244 005712                           TST     (2)             ;VALID DATA ENTRY?
010246 100031                           BPL     DMRINX          ;EXIT IF NO MORE ENTRIES
010250 032712 017000                    BIT     #17000,(2)      ;DATA ON LINE 0?
010254 001013                           BNE     DMRIND          ;BRANCH IF DATA RECEIVED ON OTHER THAN LINE 0
010256 146712 177622                    BICB    MASK,(2)        ;CLEAR NON TRANSMITTED BITS
010262 016767 177606 173352             MOV     DMTDAT,DATA
010270 121267 173346                    CMPB    (2),DATA        ;COMPARE RECEIVED & TRANSMITTED DATA
010274 001401                           BEQ     1$
010276 104434                           ERR3
010300 105267 177570            1$:     INCB    DMTDAT          ;FORM WHAT NEXT RECEIVED CHAR. SHOULD BE
010304 005022                   DMRIND: CLR     (2)+            ;CLEAR TUMBLE TABLE ENTRY
010306 020267 177566                    CMP     R2,ENDPTR       ;IS POINTER AT THE END OF THE TABLE
010312 001004                           BNE     1$
010314 016702 001362                    MOV     LOLIM,R2
010320 062702 000200                    ADD     #200,R2
010324 010267 177546            1$:     MOV     R2,DMPTR        ;RESTORE POINTER
010330 000721                           BR      DMRINB          ;LOOK AT NEXT ENTRY
                                
010332 012602                   DMRINX: MOV     (6)+,R2         ;RESTORE R0
010334 005777 001274                    TST     @DMACT
010340 001003                           BNE     BICX
010342 022626                           CMP     (6)+,(6)+
010344 000167 177100                    JMP     TRP.DM
                                
010350 042777 000200 001254     BICX:   BIC     #200,@DMCS      ;CLEAR DONE FLAG
010356 000002                           RTI
```

```
 1467  010426  104440                SET.DR: SETBRK
 1468  010430  104420                CHK.DR
 1469  010432  005767  000256                TST     EX              ;ANY ERRORS
 1470  010436  001006                BNE     TRP.DR          ;YES
 1471  010440  016702  001274                MOV     LOLIM,R2        ;SET ADDRESS
 1472  010444  012767  000001  173236        MOV     #1,DATA         ;SET DATA
 1473  010452  026712  173232        NX.DR:  CMP     DATA,(2)        ;CHECK FIRST WORD
 1474  010456  001401                BEQ     .+4             ;SAME?
 1475  010460  104434                EMT3                    ;NO
 1476  010462  062702  000002                ADD     #2,R2           ;INC ADDRESS
 1477  010466  026712  173216                CMP     DATA,(2)        ;SECOND WORD OK?
 1478  010472  001401                BEQ     .+4             ;NO
 1479  010474  104434                EMT3                    ;UP IT
 1480  010476  062702  000002                ADD     #2,R2           ;GET NEXT DATA
 1481  010502  005767  173206                INC     DATA            ;END?
 1482  010506  026702  001230                CMP     HILIM,R2        ;NO?
 1483  010512  001357                BNE     NX.DR
 1484  010514  104442                TRP.DR: NEXT
 1485  010516  005037  177776                CLR     @#177776        ;CLR PS
 1486  010522  016702  001212                MOV     LOLIM,R2        ;SET ADDRESS
 1487  010526  012767  000001  173154        MOV     #1,DATA         ;SET DATA
 1488  010534  016722  173150        NX1.DR: MOV     DATA,(2)+       ;SET UP DATA
 1489  010540  005167  173144                COM     DATA            ;- NEXT DATA
 1490  010544  016722  173140                MOV     DATA,(2)+       ;LOAD IT
 1491  010550  005467  173134                NEG     DATA            ;MAKE IT POSITIVE
 1492  010554  026702  001162                CMP     HILIM,R2        ;END?
 1493  010560  001365                BNE     NX1.DR          ;NO?
 1494  010562  012737  000240  177776        MOV     #240,@#177776   ;LOCK AGAIN
 1495  010570  012777  010426  000766        MOV     #SET.DR,@ORTRAP ;SET VECTOR
 1496  010576  016777  001142  001104        MOV     WRDCNT,@ORWC    ;SET WORD COUNT
 1497  010604  104436                ,START                  ;LOAD CURRENT ADDRESS & IE!10001 AND GO
 1498  010606  011714  010101  011714        ORCA,100!10001,ORCS
 1499  010614  000002                RTI
```

```
1500  010616  005067  000072          CH.DR:  CLR     EX              ;CLEAR ERROR FLAG
1501  010622  005777  001066                  TST     @DRCS           ;ANY ERRORS?
1502  010626  100030                          BPL     1$              ;BRANCH IF NO ERRORS
1503  010630  005777  001054                  TST     @DRMC           ;IS IT BANK 7?
1504  010634  001452                          BEQ     1$              ;YES - GET OUT
1505  010636  005267  000052                  INC     EX              ;SET ERROR FLAG
1506  010642  032737  020000  177570          BIT     #SW13,@#SWR     ;INHIBIT ERROR TYPEOUT
1507  010650  001017                          BNE     1$              ;INHIBIT TYPEOUTS
1508  010652  017706  001036                  MOV     @DRCS,TTY       ;SET TTY FOR TYPING
1509  010656  000004  011345                  TYPE    ,DR.MI          ;TYPE DEVICE MESSAGE
1510  010662  000004  011012                  TYPE    ,M3             ;TYPE CS
1511  010666  004767  001474                  JSR     PC,PRINTR       ;TYPE DRCS IN OCTAL
1512  010672  000004  011022                  TYPE    ,M4             ;TYPE ER
1513  010676  017706  001014                  MOV     @DRER,TTY       ;TYPE @DRER IN OCTAL
1514  010702  004767  001460                  JSR     PC,PRINTR       ;TYPE LEADING ZERO'S
1515  010706  104432                          STOP                    ;HALT ON ERROR
1516  010710  000002                  1$:     RTI                     ;RETURN
```

```
                                CHK:    0
                                EX:     0
                M1:     .ASCII  <15><12>"RC11,RP11,RF11,RK11,TM11,TC11,DR11B, OR DU11 AND "
                        .ASCIZ  "<RETURN>"
                M3:     .ASCIZ  "  CS = "
                MM:     .ASCIZ  "  ER = "
                M7:     .ASCIZ  <15><12>"DATA ERROR AT "
                M8:     .ASCIZ  "  TRUE = "
                M9:     .ASCIZ  "  RECEIVED = "
                M11:    .ASCIZ  "LOWER BANK = "
                M12:    .ASCIZ  "UPPER BANK = "
                M13:    .ASCIZ  <15><12>
                M15:    .ASCIZ  "  BANK"

                RF.M1:  .ASCIZ  <15><12>"RF11 DISK ERROR:"
                RC.M1:  .ASCIZ  <15><12>"RC11 DISK ERROR:"
                RK.M1:  .ASCIZ  <15><12>"RK11 DISK ERROR:"
                RP.M1:  .ASCIZ  <15><12>"RP11 DISK ERROR:"
                TC.M1:  .ASCIZ  <15><12>"DECTAPE ERROR:"
                TM.M1:  .ASCIZ  <15><12>"MAGTAPE ERROR:"
```

```
1573 011322 047522 036122    000
1574 011327    015 042012 030515  DM.M1:  .ASCIZ  <15><12>"DM11 ERROR:"
1575 011334 020061 061105 047522
1576 011342 036122    000
1577 011345    015 042012 030522  DR.M1:  .ASCIZ  <15><12>"DR11 ERROR:"
1578 011352 020061 061105 047522
1579 011360 036122    000
1580 011363    015 005012 052113  SEGA:   .ASCIZ  <15><12><12>"KT11 ERROR:"
1581 011370 030461 042440 051122
1582 011376 051117 000072
1583 011402 005015 051123 020060  SEG1:   .ASCIZ  <15><12>"SR0 = "
1584 011410 020075    000
1585 011413    015 051412 030522  SEG2:   .ASCIZ  <15><12>"SR1 = "
1586 011420 036440 000040
1587 011424 005015 051123 020062  SEG3:   .ASCIZ  <15><12>"SR2 = "
1588 011432 020075    000
1589 011435    077 005015    000  QUES:   .ASCIZ  "?"<15><12>
1590 011441    040 044527 044124  WITHSG: .ASCIZ  " WITH KT11"
1591 011446 045440 030524 000061
1592 011454 053440 052111 020110  WITHMX: .ASCIZ  " WITH MX11"
1593 011462 054115 030461    000
1594        011470                        .EVEN
1595 011470 041522 043122 045522  DEVTAB: .ASCII  "RCRFRKRPTCTMDRDM"
1596 011476 050122 041524 046524
1597 011504 051104 046504
1598 011510 000000                         0
1599 011512 177777 177777 000002  TAB1:   -1,-1,2,3,-1,4,-1,-1
1600 011520 000003 177777 000004
1601 011526 177777 177777
1602 011532 000000                 DEVADR: 0
1603 011534 000003                 PSCNT:  3
1604 011536 000000                 FLG:    0
1605
1606 011540 000204 000206         RFTRAP: 204,206
1607 011544 000210 000212         RCTRAP: 210,212
1608 011550 [illegible] [illegible]  RPTRAP: [illegible]
1609 011554 [illegible] [illegible]  RKTRAP: [illegible]
1610 011560 000214 000216         TCTRAP: 214,216
1611 011564 000124 000126         DRTRAP: 124,126
1612 011570 [illegible] [illegible]  TMTRAP: [illegible]
1613 011574 [illegible] [illegible]  DMRVEC: [illegible]
1614 011600 000250 000252         SEGVEC: 250,252
```

```
1615  011604  177460          RFCS:   177460          ;RF11  DISK  (256K)
1616  011605  177462          RFER:   177462
1617  011606  177464          RFWC:   177464
1618  011607  177466          RFCA:   177466
1619  011610  177470          RFDA:   177470
1620  011611  177440          RCCS:   177440          ;RC11  DISK  (64K)
1621  011612  177442          RCER:   177442
1622  011613  177444          RCWC:   177444
1623  011614  177446          RCCA:   177446
1624  011615  177450          RCDA:   177450
1625  011616  176700          RPCS:   176700          ;RP11  DISK  (PACK)
1626  011617  176702          RPWC:   176702
1627  011620  176704          RPBA:   176704
1628  011621  176706          RPDA:   176706
1629  011622  176710          RPMR:   176710
1630  011623  176712          RPDS:   176712
1631  011624  177400          RKCS:   177400          ;RK11  DISK  (CARTRIDGE)
1632  011625  177402          RKWC:   177402
1633  011626  177404          RKBA:   177404
1634  011627  177406          RKDA:   177406
1635  011630  177410          RKCA:   177410
1636  011631  177412          RKDB:   177412
1637  011632  172540          TCCS:   172540          ;TC11  DECTAPE
1638  011633  172542          TCDT:   172542
1639  011634  172544          TCWC:   172544
1640  011635  172546          TCCA:   172546
1641  011636  172520          TMCS:   172520          ;TM11 MAGTAPE
1642  011637  172522          TMBR:   172522
1643  011640  172524          TMCC:   172524
1644  011641  172526          TMCA:   172526
1645  011642  170400          DMCS:   170400          ;DM11  ASYNCRONOUS MODEM
1646  011643  170402          DMCT:   170402
1647  011644  170404          DMBA:   170404
1648  011645  170406          DMBC:   170406          ;DR11
1649  011646  167400          DRCS:   167400
1650  011647  167402          DRCA:   167402
1651  011650  167404          DRBA:   167404
1652  011651  167406          DRBC:   167406
1653  011652  172516          KTCSR:  172516          ;KT11 - MEMORY MANAGEMENT
1654  011653  172520          SR0:    172520
1655  011654  172522          KISAR0: 172340
1656  011655  172524          KISAR1: 172342
1657  011656  172526          KISAR2: 172344
1658  011657  172530          KISAR3: 172346
1659  011660  172532          KISAR4: 172350
1660  011661  172534          KISAR5: 172352
1661  011662  172536          KISAR6: 172354
1662  011663  172540          KISAR7: 172356
1663  011664  177604          MDC3:   177604          ;MX11
1664
1665  011665  140000          LOLIM:  140000          ;BEGINNING OF BANK
1666  011666  160000          HILIM:  160000          ;END OF BANK
1667  011667  170000          WRDCNT: -10000          ;WORD COUNT
```

```
                                ;       SLOADR          GET AND RESTORE THE LOADER
                                ;
                                ;THESE ROUTINES FIRST FIND THE LOADER (TOP OF HIGHEST
                                ;4K BANK IN 22K) AND SAVE IT AT TAG "ENDP:". ENTRY FOR
                                ;THIS ROUTINE IS "JSR PC,LODGET".
                                ;
                                ;"JSR PC,LODRES" WILL RESTORE THE LOADER IN ITS ORIGONAL
                                ;LOCATION. ".LOD:" CONTAINS THE ADDRESS OF THE LOADER.
                                ;
012746  005767  167080  LODGET: TST     OPTION          ;WHICH OPTION?
012752  100405              BLE     2$              ;NO MEMORY MANAGEMENT
012754  005077  177740          CLR     @#MMR0          ;TURN OFF MM
012760  010001          2$:     MOV     R0,R1           ;GET THE BANK
012762  020027  000006          CMP     R0,#6           ;IS IT > 6?
012766  002402              BLT     1$              ;SKIP IF LES THAN 6
012770  012700  000006          MOV     #6,R0           ;IF > 7 MAKE 7
012774  006301          1$:     ASL     R1              ;GET THE
012776  006301                  ASL     R1              ;UPPER
013000  006301                  ASL     R1              ;THREE
013002  006301                  ASL     R1              ;BITS
013004  042701  017777          BIC     #17777,R1       ;CLEAR JUNK
013010  012702  013062          MOV     #ENDP,R2        ;GET SAVE ADDRESS
013014  060201                  ADD     R2,R1           ;MAKE OTHER ONE
013016  010112  000060          MOV     R1,.LOD         ;SAVE ADDRESS
013022  012122          2$:     MOV     (1)+,(2)+       ;MOVE WORD
013024  020227  020000          CMP     R2,#20000       ;END?
013030  001373              BNE     2$              ;NO!
013032  000207                  RTS     PC              ;RETURN
                                ;
013034  005767  166762  LODRES: TST     OPTION          ;WHICH OPTION?
013040  100405              BLE     2$              ;NO MEMORY MANAGEMENT
013042  005077  177762          CLR     @#MMR0          ;TURN OFF MM
013046  012702  013062          MOV     #ENDP,R2        ;GET END OF PROGRAM
013052  016201  000024          MOV     .LOD,R1         ;GET SAVE ADDRESS
013056  012221          1$:     MOV     (2)+,(1)+       ;RESTORE WORD
013060  020227  020000          CMP     R2,#20000       ;END?
013064  001374              BNE     1$              ;LOOP
013066  005767  166730          TST     OPTION          ;WHICH OPTION?
013072  100405              BLE     2$              ;NO MEMORY MANAGEMENT
013074  005277  177620          INC     @#MMR0          ;TURN ON MM
013100  000207          2$:     RTS     PC              ;RETURN
                                ;
013102  000000  .LOD:   0                               ;STARTING ADDRESS OF LOADER
```

```
                                    ;       $TYPE           MESSAGE TYPEOUT ROUTINE

                                    ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
                                    ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
                                    ;MESSAGE STARTING IN LOCATION "ADR:", 2) "TYPE ,CHAR" - TYPES
                                    ;THE ASCII "CHAR", AND 3) "PRINT (<15><12>"MESSAGE">) - TYPES
                                    ;THE MESSAGE WHICH IS INLINE ASCII.

012104  010546                      IOTS:   MOV     TTY,-(6)        ;SAVE TTY R5=TTY
012106  016605  000022                      MOV     22(6),TTY       ;GET ADDRESS TO BE TYPED
012112  032705  177400                      BIT     #177400,TTY     ;IS IT A TYPEM?
012116  001004                              BNE     1$              ;NO
012120  010567  000064                      MOV     TTY,.TYPE       ;GET THE CHARACTER
012124  012705  012210                      MOV     #.TYPE,TTY      ;FUDGE THE ADDRESS
012130  105715                      1$:     TSTB    (TTY)           ;TERMINATOR?
012132  001406                              BEQ     2$              ;GET OUT IF SO
012134  112537  177566                      MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
012140  105737  177564                      TSTB    @#177564                ;IS THE PRINTER READY
012144  100375                              BPL     .-4             ;WAIT UNTIL IT IS
012146  000770                              BR      1$              ;GET THE NEXT CHARACTER
012150  017646  000002              2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
012154  062776  000002  000004              ADD     #2,@4(6)        ;ADD 2 TO THE ADDRESS
012162  022676  000002                      CMP     (6)+,@2(6)      ;IS IT .+2?
012166  001003                              BNE     3$              ;NO
012170  062705  000002                      ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
012174  042705  000001                      BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
012200  010566  000002                      MOV     TTY,2(6)        ;RESTORE ADDRESS
012204  012605                      3$:     MOV     (6)+,TTY        ;RESTORE TTY
012206  000002                              RTI                     ;RETURN
012210  000000                      .TYPE:  0                       ;CHARACTER TYPE LOCATION

012212  010346                      READS:  MOV     R3,-(6)         ;SAVE R3
012214  012703  012316              1$:     MOV     #INPUT,R3       ;GET ADDRESS
012220  022703  012336              2$:     CMP     #INPUT+20,R3    ;BUFFER FULL?
012224  001412                              BEQ     4$              ;YES - TYPE "?"
012226  105737  177560                      TSTB    @#177560        ;WAIT FOR
012232  100375                              BPL     .-4             ;A CHARACTER
012234  113713  177562                      MOVB    @#177562,(3)    ;GET CHARACTER
012240  142713  000200                      BICB    #200,(3)        ;GET RID OF JUNK
012244  122713  000177                      CMPB    #177,(3)        ;IS IT A RUBOUT
012250  001006                              BNE     3$              ;SKIP IF NOT
012252                              4$:
012252  000004  012256                      TYPE    ,.+2            ;.ASCIZ "?"<15><12>
012262  000754                              BR      1$              ;ZAP THE BUFFER AND LOOP
012264  111367  177720              3$:     MOVB    (3),.TYPE       ;SET UP FOR TYPING
012270  000004  012210                      TYPE    ,.TYPE          ;ECHO IT
012274  122723  000015                      CMPB    #15,(3)+        ;CHECK FOR RETURN
012300  001347                              BNE     2$              ;LOOP IF NOT RETURN
012302  105063  177777                      CLRB    -1(3)           ;ZAP RETURN (THE 15)
012306  000004  000012                      TYPE    ,12             ;TYPE A LINE FEED
012312  012603                              MOV     (6)+,R3         ;RESTORE R3
012314  000207                              RTS     PC              ;RETURN
012316  000020                      INPUT:  .BLKW   20              ;TTY INPUT AREA
```

```
                                          ;       $OCTAL          OCTAL TYPEOUT ROUTINE

                                          ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
                                          ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
                                          ;THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.

 012356  012767  170101  000202  BITYPS: MOV     #170101,.PR        ;SET BIT FLAG ANS 16. CHARACTER COUNT
 012364  000411                          BR      .PTIT              ;NOW TYPE IT IN BIT FORM
 012366  112767  000001  000172  PRINTR: MOVB    #1,.PR             ;SET ZERO FILL SWITCH
 012374  000402                          BR      .+6                ;SKIP
 012376  005067  000164          PRINTS: CLR     .PR                ;SUPRESS LEADING ZERO'S
 012402  112767  177772  000157          MOVB    #-6,.PR+1          ;SET COUNT
 012410  010446                  .PTIT:  MOV     R4,-(6)            ;SAVE R4
 012412  013704  012570                  MOV     @.PR+2,R4          ;SET POINTER TO FIRST ASCII CHAR.
 012416  105014                          CLRB    (4)                ;CLEAR FIRST BYTE
 012420  000432                          BR      .PRF               ;ROTATE FIRST BIT
 012422  010446                  PRINTA: MOV     R4,-(6)            ;SAVE R4
 012424  013704  012570                  MOV     @.PR+2,R4          ;SET UP POINTER TO OUTPUT AREA
 012430  113714  [illegible]             MOVB    MX,(4)             ;MX CONTAINS UPPER 5 BITS
 012434  006306                          ASL     TTY                ;GET RID
 012436  006306                          ASL     TTY                ;OF 3
 012440  006306                          ASL     TTY                ;JUNK BITS
 012442  106214                          ASRB    (4)                ;GET BIT13
 012444  006006                          ROR     TTY                ;PACK IT
 012446  106214                          ASRB    (4)                ;GET BIT14
 012450  006006                          ROR     TTY                ;PACK IT
 012452  152724  000060                  BISB    #'0,(4)+           ;MAKE IT ASCII
 012456  012767  175401  000102          MOV     #175401,.PR        ;-5,,1 - 5 BYTES AND FILL
 012464  105014                  .PRL:   CLRB    (4)                ;CLEAR BYTE OF CHARACTER
 012466  032767  000100  000072          BIT     #100,.PR           ;BIT TYPING MODE?
 012474  001004                          BNE     .PRF               ;YES - SKIP 2 ROTATES
 012476  006106                          ROL     TTY                ;ROTATE BIT INTO C
 012500  106114                          ROLB    (4)                ;PACK IT
 012502  006106                          ROL     TTY                ;ROTATE BIT INTO C
 012504  106114                          ROLB    (4)                ;PACK IT
 012506  006106                  .PRF:   ROL     TTY                ;ROTATE BIT INTO C
 012510  106114                          ROLB    (4)                ;PACK IT
 012512  105714                          TSTB    (4)                ;IS IT ZERO?
 012514  001402                          BEQ     .+6                ;SKIP INC
 012516  105267  000044                  INCB    .PR                ;SET FILL SWITCH
 012522  105767  000040                  TSTB    .PR                ;CHECK FILL SWITCH
 012526  001402                          BEQ     .+6                ;SKIP BITSET
 012530  152724  000060                  BISB    #'0,(4)+           ;MAKE INTO ASCII CHAR
 012534  105267  000027                  INCB    .PR+1              ;INC COUNT
 012540  [illegible]                     BNE     .PRL               ;REPEAT
 012542  023704  012570                  CMP     @.PR+2,R4          ;EMPTY BUFFER?
 012546  001002                          BNE     .+6                ;SKIP IF NOT
 012550  112724  000060                  MOVB    #'0,(4)+           ;LOAD 1 ZERO
 012554  105014                          CLRB    (4)                ;NULL TERMINATOR
 012556  000004  012570                  TYPE    ,.PR+2             ;TYPE IT
 012562  012604                          MOV     (6)+,R4            ;RESTORE R4
 012564  000207                          RTS     PC                 ;RETURN
 012566  000012                  .PR:    .BLKW   12                 ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
 1815  012612  012777  012740  000126  PDOWNS: MOV     #ILLUP,@PUVECS  ;SET FOR FAST UP
 1816  012614  012777  000340  000122          MOV     #340,@PUVECS+2  ;PRIO:7
 1817  012616  010046                          MOV     R0,-(6)         ;PUSH R0 ON STACK
 1818  012620  010146                          MOV     R1,-(6)         ;PUSH R1 ON STACK
 1819  012622  010246                          MOV     R2,-(6)         ;PUSH R2 ON STACK
 1820  012624  010346                          MOV     R3,-(6)         ;PUSH R3 ON STACK
 1821  012626  010446                          MOV     R4,-(6)         ;PUSH R4 ON STACK
 1822  012630  010546                          MOV     R5,-(6)         ;PUSH R5 ON STACK
 1823  012632  010667  000074                  MOV     SP,SAVR6        ;SAVE SP
 1824  012636  012777  012566  000072          MOV     #PUPS,@PUVECS   ;SET UP VECTOR
 1825  012644  000000                          HALT                    ;WAIT FOR PF
 1826
 1827  012646  016706  000062          PUPS:   MOV     SAVR6,SP        ;GET SP
 1828  012652  005001                          CLR     R1              ;WAIT LOOP FOR THE TTY
 1829  012654  005201                  1$:     INC     R1              ;WAIT FOR THE INC
 1830  012656  001376                          BNE     1$              ;OF WORD
 1831  012660  012605                          MOV     (6)+,R5         ;POP STACK INTO R5
 1832  012662  012604                          MOV     (6)+,R4         ;POP STACK INTO R4
 1833  012664  012603                          MOV     (6)+,R3         ;POP STACK INTO R3
 1834  012666  012602                          MOV     (6)+,R2         ;POP STACK INTO R2
 1835  012670  012601                          MOV     (6)+,R1         ;POP STACK INTO R1
 1836  012672  012600                          MOV     (6)+,R0         ;POP STACK INTO R0
 1837  012674  012737  012612  000024          MOV     #PDOWNS,@#24    ;SET UP THE POWER DOWN VECTOR
 1838  012702  012737  000340  000026          MOV     #340,@#26       ;PRIO:7
 1839  012710  000004  012724                  TYPE    ,+2             ;.ASCIZ <15><12>"POWER"
 1840  012714  000167  167230                  JMP     @TPF            ;JMP TO USER ADDRESS
 1841
 1842  012720  000000          ILLUP:  HALT                    ;THE POWER UP SEQUENCE WAS STARTED
 1843  012722  000776                  BR      .-2             ; BEFORE THE POWER DOWN WAS COMPLETE
 1844
 1845  012724  000000          SAVR6:  0                       ;PUT THE SP HERE
 1846  012726  000024  000026  PUVECS: 24,26                   ;POWER UP VECTOR
 1847
 1848  012732  116767  164604  177566  READR:  MOVB    177562,INPUT    ;READ THE CHARACTER
 1849  012740  142767  000200  177560          BICB    #200,INPUT      ;CLEAR JUNK
 1850  012746  122767  000003  177552          CMPB    #3,INPUT        ;IS IT A ↑C?
 1851  012754  001401                          BEQ     .+4             ;SKIP IF ↑C
 1852  012756  000002                          RTI                     ;RETURN IF NOT
 1853  013000  000004  013004                  TYPE    ,+2             ;.ASCIZ "↑C"<177>
 1854  013004  005067  000002                  CLR     @#2             ;DO NOT LOOP
 1855  013010  000137  001024                  JMP     @#BEGIN         ;RESTART
 1856
 1857  013014  000000          SAV6:   0
 1858  013016  000010          PATCH1: .BLKW   10              ;PATCH AREA #1
 1859  013036  000010          PATCH2: .BLKW   10              ;PATCH AREA #2
 1860  013056  000000          ENDP:   0                       ;LAST LOC OF PROGRAM
 1861  013062  000001                  .END
```

```
ABRT    001742     472#   480    490
ADRTAB  002750     608    615#
ALLOR   003442     720#   743
BADDEV  002122     455    456#
BEGIN   001024     367    370#   597    1065
BELL  = 000007     334#   657
BICX    010416     1461   1465#
BINCNT  010164     1347#  1348#  1363   1420#
BIT     003712     708#   720    731#   786#   790#
BITYPS  012366     1790#
CDATA   003264     627    703#
CHK     010712     874#   875#   896#   897#   898#   899#   900#   901#   902    909    912    959#   960#
                   961#   962#   963#   964#   965#   966#   967    974    977    1034#  1035#  1036#  1037#
                   1038#  1039#  1040#  1041#  1042   1049   1052   1102#  1103#  1104#  1105#  1106#  1107#
                   1108#  1109#  1110   1117   1120   1163#  1164#  1165#  1166#  1167#  1168#  1169#  1170#
                   1171   1178   1181   1308#  1309#  1310#  1311#  1312#  1313#  1314#  1315#  1316   1323
                   1326   1517#
CHK.DR= 104420     624#   1468
CHK.RC= 104406     619#   916    926    934
CHK.RF= 104404     618#   851    861    869
CHK.RK= 104412     621#   1056   1066   1074
CHK.RP= 104410     620#   981    991    999
CHK.TC= 104414     622#   1124   1136
CHK.TM= 104416     623#   1251   1260
CH.DR   010616     624    1500#
CH.RC   004742     619    942#
CH.RF   004360     618    877#
CH.RK   005752     621    1082#
CH.RP   005364     620    1014#
CH.TC   006302     622    628    1143#
CH.TM   007266     623    1241#
CKDATA= 104426     627#   862    917    982    1057   1125   1252
CONCOR= 104402     617#   870    925    1000   1075   1135   1261
CORE  = 104400     616#   853    918    983    1068   1127   1263
CRLF    002456     529    551#   569    590
CR1     002510     557#   567
CR2     002544     558    560#
CR3     002602     576#   586
CR4     002630     577    587#
C.RC    004650     919    926#
C.RF    004264     854    861#
C.RK    005640     1059   1066#
C.RP    005232     984    991#
DATA    003710     707#   711    732#   784#   789#   793    814    1446#  1447   1472#  1473   1477   1481#
                   1487#  1488   1489#  1490   1491#
DEVADR  011532     471#   472#   473    477    481    666#   666    1602#
DEVTAB  011470     471    478    1596#
DISPLA= 177570     333#   691#
DMACT   011702     1371#  1384#  1460   1646#
DMADR   011706     1374#  1648#
DMBAIM  010150     1389   1414#
DMBAT   010160     1343#  1344#  1345   1418#
DMBAK   011704     1372#  1647#
DMCAT   010154     1340#  1341   1374   1416#
DMCS    011700     531    1270#  1387#  1388#  1397   1401   1404#  1405   1407#  1465#  1645#
DMPRIA  007640     1353#  1356
```

```
DMPRIB    007654           1357#    1359
DMPRIC    007676           1353#    1355
DMPRIC    007706           1357#    1359
DMPTR     010144           1375*    1376*    1377     1412#    1424     1456*
DMRINB    010242           1433#    1457
DMRINC    010316           1427     1443#
DMRINO    010352           1444     1451#
DMRINT    010166           1380     1423#
DMRINX    010400           1431     1442     1459#
DMRVEC    011574           1379     1613#
DMTDAT    010142           1373*    1411#    1446     1450*
DMTINC    010124           1398     1405#
DMTINT    010064           1380     1397#
DMACT     010156           1391*    1392*    1393     1417#
DM.M1     011327           1399     1420     1435     1574#
DOCONE    001400            419#
DOMX      001434            417      429#
DOSEG     001476            418      437#
DRCA      011712           1498     1650#
DRCS      011714            52[illegible]*  1498     1501     1508     1651#
DRDB      011716           1653#
DRER      011716           1513     1652#
DRTRAP    011564            3[illegible]*   3[illegible]*   493*     1496*    1611#
DRWC      011710           149[illegible]*  1502     1649#
DR.M1     011346           1509     1577#
DUM1   =  104446            [illegible]#
DUMP   =  104450            [illegible]#
ENDP      012052           1667     1698     1860#
ENDPTR    010146           1377*    1378*    1413#    1452
ENDTAB=   003020            609      637#
ERRORS    001002            [illegible]#
ERRB   =  104430            [illegible]#   1217     1280
ERRG   =  104434            630#     713      1202     1236     1449     1475     1479
ER2       002774            630      804#
EX        010714            704      877#     880*     942*     945*     1014*    1017*    1082*    1085*    1143*    1146*    1291*    1294*
                           1469     1500*    1505*    1518#
FLG       011536            662*     662*     683      1604#
FOUBER    006756           1100     1223#
GDATA     002440            710      719#     792
GETMON    003022            475      640#     6[illegible]
HILIM     011742            3[illegible]*   546      715      794      801      840*     1482     1492     1664#
HLT    =  104000            [illegible]#
HLTADR    001012            3[illegible]#
ICNT      001000            3[illegible]#
ILLUP     012740           1815     1842#
INPUT     012316            481*     482*     483*     485      505      509      513      517      521      525      529      534      538
                            5[illegible]     575      6[illegible]     773      807      1741     1742     1761#    1848*    1849*    1850
IOTS      012104            374      1717#
KISAR0    011722            844*     1655#
KISAR6    011726            443*     444      446      828*     846*     1657#
KISAR7    011730            8[illegible]*   1658#
KISDR0    011724            8[illegible]*   16[illegible]#
KISDR6    011720            847*     16[illegible]#
KISDR7    011734            8[illegible]*   1660#
LAD       001010            3[illegible]#
LIMIT     001020            3[illegible]#   466*     507
```

```
LODGET   011746       455     1675#
LOODES   012034       467      640    1695#
LOLIM    011740       376#     545     701     709     791     799     838#   1351    1375    1454    1471    1486    1663#
MAP      004160       634      644#
MAPIT  = 104444       439      634#
MASK     010152      1415#    1445
MEMORY   003714       400#     671     674#    676#    678#    719     786#
MIN      001016       367#     397#    412#    556#    562#    564#    566#    568     570#    571     589     670     679
MX       001014       366#     398#    413#    571#    655     670#    679     690#    691     695     758     770     822
                      886      905     960     971    1036    1046    1103    1114    1164    1175    1309    1320    1334
                     1304     1700    1815
MXC3     011736       410      432#    436#    830#   1661#
M1       010716       540     1519#
M11      011103       553     1542#
M12      011121       572     1545#
M13      011137       593      594    1540#
M15      011142       905      970    1045    1113    1174    1319    1549#
M3       011012       885      950    1022    1090    1151    1299    1400    1510    1530#
M4       011022       887      952    1024    1092    1153    1301    1512    1532#
M7       011032       810     1524#
M8       011053       813      908     973    1048    1116    1177    1322    1537#
M9       011066       816      911     976    1051    1119    1180    1325    1539#
N      = 000052       344#     616     617#    618#    619#    620#    621#    622#    623#    624#    625#    626#    627#
                      628#     629#    630#    631#    632#    633#    634#    635#    636#    637#
NEWDEV   001750       473#     686
NEXT   = 104442       633#     788    1333    1404
NODEV    002650       499      593#
NXT      002044       633      652#
NXTCR    002544       572#     588
NX.OR    010452      1473#    1483
NX1.OR   010534      1488#    1493
OPTION   001022       369#     401#    406#    411#    416     460     823    1675    1695    1703
PATCH1   013022       635     1858#
PATCH2   013046       636     1859#
PC     =%000007       343#     458#    475#    502#    554#    573#    640#    645#    651#    668#    699#    710#    733#
                      792#     812#    815#    818#    886#    889#    907#    951#    954#    972#   1023#   1026#   1047#
                     1091#    1094#   1115#   1152#   1155#   1176#   1300#   1303#   1321#   1402#   1511#   1514#   1693#
                     1706#    1760#   1813#
PCNT     001004       363#
PDOWNB   012512       376     1815#   1837
POTS     003644       629      752#
PRINTA   012422       812     1770#
PRINTR   012366       815      818     886     889     951     954    1023    1026    1091    1094    1152    1155    1300
                     1303     1402    1511    1514    1770#
PRINTS   012376       458      639     907     910     913     972     975     978    1047    1050    1053    1115    1118
                     1121     1176    1179    1182    1321    1324    1327    1772#
PS     = 177776       331#     470#    685#
PSCNT    011524       789#     660#    669#   1603#
PUPS     012666      1824     1827#
PUVECS   012746      1815#    1816#   1824#   1846#
QUES     011426      1590#
RCCA     011654       923      931     940    1623#
RCCS     011516       907#     923     931     940     943     948     959    1620#
RCDA     011636       921#     924#    938#   1624#
RCER     011620       953     1621#
RCTRAP   011544       381#     388#    400#    919#    927#    936#   1607#
```

```
RCWC     011622        920*    929*    937*    1622#
RC.M1    011174        949     959     1556#
RERDR    012752        374     1840#
REROS    012212        502     554     573     1740#
REDO     001712        454     461     464     466#
RETI     006642        1196#   1207    1210
RETI1    007006        1228#   1240    1251
RETR     006712        1210#   1221
RETR1    007056        1243#   1254
RETURN   007130        1193*   1222    1226*   1255    1260#
REWIND=  104424        [illegible]#   1264    1268
RFCA     011612        858     866     875     1618#
RFCS     011604        515*    858     866     875     878     883     894     1615#
RFDA     011614        856*    864*    873*    1619#
RFER     011606        858     1616#
RFTRAP   011540        380*    387*    487*    854*    862*    871*    1606#
RFWC     011610        855*    863*    872*    1617#
RF.M1    011151        894     904     1552#
RKCA     011652        1063    1071    1080    1634#
RKCS     011644        519*    1063    1071    1080    1083    1088    1096*   1097    1102    1631#
RKDA     011654        1061*   1069*   1078*   1636#
RKER     011646        1093    1632#
RKTRAP   011554        383*    390*    490*    1059*   1067*   1076*   1609#
RKWC     011650        1060*   1068*   1077*   1633#
RK.M1    011217        1089    1112    1560#
RPCA     011634        988     996     1005    1627#
RPCS     011630        511*    988     996     1005    1007*   1008    1012*   1015    1020    1028*   1029    1034    1625#
RPDA     011636        986*    994*    1003*   1628#
RPDS     011642        1010    1630#
RPER     011640        1025    1629#
RPTRAP   011550        382*    389*    489*    984*    992*    1001*   1608#
RPWC     011632        985*    993*    1002*   1626#
RP.M1    011242        1021    1044    1564#
RSIT     001734        471#    476
RTPF     002170        494     503#    1840
RW.TM    007174        626     1272#
R.RC     004704        927     934#
R.RF     004322        862     869#
R.RK     006214        1067    1074#
R.RP     005266        992     999#
R.TC     006654        1129    1135#
R.TM     007254        1245    1250#
R0       =%000000      335#    421*    423*    430*    431*    432     434     446*    447*    448*    449*    450*    457
                       452     574*    501*    583*    586*    587     589     643     647*    655     1678    1817    1836*
R1       =%000001      336#    420*    424*    425     558*    575*    1350    1352*   1354*   1356*   1358*   1362*   1364*
                       1366*   1367    1374*   1379*   1380*   1670*   1679    1681*   1682*   1683*   1684*   1685*   1686*
                       1688*   1689    1694*   1818    1828*   1829*   1835*
R2       =%000002      337#    709*    715     723     726     735     736     745     791*    794     799*    801     811
                       1349    1351*   1360*   1391*   1423    1424*   1427*   1452    1454*   1455*   1456    1459*   1471*
                       1476*   1480*   1482    1486*   1492    1687*   1688    1691    1698*   1701    1819    1834*
R3       =%000003      338#    477*    478*    644*    648*    649*    653*    723*    724*    725*    726*    727*    728*
                       729*    735*    737*    742*    745*    746*    747     749*    750*    1740    1741*   1742    1759*
                       1820    1833*
R4       =%000004      339#    545*    546     701*    1774    1776*   1778    1779*   1807    1812*   1821    1832*
R5       =%000005      340#    605*    606*    607*    608*    609     736*    738*    739*    740*    741*    742     747*
                       740*    1822    1831*
```

```
SAV6      013020      1057#
SCF1      007050      1236     1239     1242#
SCF1A     007106      1247     1253#
SCF2      007066      1242     1246#
SCOPE  =  104400       320#
SC1       007016      1222     1231#
SC1A      007036      1232     1238#
SEARCH=   104422       625#    1128     1137
SEGA      011363      1580#
SEGVEC    011600      1614#
SEG1      011402      1583#
SEG2      011413      1586#
SEG3      011424      1587#
SETBAK    004070       632      822#
SETBAN=   104440       632#     700      703      798     1425     1467
SET.DR    010426      1467#    1495
SF1       006704      1203     1206     1209#
SF1A      006742      1214     1220#
SF2       006722      1209     1213#
SIZEIT    001370       403      416#
SP     =%000006        342#     373#     452#     503#    1023     1027#
SRO       011720       405      440#    1654#    1677#    1697#    1706#
SR1       006652      1190     1198#
SR1A      006672      1199     1205#
STOP   =  104432       629#     819      880      914      955      979     1027     1054     1095     1122     1156     1183     1304
                      1328     1403     1430     1515
SWR    =  177570       332#     333      402      653      667      687      692      753      804      881      892      946      957
                      1018     1032     1086     1100     1147     1161     1295     1306     1506
SW10   =  002000       349#     657
SW11   =  004000       348#
SW12   =  010000       347#
SW13   =  020000       346#     804      881      892      946      957     1018     1032     1086     1100     1147     1161     1295
                      1306     1506
SW14   =  040000       345#     653
SW8    =  000400       351#     687      692
SW9    =  001000       350#     402
S.TC      006666       625     1185#
TAB1      011512       479     1599#
TCCA      011664      1132     1141     1639#
TCCS      011656       527#     909#    1126#    1132     1141     1144     1149     1157#    1158     1163     1195#    1198     1210#
                      1213     1228#    1231     1243#    1246     1636#
TCDB      011666      1205     1220     1238     1253     1640#
TCER      011660      1154     1200     1215     1223     1248     1637#
TCTRAP    011560       384#     391#     491#    1129#    1138#    1190#    1209#    1223#    1242#    1610#
TCWC      011662      1130#    1139#    1638#
TC.MI     011265      1150     1173     1568#
TC.TA     007124      1186#    1187     1189#    1191     1220     1224     1253     1258#
TC.WC     007126      1259#
TMBC      011674      1264#    1267#    1284#    1285#    1643#
TMCA      011676      1269     1287     1644#
TMCS      011670       523#    1269     1275#    1287     1292     1297     1308     1641#
TMER      011672      1302     1642#
TMRET     007264      1273#    1278     1290#
TMTRAP    011570       385#     392#     492#    1266#    1272#    1283#    1612#
TM.MI     011306      1290     1310     1571#
TP.MP     006176       983#    1013
```

```
TRAPS    002716       378     605#
TRP.OM   007516       533    1333#    1438     1463
TRP.DR   010514       393     493     1470     1484#
TRP.RC   004614       388     488      918#
TRP.RF   004232       387     487      853#
TRP.RK   006624       390     490     1058#
TRP.RP   005554       389     489     1007#
TRP.TC   006230       391     491     1127#
TRP.TM   007136       392     492     1263#
TTY     =%000006      341#    457#     695#     696      698#     758#     760#     761#     762#     763#     764#     767      770#
                      771#    772#     775#     776#     780      811#     814#     817#     822#     826#     827#     828      830
                      833#    834#     835#     836#     837#     838      839#     840      883#     888#     906#     909#     912#
                      940#    953#     971#     974#     977#    1020#    1025#    1046#    1049#    1052#    1088#    1093#    1114#
                     1117#   1120#    1149#    1154#    1175#    1178#    1181#    1297#    1302#    1320#    1323#    1326#    1334#
                     1336#   1336#    1337#    1338#    1339#    1340     1384#    1385#    1386#    1387     1401#    1508#    1513#
                     1717    1718#    1719     1721     1722#    1723     1725     1733#    1734#    1735     1736#    1781#    1782#
                     1783#   1785#    1787#    1793#    1795#    1797#
TUNTAB   010162      1345#   1346#    1347     1419#
TYPE   = 000004       330#    456      459      463      465      484      485      497      501      540      553      572      593
                      594     595      596      641      659      689      694      810      813      816      884      885      887
                      904     905      908      911      949      950      952      969      970      973      976     1021     1022
                     1024    1044     1045     1048     1051     1089     1090     1092     1112     1113     1116     1119     1150
                     1151    1153     1173     1174     1177     1180     1298     1299     1301     1318     1319     1322     1325
                     1399    1400     1428     1429     1435     1436     1509     1510     1512     1751     1754     1758     1811
                     1839    1853
TYPEIT   001560       427     437      450#
T0       002142       469     499#     542      591
T1       002222       506     509#
T1XOR8   003450       723#
T2       002242       510     513#
T3       002262       514     517#
T2XOR9   0035[illegible]       720     745#
T3X9     003522       742#    751
T4       0023[illegible]       518     521#
T5       002322       522     525#
T6       002342       526     529#
T7       002370       530     534#
T8       002410       535     538#
T8XOR1   003604       721     735#
WAIT     002430       508     512      516      520      524      528      537      544#    1393
WANT     007122      1191#   1192#    1205     1224#    1225#    1238     1257#
WITHMX   011454       465    1592#
WITHSG   011441       463    1590#
WRADR    003716       616     788#
WRCADR   003754       617     798#
WRDCNT   011744       855     863      872      920      928      937      985      993     1002     1060     1068     1077     1130
                     1139    1266     1284     1496     1665#
W.RC     004610       916#    936
W.RF     004226       851#    871
W.RK     006620      1056#   1076
W.RP     005172       981#   1001
W.TC     006216      1124#   1138
W.TM     007132      1261#   1283
X.TM     007220      1272    1278#
.      = 013064       353#    354      355#     359#     372      456      457#     459      460#     484      486#     497      501
                      596     610      613      637      689      690#     694      705      712      754      805     1009     1011
```

```
                         1030    1098    1159    1201    1216    1224    1248    1429    1430#   1436    1437#   1474    1478
                         1594#   1727    1745    1751    1761#   1771    1800    1803    1808    1814#   1815    1829    1843
                         1851    1852    1858#   1859#
.BEGIN   003552          631     758#
.LOD     012102          1692#   1693    1708#
.PR      012566          1766#   1770#   1772#   1773#   1775    1779    1789#   1791    1801#   1802    1806#   1807    1811
                         1814#
.PRF     012506          1777    1792    1797#
.PRL     012464          1790#   1805
.PTIT    012410          1769    1774#
.SAVR6   012744          1822#   1827    1845#
.START*  104436          631#    857     865     874     908     920     929     987     995     1004    1062    1070    1079
                         1131    1140    1204    1206    1407
.TYPE    012210          1721#   1722    1730#   1753#   1754
```

C05


```
BITYPE       18    3448
BLKBLK       18
BUFHDR       18
DECML        18    3448
DEVCHK     3708    877    942    1014   1082   1143   1291
DUMP         18    3448   814    817    888    953    1025   1093   1154   1302   1401   1513
DUMP18       18    2718   811
FILBLK       18
GO         3708    857    865    874    922    930    939    987    995    1004   1062   1070   1079   1131   1140
           1258    1286   1497
LNKBLK       18
ODT11        18
ODT11X       18
POP          18    3448   647    1831
PRINT        18    3448   456    459    484    496    501    596    689    694    1429   1436   1750   1839   1853
PUSH         18    3448   643    1817
SCOPE.       18
SCOP         18
SCOUMP       18    3448   457    906    971    1046   1114   1175   1320
SET        5908    616    617    618    619    620    621    622    623    624    625    626    627    628    629
            630    631    632    633    634    635    636
TRACE        18
TRNBLK       18
TYPEM        18
WRITE        18
SCATCH       18    2718   353
SCMTAG       18    2718   359
SDECML       18
SEND         18    2718
SEQUAT       18    2718   328
SHLT         18    2718
SOCRAT       18
SLOADR       18    2718   1666
SOCTAL       18    2718   1763
SPOWER       18    2718   1815
SRAND        18
SRANDM       18
SREAD        18    2718   1740
SSCOPE       18    2718
SSET       5908    616    617    618    619    620    621    622    623    624    625    626    627    628    629
            630    631    632    633    634    635    636
SSETUP       18
SSRAT        18
SSWDOC       18    2718
STRAP        18    2718   598
STYPE        18    2718   1709
SUNAT        18
.SCOP        18
.SCOPE       18
```

ADC	549														
ADD	[illegible]	443	472	[illegible]	665	726	728	742	760	777	782	[illegible]	1186	1192	1342
	1344	[illegible]	[illegible]	[illegible]	1376	1378	[illegible]	1476	1480	1688	1730	1733			
ASL	447	1257	[illegible]	[illegible]	1781	1782	1783								
ASR	[illegible]														
ASRB	1784	1786													
BCC	720														
BEQ	451	459	480	[illegible]	547	[illegible]	577	664	673	682	684	688	693	697	705
	712	[illegible]	[illegible]	[illegible]	[illegible]	1043	1111	1172	1317	1398	1434	1448	1474	1478	1504
	1724	1743	1800	[illegible]	1851										
BGT	[illegible]	[illegible]	721	1229											
BIC	[illegible]	[illegible]	734	772	837	901	966	1041	1109	1170	1315	1339	1386	1404	1407
	1462	1486	1734												
BICB	1486	1747	[illegible]												
BIS	[illegible]	807	511	515	519	523	527	536	775	776	1288				
BISB	1788	1804													
BIT	408	[illegible]	567	607	[illegible]	[illegible]	804	881	892	946	957	1018	1032	1086	1100
	1147	1161	[illegible]	[illegible]	1397	1443	1506	1719	1791						
BLE	[illegible]	[illegible]	[illegible]	1704											
BLT	[illegible]	[illegible]	1206												
BMI	417	[illegible]	610	672	720	825	1201	1216	1234	1249	1427				
BNE	[illegible]	418	[illegible]	[illegible]	[illegible]	474	[illegible]	510	514	518	522	526	530	535	650
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	716	774	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1019
	[illegible]	1087	1101	1148	[illegible]	1221	1254	[illegible]	1307	[illegible]	[illegible]	1366	1369	1444	1453
	1461	1470	[illegible]	1493	[illegible]	[illegible]	1702	1720	1732	1749	1756	1790	1806	1808	1830
BPL	[illegible]	784	[illegible]	[illegible]	1009	1011	1016	1030	1084	1098	1145	1159	1188	1199	1214
	[illegible]	1247	[illegible]	1406	1442	1502	[illegible]	1727	1745						
BR	407	427	437	[illegible]	476	494	[illegible]	512	516	520	524	528	537	542	550
	[illegible]	[illegible]	613	675	677	743	751	1013	1203	1207	1218	1236	1240	1251	1431
	1467	[illegible]	[illegible]	1769	1771	1777	1843								
CLC	[illegible]	[illegible]	[illegible]												
CLR	[illegible]	[illegible]	[illegible]	401	421	470	483	500	541	552	555	574	648	652	674
	[illegible]	707	[illegible]	752	[illegible]	790	[illegible]	[illegible]	873	877	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	[illegible]	1003	1012	1014	1061	1067	1078	1082	1143	[illegible]	[illegible]	1363	1366	1370	1371
	1472	1473	[illegible]	1451	1486	1500	1677	1697	1772	1828	1864				
CLRB	1757	1776	1790	1810											
CMP	[illegible]	434	444	505	509	513	517	521	525	529	533	546	587	599	609
	679	711	715	[illegible]	773	781	794	801	807	[illegible]	[illegible]	1238	1253	1438	1452
	1462	1473	1477	[illegible]	1492	1679	1691	1701	1731	1742	1807				
CMPB	479	[illegible]	902	[illegible]	1042	1110	1171	1316	1447	1748	1755	1850			
COM	731	732	[illegible]	1409											
COMB	[illegible]	411	[illegible]												
DEC	[illegible]	570	660	1364	1368	1364									
EMT	[illegible]														
HALT	[illegible]	612	735	1025	1042										
INC	[illegible]	431	440	453	649	662	690	880	945	1017	1086	1146	1294	1481	1505
	1706	[illegible]													
INCB	[illegible]	951	1036	1104	1165	1310	1368	1450	1801	1806					
IOT	[illegible]														
JMP	[illegible]	438	533	591	597	611	[illegible]	1222	1255	1278	1393	1439	1463	1840	1855
JSR	[illegible]	468	567	[illegible]	[illegible]	[illegible]	673	640	[illegible]	668	671	710	792	812	815
	818	[illegible]	[illegible]	907	910	913	951	[illegible]	972	975	978	1023	1026	1047	1060
	[illegible]	1051	1094	1115	1118	1121	1152	1155	1176	1179	1182	1300	1303	1321	1324
	1327	1408	1511	1514											
MOV	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387
	388	389	390	391	392	393	394	395	396	399	404	409	412	413	414

	419	420	429	430	432	436	441	446	451	452	457	466	471	477	481
	[illegible]	[illegible]	487	[illegible]	[illegible]	490	491	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	544	545	551
	[illegible]	571	573	[illegible]	[illegible]	[illegible]	617	[illegible]	[illegible]	670	[illegible]	[illegible]	701	709	723
	735	736	746	747	750	755	756	757	770	778	779	780	791	793	799
	809	811	814	817	822	828	830	838	840	844	845	846	847	848	849
	851	[illegible]	862	863	871	872	[illegible]	[illegible]	[illegible]	919	[illegible]	927	[illegible]	[illegible]	937
	[illegible]	[illegible]	971	[illegible]	[illegible]	[illegible]	[illegible]	1001	1002	1007	1020	1025	1028	1046	1059
	1060	1067	1068	1076	1077	1080	1093	[illegible]	1114	[illegible]	[illegible]	1130	1138	[illegible]	1149
	1154	1157	1175	[illegible]	1190	1191	[illegible]	[illegible]	1209	1210	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
	1243	[illegible]	[illegible]	[illegible]	1273	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1334	1340	1341	1343
	[illegible]	1347	[illegible]	[illegible]	1351	[illegible]	[illegible]	[illegible]	1357	[illegible]	1374	1375	1377	1379	1380
	1381	[illegible]	[illegible]	[illegible]	1387	[illegible]	[illegible]	1391	1401	[illegible]	[illegible]	1437	1446	1454	[illegible]
	[illegible]	1471	1472	[illegible]	1487	[illegible]	1490	1494	[illegible]	[illegible]	1509	1513	1678	1681	1687
	[illegible]	[illegible]	[illegible]	[illegible]	1700	1717	1718	1721	1722	[illegible]	[illegible]	[illegible]	1740	1741	[illegible]
	[illegible]	1774	1775	1778	1779	[illegible]	1812	1815	1816	1817	1818	1819	1820	1821	1822
	1823	1824	1827	1831	[illegible]	[illegible]	1834	1835	1836	1837	1838				
MOVB	[illegible]	894	895	909	912	959	960	974	977	1034	1035	1049	1052	1102	1103
	1117	1120	1163	1164	1178	1181	1308	1309	1323	1326	1367	1725	1746	1753	1770
	1773	1780	1809	1848											
NEG	[illegible]	1431													
RESET	320	[illegible]													
ROL	725	738	739	740	741	771	1793	1795	1797						
ROLB	[illegible]	[illegible]	584	[illegible]	[illegible]	581	582	583	584	585	1794	1796	1798		
ROR	727	[illegible]	746	749	750	760	761	762	763	833	834	835	836	898	899
	900	[illegible]	964	965	1038	1039	1040	1106	1107	1108	1167	1168	1169	1312	1313
	1314	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]				
RORB	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1105	1166	1311			
RTI	702	706	717	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	831	841	850	859	867
	876	891	915	[illegible]	[illegible]	941	[illegible]	[illegible]	[illegible]	997	1006	1031	1055	1064	1072
	1081	[illegible]	1123	1133	1142	1160	[illegible]	1196	1211	1229	1244	1270	1276	[illegible]	1305
	[illegible]	1408	[illegible]	[illegible]	1515	1737	1852								
RTS	651	729	[illegible]	1706	1760	1813									
SUB	478	[illegible]	[illegible]												
SWAB	448	678	724	737	748	[illegible]									
TRAP	[illegible]	616	617	618	619	620	621	622	623	624	625	626	627	628	629
	[illegible]	631	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]								
TST	406	410	416	[illegible]	433	[illegible]	460	468	473	531	568	666	671	681	[illegible]
	704	714	719	753	[illegible]	878	943	1010	1015	1083	1144	1187	1194	1198	1200
	1213	1215	1227	1231	[illegible]	[illegible]	1248	1274	1292	1406	1426	1433	1440	1441	1460
	[illegible]	1501	1503	1675	[illegible]	1703									
TSTB	271	[illegible]	567	576	1008	1029	1097	1158	1723	1726	1744	1799	1802		
.ASCII	1519	[illegible]													
.ASCIZ	457	460	485	498	502	597	680	685	1430	1437	1528	1530	1532	1534	1537
	[illegible]	1542	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1564	1568	1571	1574	1577	1580	1583
	[illegible]	[illegible]	[illegible]	[illegible]	1592	1752	1840	1854							
.BLKW	1761	1814	1858	1869											
.ENABL	1	271													
.END	1861														
.ENDC	344	804	891	949	956	1021	1031	1089	1099	1150	1160	1298	1305	1679	1699
	1705	1723	1738	1743	[illegible]	1790	1815	1823	1831	1840	1841				
.EVEN	457	460	485	498	502	597	680	685	1430	1437	1594	1752	1840	1854	
.IF	344	804	891	949	956	1021	1028	1089	1096	1150	1157	1298	1305	1675	1695
	1703	1719	[illegible]	1742	1761	[illegible]	1778	1815	1823	1831	1839	1840			
.IFF	[illegible]	[illegible]	1738	1743	1762	1840									
.IIF	1761	1815													
.INP	643	647	1817	1831											

F05


```
.LIST        1     271     344     354     370     457     460     485     498     499     502     597     598     616     617
           618     619     620     621     622     623     624     625     626     627     628     629     630     631     632
           633     634     635     636     637     652     690     695     851     877     916     942     981    1014    1056
          1082    1124    1143    1261    1291    1330    1430    1437    1467    1500    1615    1666    1709    1740    1752
          1762    1815    1840    1854
.MACRO       1     370     598
.MCALL     271     344
.NLIST       1     271     344     354     370     457     460     485     498     499     502     597     598     616     617
           618     619     620     621     622     623     624     625     626     627     628     629     630     631     632
           633     634     635     636     637     652     690     695     851     877     916     942     981    1014    1056
          1082    1124    1143    1261    1291    1330    1430    1437    1467    1500    1615    1666    1709    1740    1752
          1762    1815    1840    1854
.PAGE      598     877     942    1014    1082    1143    1291    1709    1762    1815
.REM         1     272
.REPT      354
.SBTTL     370     499     598     652     851     877     916     942     981    1014    1056    1082    1124    1143    1261
          1291    1330    1467    1500    1615    1666    1709    1740    1762    1815
.TITLE     271


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZQMAB.SEQ=DZQMAB.SML,DZQMAB.CMB
RUN-TIME: 17 23 3 SECONDS
RUN-TIME RATIO: 286/44=6.4
CORE USED:  22K  (43 PAGES)
```