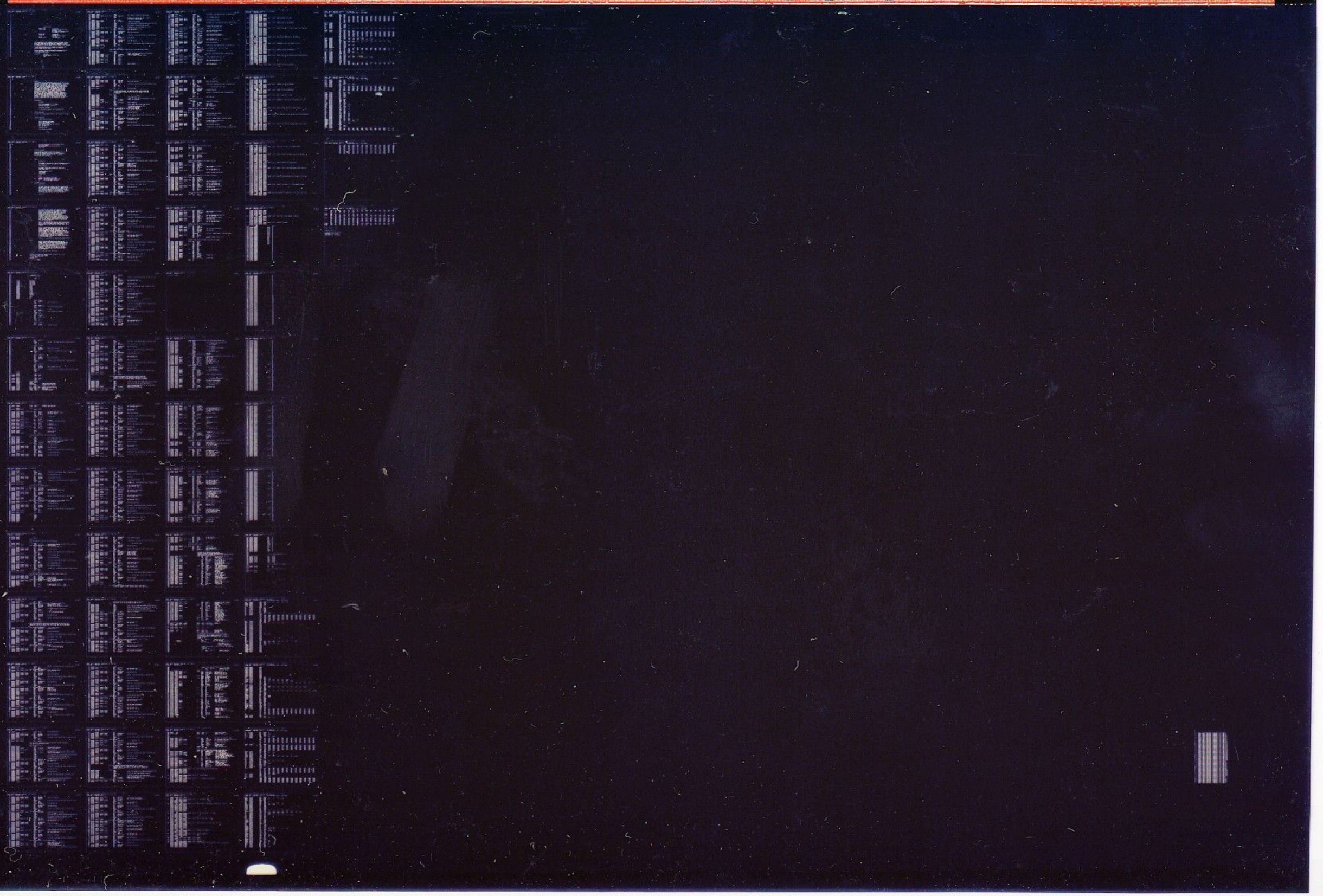
CTS11-JC CTS11-JC 7 BIT ASCII AH-F415A-MC
CZCTAA0 FICHE 1 OF 1



SEQ 0001

```
1                               .REM _
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

IDENTIFICATION

PRODUCT CODE:	AC-F414A-MC
PRODUCT NAME:	CZCTAA0 CTS11-JC 7 BIT ASCII 8035-8045 (11/70) 80 COLUMN CARD TERMINAL CONTROL DATA FORMAT: 7 BIT ASCII
UPDATE DATE	20-MARCH 80
MAINTAINER:	JIM BENNETT COMPUTER SPECIAL SYSTEMS
AUTHOR:	P.W. DUKE
UPDATE AUTHOR:	VIJAY ANANDWALA

```
1.0     ABSTRACT

        THIS PROGRAM TESTS THE OPERATIONAL ABILITY OF THE CTS11-JC TO
        INTERFACE TO AN 8035-8045  80 COLUMN READER/PUNCH.  THE PROGRAM
        CONSISTS OF TWO SETS OF TESTS, LOGIC TESTS AND CARD TESTS.
        THE LOGIC TESTS ARE PERFORMED WITHOUT CARDS IN THE 8035-8045 .
        THE CARD TESTS ARE A FOUR PASS SERIES OF TESTS WHICH USE 20
        CARDS.  THE CARD TESTS SHOULD NOT BE RUN UNTIL THE LOGIC
        TESTS HAVE RUN SUCCESSFULLY.  THE PROGRAM IS DEVICE CODE
        INDEPENDENT AND THERE IS AN INPUT ROUTINE WHICH ALLOWS
        THE OPERATOR TO SPECIFY DEVICE ADDRESS, INTERRUPT VECTOR
        ADDRESS AND BR LEVEL.  THIS ROUTINE IS RUN AT THE
        BEGINNING OF EVERY PASS THROUGH THE PROGRAM.  THERE IS A
        GREAT DEAL OF INTERACTION REQUIRED BETWEEN THE OPERATOR
        AND THE PROGRAM.  THIS INTERACTION IS ACCOMPLISHED BY MEANS
        OF TELETYPE DIRECTIONS AND MESSAGES TYPED OUT DURING THE
        RUNNING OF THE PROGRAM WHICH FREQUENTLY REQUIRES OPERATOR
        RESPONSE.  DIRECTIONS GIVEN TO THE OPERATOR VIA THE TELETYPE
        MUST BE FOLLOWED EXACTLY TO ACHIEVE SUCCESSFUL TESTING.


2.0     REQUIREMENTS

        2.1     EQUIPMENT

                PDP-11 WITH MINIMUM OF 4K OF MEMORY
                CTS11-JC INTERFACE
                8035-8045  80 COLUMN READER/PUNCH

        2.2     PRELIMINARY PROGRAMS

                ALL PROCESSOR DIAGNOSTICS MUST RUN ERROR FREE


3.0     LOADING PROCEDURE

        PROCEDURE FOR LOADING NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.


4.0     STARTING PROCEDURE

        4.1     CONTROL SWITCH SETTINGS

                SR15=1 CONTINUE AFTER ERROR
                SR14=1 DELETE ERROR TYPEOUTS
                SR13=1 LOOP ON ERROR
                SR2=1 BYPASS DEVICE CODE ENTRY ROUTINE
                SR1=1 DO LOGIC TESTS
                SR0=1 DO CARD TESTS

        4.2     OPERATOR ACTION

                LOAD PROGRAM INTO MEMORY
```

```
                                SET SWITCH REGISTER TO STARTING ADDRESS 200(8)
                                PRESS LOAD ADDRESS
                                SET SWITCH REGISTER TO DESIRED OPTIONS
                                PRESS START


        5.0     OPERATING PROCEDURE

                FOLLOW DIRECTIONS GIVEN ON THE TELETYPE EXACTLY.  THIS IS
                NECESSARY FOR SUCCESSFUL TESTING.  AT THE END OF A
                COMPLETE PASS THROUGH THE SELECTED ROUTINES THE PROGRAM
                WILL TYPE OUT ''END OF TESTING'' AND HALT.  TO REPEAT
                FOR ONE PASS, PRESS CONTINUE.


        6.0     ERRORS

                6.1     ERROR HALTS

                        THE MEANING OF AN ERROR HALT CAN BE DETERMINED BY READING
                        THE PROGRAM LISTING AT THE ADDRESS OF THE ERROR HALT.

                6.2     ERROR TYPEOUTS

                        ALL ERROR TYPEOUTS WILL INCLUDE THE FIRST LINE
                        SHOWN BELOW, AND WHEN RELEVANT ONE OR MORE OF THE
                        OTHER THREE LINES.

                        ''ERROR AT ADDRESS AAAAAA
                        ''GOOD=BBBBBB
                        ''BAD =CCCCCC
                        ''COLUMN #=DD''

                        WHERE:

                        AAAAAA = THE ADDRESS OF THE ERROR
                        BBBBBB = THE EXPECTED RESULT FROM A TEST
                        CCCCCC = THE ACTUAL RESULT FROM A TEST
                        DD     = THE COLUMN NUMBER IN ERROR


        7.0     PROGRAM DESCRIPTION

                7.1     LOGIC TESTS

                        THE LOGIC TESTS ARE A SERIES OF 6 TESTSS WHICH DO NOT
                        REQUIRE CARDS TO BE IN THE 8035-8045 .  THESE 6 TESTS PERFORM
                        BASIC CHECKS ON THE 4 REGISTERS ON THE INTERFACE AND
                        TESTS AS MUCH AS POSSIBLE OF THE OPERATIONAL CAPABILITY
                        OF THE INTERFACE WHILE IN A STATIC CONDITION.  CORRECT
                        RESPONSE BY THE OPERATOR TO INSTRUCTIONS GIVEN VIA TYPE
                        OUTS IS NECESSARY FOR THIS TEST TO BE SUCCESSFUL.

                7.2     CARD TESTS
```

```
                        THE CARD TESTS ARE A SERIES OF 4 ROUTINES (PASSES)
                        WHICH CHECK THE ABILITY OF THE INTERFACE TO PERFORM
                        CORRECTLY IN CARD HANDLING OPERATIONS.  THE TESTS
                        REQUIRE A DECK OF 20 BLANK CARDS WHICH ARE PASSED
                        THROUGH THE 8035-8045  4 TIMES, WHICH AT THE
                        CONCLUSION OF THE 4 PASSES CONTAIN PUNCHED
                        INFORMATION WHICH MUST BE VISUALLY VERIFIED BY THE
                        OPERATOR.  VERIFICATION OF CORRECT OPERATION OF THESE
                        4 PASSES IS DONE BY THE OPERATOR BASED ON INSTRUCTIONS
                        GIVEN VIA TYPE OUTS AT THE BEGINNING AND END OF EACH
                        PASS.  DETAILED DESCRIPTIONS OF WHAT IS DONE DURING
                        EACH PASS IS GIVEN BELOW.

                        PASS 1 - THE OPERATOR PUTS 10 BLANK CARDS IN EACH INPUT
                        HOPPER.  THE PROGRAM READS THESE CARDS AND PUTS THEM INTO
                        THE 2 STACKERS, 10 BLANK CARDS IN EACH STACKER.  THE
                        OPERATOR VERIFIES THAT THERE ARE 10 BLANK CARDS IN EACH
                        STACKER.

                        PASS 2 - THE OPERATOR PUTS THE 20 BLANK CARDS IN THE
                        PRIMARY HOPPER.  THE PROGRAM READS THESE CARDS AND ALSO
                        PUNCHES THE CHARACTER SET ON EACH CARD.  THE
                        CARDS ARE PUNCHED IN A PRECESS FASHION WITH THE 1ST
                        PUNCHED CHARACTER OF CARD #1 APPEARING IN COLUMN #11
                        AND SO ON UNTIL THE 1ST PUCNCHED CHARACTER OF CARD #20
                        IS IN COLUMN #31.  THE CARDS ARE ALL PLACED IN STACKER
                        #1, THE OPERATOR VERIFIES THAT THE CHARACTER
                        SET APPEARS IN EACH CARD AND THAT PUNCHING WAS DONE IN
                        A PRECESS FASHION.  THE CHARACTERS APPEAR ON THE CARDS
                        IN THE ORDER SHOWN BELOW:

                        '"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ_

.REM _                  PASS 3 - THE OPERATOR PUTS THE 20 CARDS IN THE
                        PRIMARY HOPPER.  THE PROGRAM READS THE CARDS AND ALSO
                        PUNCHES MORE CHARACTERS ON EACH CARD BEGINNING WITH
                        COLUMN #1, CARD #1 WILL HAVE 1 CHARACTER PUNCHED ON IT,
                        CARD #2 WILL HAVE 2 CHARACTERS UP TO CARD #20 WHICH
                        WILL HAVE 20 CHARACTERS PUNCHED ON IT.  ALL THE
                        CHARACTERS PUNCHED IN THIS TEST WILL BE THE "A"
                        CHARACTER.  THE CARDS WILL BE PLACED IN STACKER #1,
                        THE OPERATOR VERIFIES THAT THESE CHARACTERS WERE
                        PUNCHED CORRECTLY

                        THIS IS THE END OF THE CARD TESTS.

-

;
;CTS11-JC 80 COLUMN CARD TERMINAL CONTROL INTERFACE TEST (11/70)
;DATA IS IN  (7 BIT) ASCII FORMAT
;   ***** MODEL 8035-8045  *****
;DECSPEC-AONCDA
;AUTHORS:
;       P. W. DUKE
;       A. L. UNSER
```

```
                        ;19-DEC-73
                        ;[UPDATED-MARCH-80]
                        ;
                                .LIST   ME
                                .ABS

177776                  PS=177776
177570                  SR=177570
177560                  TKS=177560
177562                  TKB=177562
177564                  TPS=177564
177566                  TPB=177566
000000                  R0=%0
000001                  R1=%1
000002                  R2=%2
000003                  R3=%3
000004                  R4=%4
000005                  R5=%5
000006                  SP=%6
000007                  PC=%7
000000                  XX=HALT
104000                  ERR=EMT
104400                  TYPE=TRAP
000004                  COL=4
000003                  GB=3
000002                  G=2
000001                  B=1


                        .MACR   WTDONE
                        BIT     #200,@TCR
                        BEQ     .-6             ;WAIT FOR DONE FLAG
                        .ENDM

                        .MACR   WTODR
                        BITB    #200,@TIRHI
                        BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
                        .ENDM

                        .MACR   WTIDR
                        BIT     #200,@TIR
                        BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
                        .ENDM
                        .MACR   LOOP A
                        .NLIST
                        CMP     #1,SCOPE
                        BNE     .+16
                        CMP     #.-10,LASTPC
                        BNE     .+6
                        .LIST
                        JMP     @#A
                        .ENDM

                        .MACR   CKTCR A,B
                        LP=.
                        MOV     A,GOOD
                        MOV     @TCR,BAD        ;READ CMD REGISTER
```

```
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    B
                                        .ENDM

                                        .MACR   CKTSR A,B
                                        LP=.
                                        MOV     A,GOOD
                                        MOV     @TSR,BAD        ;READ STATUS REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    B
                                        .ENDM

                                        .MACR   CKTDR A,B
                                        LP=.
                                        MOV     A,GOOD
                                        MOV     @TDR,BAD        ;READ DATA REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;DATA REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    B
                                        .ENDM

                                        .MACR   CKTIR A,B
                                        LP=.
                                        MOV     A,GOOD
                                        MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    B
                                        .ENDM
            000000                      .=0
            000030                      .=30
000030      013126                      .WORD   ERROR
000032      000340                      .WORD   340
            000034                      .=34
000034      013312                      .WORD   TYPOUT
000036      000340                      .WORD   340
            000174                      .=174
000174      000000          DISPREG:.WORD 0
000176      000000          SWREG:  .WORD 0
            000004          ERRVEC=4                ;TIME OUT AND OTHER ERROR
            177570          DSWR=177570             ;HARDWARE SWITCH REGISTER
            177570          DDISP=177570            ;HARDWARE DISPLAY REGISTER
            000500                      .=500
000500      177570          SWR:    .WORD   DSWR  ;;ADDRESS OF SWITCH REGISTER
000502      177570          DISPLAY:.WORD   DDISP  ;ADDRESS OF DISPLAY REGISTER
000504      000000          APASS:  .WORD   0     ;PASS COUNT
000506      000000          AENVM:  .WORD   0     ;ENVIRONMENT REGISTER
000510      000000          ASWREG: .WORD   0     ;APT SWITCH REGISTER
```

```
                                AUTOB:  .WORD   0       ;AUTOMATIC MODE INDICATOR
                                INTAG:  .WORD   0       ;INTERRUPT MODE INDICATOR

                                .=200
000200  000167  000574                  JMP     START
                001000                  .=1000

001000  012767  000340  176770  START:  MOV     #340,PS         ;SET BPU TO LEVEL #7
001006  012706  001000                  MOV     #1000,SP        ;INITIALIZE THE STACK POINTER
001012  005067  017514                  CLR     SCOPE           ;INITIALIZE SCOPE
001016  004767  013520                  JSR     PC,SWRCK
001022  104400                          TYPE
001024  017303                          PROGNM                   ;''CZCTAA.SRC''
001026  004767  012722                  JSR     PC,GTSWR

001032  032777  000004  177440          BIT     #4,@SWR
001040  001002                          BNE     ST1             ;IF SR2=0
001042  004767  011644                  JSR     PC,DEVCOD       ;GO TO DEVICE CODE ENTRY ROUTINE

001046  032777  000002  177424  ST1:    BIT     #2,@SWR
001054  001402                          BEQ     ST2             ;IF @SWR1=1
001056  004767  000026                  JSR     PC,INIT         ;GO DO LOGIC TESTS

001062  032777  000001  177410  ST2:    BIT     #1,@SWR
001070  001402                          BEQ     ST3             ;IF SR0=1
001072  004767  003144                  JSR     PC,PASS1        ;GO DO CARD TESTS

001076  104400                  ST3:    TYPE
001100  017442                          ENDMSG                  ;''END OF TESTING''
001102  000000                          XX                      ;HALT AT END
001104  000137  001046                  JMP     @#ST1
                                ;
                                ;3 REGISTERS SHOULD EQUAL ZERO AT INITIALIZATION
                                ;
001110  104400                  INIT:   TYPE
001112  015177                          LOGTST                  ;''***** LOGIC TESTS *****''
001114  104400                          TYPE
001116  015233                          INITM1                  ;''PUT THE 8035-8045  OFF-LINE, REMOVE ALL CARDS
001120  104400                          TYPE
001122  017001                          P2MSG4                  ;PRESS STOP-RESET ON 8035-8045
001124  004767  010656                  JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
001130  000005                  INIT1:  RESET
001132                                  CKTCR   #0,INIT1        ;TEST CMD REG FOR 0
(1)             001132                  LP=.
(1)     001132  012767  000000  017350  MOV     #0,GOOD
(1)     001140  017767  017374  017332  MOV     @TCR,BAD        ;READ CMD REGISTER
(1)     001146  026767  017336  017324  CMP     GOOD,BAD
(1)     001154  001401                  BEQ     .+4
(1)     001156  104003                  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)     001160                          LOOP    INIT1
(2)     001200  000137  001130          JMP     @#INIT1
001204                                  CKTSR   #0,INIT1        ;TEST STATUS REG FOR 0
(1)             001204                  LP=.
(1)     001204  012767  000000  017276  MOV     #0,GOOD
(1)     001212  017767  017326  017260  MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)     001220  026767  017264  017252  CMP     GOOD,BAD
```

```
   (1)  001226  001401                          BEQ     .+4
   (1)  001230  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  001232                                  LOOP    INIT1
   (2)  001252  000137  001130                  JMP     @#INIT1
   380  001256                                  CKTIR   #0,INIT1        ;TEST INTERRUPT REG FOR 0
   (1)          001256                          LP=.
   (1)  001256  012767  000000  017224          MOV     #0,GOOD
   (1)  001264  017767  017264  017206          MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
   (1)  001272  026767  017212  017200          CMP     GOOD,BAD
   (1)  001300  001401                          BEQ     .+4
   (1)  001302  104003                          ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  001304                                  LOOP    INIT1
   (2)  001324  000137  001130                  JMP     @#INIT1
   381
   382  001330  000137  001334                  JMP     @#TEST1
   383                                  ;
   384                                  ;LOAD-READ TEST OF COMMAND REGISTER
   385                                  ;
   386  001334  012702  001520          TEST1:  MOV     #T1LST,R2
   387  001340  012267  017146          T1A:    MOV     (R2)+,HOLD
   388  001344  016777  017142  017166  T1B:    MOV     HOLD,@TCR       ;LOAD CMD REGISTER
   389  001352                                  CKTCR   HOLD,T1B        ;TEST CMD REG FOR CORRECT VALUE
   (1)          001352                          LP=.
   (1)  001352  016767  017134  017130          MOV     HOLD,GOOD
   (1)  001360  017767  017154  017112          MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  001366  026767  017116  017104          CMP     GOOD,BAD
   (1)  001374  001401                          BEQ     .+4
   (1)  001376  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  001400                                  LOOP    T1B
   (2)  001420  000137  001344                  JMP     @#T1B
   390  001424  020227  001552                  CMP     R2,#T1END+2
   391  001430  001343                          BNE     T1A
   392                                  ;NOW SEE IF RESET CLEARS ALL BITS THAT CAN BE SET IN THE COMMAND REGISTER
   393  001432  012777  074136  017100  T1C:    MOV     #74136,@TCR     ;LOAD COMMAND REG WITH ALL BITS
   394  001440  000005                          RESET
   395  001442                                  CKTCR   #0,T1C          ;TEST CMD REG FOR 0
   (1)          001442                          LP=.
   (1)  001442  012767  000000  017040          MOV     #0,GOOD
   (1)  001450  017767  017064  017022          MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  001456  026767  017026  017014          CMP     GOOD,BAD
   (1)  001464  001401                          BEQ     .+4
   (1)  001466  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  001470                                  LOOP    T1C
   (2)  001510  000137  001432                  JMP     @#T1C
   396
   397  001514  000137  001552                  JMP     @#TEST2
   398
   399  001520  000000                  T1LST:  0
   400  001522  040000                          40000
   401  001524  020000                          20000
   402  001526  010000                          10000
   403  001530  004000                          4000
   404  001532  000100                          100
   405  001534  000020                          20
   406  001536  000010                          10
   407  001540  000004                          4
```

```
     001542  000002                             2
     001544  000000                             0
     001546  074136                             74136
     001550  000000                     T1END:  0
                                        ;
                                        ;TEST THAT DONE BIT IS ONLY BIT SET IN 3 REGISTERS
                                        ;
     001552  104400                     TEST2:  TYPE
     001554  015412                             T2MSG1                      ;"PUT 8035-8045  ON-LINE AND PRESS START ON THE 8035"
     001556  004767  010224                     JSR     PC,CONTIN           ;"PRESS CR TO CONTINUE"
                                        ;DONE SHOULD BE THE ONLY BIT SET IN 3 REGISTERS
     001562                                     CKTCR   #200,LP             ;TEST CMD REG FOR DONE FLAG
(1)          001562                             LP=.
(1)  001562  012767  000200  016720             MOV     #200,GOOD
(1)  001570  017767  016744  016702             MOV     @TCR,BAD            ;READ CMD REGISTER
(1)  001576  026767  016706  016674             CMP     GOOD,BAD
(1)  001604  001401                             BEQ     .+4
(1)  001606  104003                             ERR+GB                      ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  001610                                     LOOP    LP
(2)  001630  000137  001562                     JMP     @#LP
     001634                                     CKTSR   #0,LP               ;TEST STATUS REG FOR 0
(1)          001634                             LP=.
(1)  001634  012767  000000  016646             MOV     #0,GOOD
(1)  001642  017767  016676  016630             MOV     @TSR,BAD            ;READ STATUS REGISTER
(1)  001650  026767  016634  016622             CMP     GOOD,BAD
(1)  001656  001401                             BEQ     .+4
(1)  001660  104003                             ERR+GB                      ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  001662                                     LOOP    LP
(2)  001702  000137  001634                     JMP     @#LP
     001706                                     CKTIR   #0,LP               ;TEST INTERRUPT REG FOR 0
(1)          001706                             LP=.
(1)  001706  012767  000000  016574             MOV     #0,GOOD
(1)  001714  017767  016634  016556             MOV     @TIR,BAD            ;READ INTERRUPT REGISTER
(1)  001722  026767  016562  016550             CMP     GOOD,BAD
(1)  001730  001401                             BEQ     .+4
(1)  001732  104003                             ERR+GB                      ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  001734                                     LOOP    LP
(2)  001754  000137  001706                     JMP     @#LP

     001760  000137  001764                     JMP     @#TEST3
                                        ;
                                        ;TEST THAT DONE BIT CAUSES INTERRUPT TO CORRECT ADDRESS AND BR LEVEL
                                        ;
     001764  012767  000340  176004     TEST3:  MOV     #340,PS
     001772  012767  000007  016520             MOV     #7,LEVEL
     002000  012777  002072  016552             MOV     #T3INT,@PCV         ;SETUP PC VECTOR
     002006  012777  000340  016546             MOV     #340,@PSV           ;SETUP PS VECTOR
                                        ;SHOULD GET AN INTERRUPT WHEN BPU IS SET TO ONE LEVEL BELOW BR LEVEL FOR DEVICE
     002014  012777  000100  016516             MOV     #100,@TCR           ;ENABLE INTERRUPT IN CMD REG
     002022  162767  000040  175746     T3A:    SUB     #40,PS              ;DROP BPU LEVEL BY 1
     002030  000240                             NOP
     002032  000240                             NOP
     002034  005367  016460                     DEC     LEVEL               ;IF LEVEL IS 0 OR HIGHER
     002040  100370                             BPL     T3A                 ;GO WAIT FOR INT
     002042  104000                             ERR                         ;GOT NO INTERRUPT WITH BPU AT LEVEL 0
     002044                                     LOOP    TEST3
```

```
   (1)  002064  000137  001764                    JMP     @#TEST3
   440  002070  000427                            BR      T3B
   441                                    ;GOT INTERRUPT, NOW TEST TO SEE THAT IT OCCURRED AT CORRECT LEVEL
   442  002072  062706  000004             T3INT: ADD     #4,SP           ;HOUSE KEEPING OF STACK
   443  002076  016767  016462  016404            MOV     BRLV,GOOD       ;GOOD=BR LEVEL OF DEVICE
   444  002104  016767  016410  016366            MOV     LEVEL,BAD       ;BAD=LEVEL AT WHICH INTERRUPT OCCURRED
   445  002112  026767  016372  016360            CMP     GOOD,BAD
   446  002120  001401                            BEQ     .+4
   447  002122  104003                            ERR+GB                  ;INTERRUPT OCCURRED AT WRONG LEVEL
   448  002124                                    LOOP    TEST3
   (1)  002144  000137  001764                    JMP     @#TEST3
   449  002150  000005                     T3B:   RESET                   ;CLEAR OUT CMD REG WITH RESET
   450  002152                                    CKTCR   #200,T3B                ;TEST CMD REG FOR DONE
   (1)          002152                            LP=.
   (1)  002152  012767  000200  016330            MOV     #200,GOOD
   (1)  002160  017767  016354  016312            MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  002166  026767  016316  016304            CMP     GOOD,BAD
   (1)  002174  001401                            BEQ     .+4
   (1)  002176  104003                            ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  002200                                    LOOP    T3B
   (2)  002220  000137  002150                    JMP     @#T3B
   451
   452  002224  000137  002230                    JMP     @#TEST4
   453                                    ;
   454                                    ;TEST THAT HOPPER EMPTY CAN BE DETECTED WHEN HOPPERS ARE SELECTED, THAT
   455                                    ;HOPPER EMPTY BEING SET CAUSES INPUT ERROR TO SET AND THIS IN TURN CAUSES
   456                                    ;ERROR BIT TO SET IN THE CMD REG. TEST THAT ERROR BIT CAUSES AN INTERRUPT.
   457                                    ;
   458                                    ;TEST FOR PRIMARY HOPPER EMPTY
   459  002230  112777  000040  016304     TEST4: MOVB    #40,@TCRHI      ;SELECT PRIMARY HOPPER
   460  002236                                    CKTSR   #220,TEST4      ;TEST FOR IE + HE
   (1)          002236                            LP=.
   (1)  002236  012767  000220  016244            MOV     #220,GOOD
   (1)  002244  017767  016274  016226            MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  002252  026767  016232  016220            CMP     GOOD,BAD
   (1)  002260  001401                            BEQ     .+4
   (1)  002262  104003                            ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  002264                                    LOOP    TEST4
   (2)  002304  000137  002230                    JMP     @#TEST4
   461  002310                                    CKTCR   #120200,TEST4   ;TEST FOR ERR + HSO + DONE
   (1)          002310                            LP=.
   (1)  002310  012767  120200  016172            MOV     #120200,GOOD
   (1)  002316  017767  016216  016154            MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  002324  026767  016160  016146            CMP     GOOD,BAD
   (1)  002332  001401                            BEQ     .+4
   (1)  002334  104003                            ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  002336                                    LOOP    TEST4
   (2)  002356  000137  002230                    JMP     @#TEST4
   462                                    ;TEST THAT RESET WILL CLEAR OUT CMD REG
   463  002362  000005                     T4A:   RESET                   ;CLEAR OUT CMD REG WITH RESET
   464  002364                                    CKTCR   #200,T4A                ;TEST CMD REG FOR DONE
   (1)          002364                            LP=.
   (1)  002364  012767  000200  016116            MOV     #200,GOOD
   (1)  002372  017767  016142  016100            MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  002400  026767  016104  016072            CMP     GOOD,BAD
   (1)  002406  001401                            BEQ     .+4
```

```
(1)  002410  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  002412                                    LOOP    T4A
(2)  002432  000137  002362                    JMP     @#T4A
465                                 ;TEST FOR SECONDARY HOPPER EMPTY
466  002436  112777  000100  016076  T4B:     MOVB    #100,@TCRHI     ;SELECT SECONDARY HOPPER
467  002444                                    CKTSR   #220,T4B        ;TEST FOR IE + HE
(1)          002444                            LP=.
(1)  002444  012767  000220  016036            MOV     #220,GOOD
(1)  002452  017767  016066  016020            MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  002460  026767  016024  016012            CMP     GOOD,BAD
(1)  002466  001401                            BEQ     .+4
(1)  002470  104003                            ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  002472                                    LOOP    T4B
(2)  002512  000137  002436                    JMP     @#T4B
468  002516                                    CKTCR   #140200,T4B     ;TEST FOR ERR + HS1 + DONE
(1)          002516                            LP=.
(1)  002516  012767  140200  015764            MOV     #140200,GOOD
(1)  002524  017767  016010  015746            MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  002532  026767  015752  015740            CMP     GOOD,BAD
(1)  002540  001401                            BEQ     .+4
(1)  002542  104003                            ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  002544                                    LOOP    T4B
(2)  002564  000137  002436                    JMP     @#T4B
469                                 ;TEST THAT ERROR BIT IN CMD REG CAUSES AN INT
470  002570  112777  000100  015742            MOVB    #100,@TCR       ;ENABLE INT
471  002576  012777  002662  015754            MOV     #T4INT,@PCV     ;SETUP PC VECTOR
472  002604  012777  000340  015750            MOV     #340,@PSV       ;SETUP PS VECTOR
473  002612  012767  000040  175156  T4C:     MOV     #40,PS          ;SET BPU TO LEVEL #1
474  002620  000240                            NOP
475  002622  000240                            NOP
476  002624  012767  000340  175144            MOV     #340,PS         ;SET BPU TO LEVEL #7
477  002632  104000                            ERR                     ;WITH ERROR BIT SET IN CMD REG INT DIDN'T OCCUR
478  002634                                    LOOP    T4C
(1)  002654  000137  002612                    JMP     @#T4C
479  002660  000402                            BR      T4D
480  002662  062706  000004          T4INT:    ADD     #4,SP           ;HOUSE KEEPING ON STACK
481  002666                           T4D:     CKTCR   #140300,T4D     ;TEST FOR ERR + HS1 + IE + DONE
(1)          002666                            LP=.
(1)  002666  012767  140300  015614            MOV     #140300,GOOD
(1)  002674  017767  015640  015576            MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  002702  026767  015602  015570            CMP     GOOD,BAD
(1)  002710  001401                            BEQ     .+4
(1)  002712  104003                            ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  002714                                    LOOP    T4D
(2)  002734  000137  002666                    JMP     @#T4D
482                                 ;TEST THAT WRITING A 1 TO ERROR BIT WILL CLEAR IT
483  002740  012777  000400  015572  T4E:     MOV     #400,@TCR       ;CLEAR ERROR BIT BY WRITING A 1 TO IT
484  002746                                    CKTCR   #200,T4E                ;TEST CMD REG FOR DONE
(1)          002746                            LP=.
(1)  002746  012767  000200  015534            MOV     #200,GOOD
(1)  002754  017767  015560  015516            MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  002762  026767  015522  015510            CMP     GOOD,BAD
(1)  002770  001401                            BEQ     .+4
(1)  002772  104003                            ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  002774                                    LOOP    T4E
(2)  003014  000137  002740                    JMP     @#T4E
```

```
                                        CKTSR   #0,T4E          ;TEST STATUS REG FOR 0
(1)     003020                          LP=.
(1)     003020  012767  000000  015462  MOV     #0,GOOD
(1)     003026  017767  015512  015444  MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)     003034  026767  015450  015436  CMP     GOOD,BAD
(1)     003042  001401                  BEQ     .+4
(1)     003044  104003                  ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)     003046                          LOOP    T4E
(2)     003066  000137  002740          JMP     @#T4E

487     003072  000137  003076          JMP     @#TEST5

                                ;
                                ;TEST FOR PROPER STACKER FULL DETECTION FROM BOTH STACKERS INDIVIDUALLY
                                ; AND THEN IN STACKER OVERFLOW MODE
                                ;
                                ;TEST FOR STACKER #1 FULL
                                TEST5:  TYPE
                                        T5M1                    ;'PUT 8035-8045  OFF-LINE
                                        JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE''
                                        RESET
                                        MOVB    #10,@TCRHI      ;SELECT STACKER #1
                                        TYPE
                                        T2MSG1                  ;'PUT 8035-8045  ON-LINE AND PRESS START ON 8035''
                                        JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE''
                                        TYPE
                                        T5MSG1                  ;'PULL CARD PLATE OF STACKER #1 BACK TO POSITION WHICH
                                        TYPE
                                        T5MSG2                  ;WILL SIMULATE STACKER FULL AND RELEASE''
                                        TYPE
                                        T5MSG3                  ;'PROGRAM WILL WAIT FOR STACKER FULL FLAG''
                                T5A:    BITB    #20,@TSRHI
                                        BEQ     T5A             ;WAIT FOR STACKER FULL FLAG
                                        CKTSR   #110000,T5A     ;TEST FOR OE + SF
(1)     003152                          LP=.
(1)     003152  012767  110000  015330  MOV     #110000,GOOD
(1)     003160  017767  015360  015312  MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)     003166  026767  015316  015304  CMP     GOOD,BAD
(1)     003174  001401                  BEQ     .+4
(1)     003176  104003                  ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)     003200                          LOOP    T5A
(2)     003220  000137  003142          JMP     @#T5A

510     003224                          CKTCR   #104000,LP      ;TEST FOR ERR + SSO
(1)     003224                          LP=.
(1)     003224  012767  104000  015256  MOV     #104000,GOOD
(1)     003232  017767  015302  015240  MOV     @TCR,BAD        ;READ CMD REGISTER
(1)     003240  026767  015244  015232  CMP     GOOD,BAD
(1)     003246  001401                  BEQ     .+4
(1)     003250  104003                  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)     003252                          LOOP    LP
(2)     003272  000137  003224          JMP     @#LP

511     003276  104400                  TYPE
512     003300  016715                  P2MSG3          ;'PRESS STOP-RESET AND PRESS START ON THE 8035-8045 ''
513     003302  132777  000020  015236  T5B:    BITB    #20,@TSRHI
514     003310  001374                  BNE     T5B             ;WAIT FOR STACKER FULL TO GO AWAY
515     003312                          CKTSR   #0,T5B          ;TEST STATUS REG FOR 0
(1)     003312                          LP=.
```

```
    (1)  003312  012767  000000  015170          MOV     #0,GOOD
    (1)  003320  017767  015220  015152          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  003326  026767  015156  015144          CMP     GOOD,BAD
    (1)  003334  001401                          BEQ     .+4
    (1)  003336  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
    (1)  003340                                  LOOP    T5B
    (2)  003360  000137  003302                  JMP     @#T5B
  516    003364                                  CKTCR   #104000,LP      ;TEST FOR ERR + SSO
    (1)          003364                          LP=.
    (1)  003364  012767  104000  015116          MOV     #104000,GOOD
    (1)  003372  017767  015142  015100          MOV     @TCR,BAD        ;READ CMD REGISTER
    (1)  003400  026767  015104  015072          CMP     GOOD,BAD
    (1)  003406  001401                          BEQ     .+4
    (1)  003410  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
    (1)  003412                                  LOOP    LP
    (2)  003432  000137  003364                  JMP     @#LP
  517                           ;TEST FOR STACKER #2 FULL
  518    003436  112777  000031  015076          MOVB    #31,@TCRHI      ;SELECT STACKER #2 + CLEAR ERR
  519    003444  104400                          TYPE
  520    003446  016046                          T5MSG5                  ;'PULL CARD PLATE OF STACKER #2 BACK TO POSITION WHICH
  521    003450  104400                          TYPE
  522    003452  015665                          T5MSG2                  ;WILL SIMULATE STACKER FULL AND HOLD IT THERE''
  523    003454  104400                          TYPE
  524    003456  015735                          T5MSG3                  ;'PROGRAM WILL WAIT FOR STACKER FULL FLAG''
  525    003460  132777  000020  015060  T5C:    BITB    #20,@TSRHI
  526    003466  001774                          BEQ     T5C             ;WAIT FOR STACKER FULL FLAG
  527    003470                                  CKTSR   #110000,T5C     ;TEST FOR OE + SF
    (1)          003470                          LP=.
    (1)  003470  012767  110000  015012          MOV     #110000,GOOD
    (1)  003476  017767  015042  014774          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  003504  026767  015000  014766          CMP     GOOD,BAD
    (1)  003512  001401                          BEQ     .+4
    (1)  003514  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
    (1)  003516                                  LOOP    T5C
    (2)  003536  000137  003460                  JMP     @#T5C
  528    003542  104400                          TYPE
  529    003544  016715                          P2MSG3                  ;'PRESS STOP-RESET AND PRESS ON THE 8035-8045 ''
  530    003546  132777  000020  014772  T5D:    BITB    #20,@TSRHI
  531    003554  001374                          BNE     T5D             ;WAIT FOR STACKER FULL TO GO AWAY
  532    003556                                  CKTSR   #0,T5D          ;TEST STATUS REG FOR 0
    (1)          003556                          LP=.
    (1)  003556  012767  000000  014724          MOV     #0,GOOD
    (1)  003564  017767  014754  014706          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  003572  026767  014712  014700          CMP     GOOD,BAD
    (1)  003600  001401                          BEQ     .+4
    (1)  003602  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
    (1)  003604                                  LOOP    T5D
    (2)  003624  000137  003546                  JMP     @#T5D
  533                           ;TEST FOR STACKER FULL IN STACKER OVERFLOW MODE
  534    003630  104400                          TYPE
  535    003632  015536                          T5M2                    ;'PRESS START ON THE 8035-8045 ''
  536    003634  004767  006146                  JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE
  537    003640  000005                          RESET                   ;CLEAR OUT CMD REG
  538    003642  012777  000001  014670          MOV     #1,@TCR         ;SET STACKER OVERFLOW MODE + GO
  539    003650  104400                          TYPE
  540    003652  015576                          T5MSG1                  ;'PULL CARD PLATE OF STACKER #1 BACK TO POSITION WHICH
```

```
                                        TYPE
                                        T5MSG2                  ;WILL SIMULATE STACKER FULL AND RELEASE''
003654  104400
003656  015665
003660  132777  000020  014660  T5E:    BITB    #20,@TSRHI
003666  001401                        BEQ     .+4
003670  104000                        ERR                     ;STACKER FULL FLAG SET WITH ONLY STACKER #1
                                                              ;FULL WHEN IN OVERFLOW MODE
003672                                LOOP    T5E
(1) 003712  000137  003660            JMP     @#T5E
003716  004767  006064                JSR     PC,CONTIN        ;'PRESS CR TO CONTINUE''
003722  104400                        TYPE
003724  016046                        T5MSG5                  ;'PULL CARD PLATE OF STACKER #2 BACK TO POSITION WHICH
003726  104400                        TYPE
003730  015665                        T5MSG2                  ;WILL SIMULATE STACKER FULL AND RELEASE''
003732  104400                        TYPE
003734  015735                        T5MSG3                  ;'PROGRAM WILL WAIT FOR STACKER FULL FLAG''
003736  132777  000020  014602  T5F:    BITB    #20,@TSRHI
003744  001774                        BEQ     T5F             ;WAIT FOR STACKER FULL FLAG
003746                                CKTSR   #110000,T5F     ;TEST FOR OE + SF
(1)     003746                        LP=.
(1) 003746  012767  110000  014534    MOV     #110000,GOOD
(1) 003754  017767  014564  014516    MOV     @TSR,BAD        ;READ STATUS REGISTER
(1) 003762  026767  014522  014510    CMP     GOOD,BAD
(1) 003770  001401                    BEQ     .+4
(1) 003772  104003                    ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 003774                            LOOP    T5F
(2) 004014  000137  003736            JMP     @#T5F
004020                                CKTCR   #100000,LP      ;TEST FOR ERR
(1)     004020                        LP=.
(1) 004020  012767  100000  014462    MOV     #100000,GOOD
(1) 004026  017767  014506  014444    MOV     @TCR,BAD        ;READ CMD REGISTER
(1) 004034  026767  014450  014436    CMP     GOOD,BAD
(1) 004042  001401                    BEQ     .+4
(1) 004044  104003                    ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 004046                            LOOP    LP
(2) 004066  000137  004020            JMP     @#LP
004072  000005                        RESET

004074  000137  004100                JMP     @#TEST6

;
;FORCE INPUT CHECK ERROR BY TRYING TO FEED A CARD FROM AN EMPTY HOPPER
;
004100  000005                TEST6:  RESET
004102  104400                        TYPE
004104  016715                        P2MSG3                  ;'PRESS  STOP-RESET AND PRESS ON THE 8035-8045 ''
004106  004767  005674                JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE''
004112  012777  020001  014420        MOV     #20001,@TCR     ;ISSUE GO COMMAND TO PRIMARY HOPPER
004120  005067  014360                CLR     CNT
004124  162767  000001  014352        SUB     #1,CNT
004132  001374                        BNE     .-6
004134  162767  000001  014342        SUB     #1,CNT
004142  001374                        BNE     .-6
004144  162767  000001  014332        SUB     #1,CNT
004152  001374                        BNE     .-6
004154  162767  000001  014322        SUB     #1,CNT
004162  001374                        BNE     .-6             ;DELAY 1 SECOND
004164                                CKTSR   #320,LP         ;TEST FOR IE + IC + HE
```

```
(1)             004164
(1)   004164  012767  000320  014316          LP=.
(1)   004164  012767  000320  014316          MOV     #320,GOOD
(1)   004172  017767  014346  014300          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)   004200  026767  014304  014272          CMP     GOOD,BAD
(1)   004206  001401                          BEQ     .+4
(1)   004210  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)   004212                                   LOOP    LP
(2)   004232  000137  004164                  JMP     @#LP
580   004236  000005                          RESET
581
582   004240  000207                          RTS     PC              ;EXIT LOGIC TEST
583                                  ;
584                                  ;THIS TEST FEEDS 20 BLANK CARDS, 10 FROM EACH HOPPER, INTO THE STACKERS
585                                  ;       (10 INTO EACH STACKER). THE CARDS ARE TESTED FOR ALL SPACES AND
586                                  ;       TESTING IS DONE TO INSURE THAT THE INPUT DATA REQUEST CAN CAUSE
587                                  ;       AN INTERRUPT.
588                                  ;
589   004242  000005                 PASS1:  RESET
590   004244  104400                         TYPE
591   004246  016274                         CRDTST                  ;''***** CARD TESTS *****''
592   004250  104400                         TYPE
593   004252  016327                         P1MSG1                  ;'REMOVE ALL CARDS AND PUT THE 8035-8045  OFF-LINE''
594   004254  004767  005526                 JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE''
595   004260  004767  007420                 JSR     PC,CKSWR
596   004264  000005                         RESET
597   004266  112777  000010  014246         MOVB    #10,@TCRHI      ;SELECT STACKER #1
598   004274  104400                         TYPE
599   004276  016412                         P1MSG2                  ;'PUT 10 BLANK CARDS IN EACH INPUT HOPPER''
600   004300  104400                         TYPE
601   004302  016464                         P1MSG3                  ;'PUT 8035-8045  ON-LINE AND PRESS
602                                                                   ; START ON THE 8035-8045 ''
603   004304  004767  005476                 JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE''
604                                  ;DONE + SSO SHOULD BE THE ONLY BITS SET IN THE CMD REG
605   004310                                 CKTCR   #4200,LP        ;TEST FOR SSO + DONE FLAG
(1)             004310                         LP=.
(1)   004310  012767  004200  014172          MOV     #4200,GOOD
(1)   004316  017767  014216  014154          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)   004324  026767  014160  014146          CMP     GOOD,BAD
(1)   004332  001401                          BEQ     .+4
(1)   004334  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)   004336                                   LOOP    LP
(2)   004356  000137  004310                  JMP     @#LP
606                                ;SEE IF ISSUING GO CLEARS THE DONE FLAG (READ 1ST CARD)
607   004362  012777  024001  014150          MOV     #24001,@TCR     ;ISSUE HSO + SSO + GO
608   004370                                 CKTCR   #24000,LP       ;TEST FOR HSO + SSO
(1)             004370                         LP=.
(1)   004370  012767  024000  014112          MOV     #24000,GOOD
(1)   004376  017767  014136  014074          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)   004404  026767  014100  014066          CMP     GOOD,BAD
(1)   004412  001401                          BEQ     .+4
(1)   004414  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)   004416                                   LOOP    LP
(2)   004436  000137  004370                  JMP     @#LP
609   004442  005067  014036                  CLR     CNT
610   004446  162767  000001  014030          SUB     #1,CNT
611   004454  001374                          BNE     .-6
```

```
                                        SUB     #1,CNT
                                        BNE     .-6             ;DELAY 1 SECOND
                                        CKTSR   #4011,LP        ;TEST FOR CIW + RDR
                                        LP=.
                                        MOV     #4011,GOOD
                                        MOV     @TSR,BAD        ;READ STATUS REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    LP
                                        JMP     @#LP
                                        CKTIR   #200,LP         ;TEST FOR IDR
                                        LP=.
                                        MOV     #200,GOOD
                                        MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    LP
                                        JMP     @#LP
                    ;SEE IF INPUT DATA REQUEST WILL CAUSE AN INTERRUPT
                                        BIS     #1100,@TCR      ;ENABLE INT
                                        MOV     #P1INT,@PCV     ;SETUP PC VECTOR
                                        MOV     #340,@PSV       ;SETUP PS VECTOR
                    P1A:                MOV     #40,PS          ;SET BPU TO LEVEL #1
                                        NOP
                                        NOP
                                        MOV     #340,PS         ;SET BPU TO LEVEL #7
                                        ERR                     ;INPUT DATA REQUEST DIDN'T CAUSE AN INTERRUPT
                                        LOOP    P1A
                                        JMP     @#P1A
                                        BR      P1B
                    P1INT:              ADD     #4,SP           ;HOUSE KEEPING ON STACK
                    P1B:                JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
                                        CKTSR   #4010,LP        ;TEST FOR CIW
                                        LP=.
                                        MOV     #4010,GOOD
                                        MOV     @TSR,BAD        ;READ STATUS REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    LP
                                        JMP     @#LP
                                        CKTCR   #25300,LP       ;TEST FOR HSO + SSO + DONE + INT EN
                                        LP=.
                                        MOV     #25300,GOOD
                                        MOV     @TCR,BAD        ;READ CMD REGISTER
                                        CMP     GOOD,BAD
                                        BEQ     .+4
                                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
                                        LOOP    LP
                                        JMP     @#LP
                    ;READ CARDS #2-9 FROM PRIMARY HOPPER
                                        MOV     #10,CNT1
                    P1D:                MOV     #25001,@TCR     ;ISSUE HSO + SSO + GO
                                        WTIDR
```

```
   (1)  005054  032777  000200  013472          BIT     #200,@TIR
   (1)  005062  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
   635  005064                                   CKTSR   #4011,LP        ;TEST FOR CIW + RDR
   (1)          005064                          LP=.
   (1)  005064  012767  004011  013416          MOV     #4011,GOOD
   (1)  005072  017767  013446  013400          MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  005100  026767  013404  013372          CMP     GOOD,BAD
   (1)  005106  001401                           BEQ     .+4
   (1)  005110  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  005112                                   LOOP    LP
   (2)  005132  000137  005064                   JMP     @#LP
   636  005136  004767  004674                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
   637  005142                                   CKTCR   #25200,LP       ;TEST FOR HS0 + SS0 + DONE
   (1)          005142                          LP=.
   (1)  005142  012767  025200  013340          MOV     #25200,GOOD
   (1)  005150  017767  013364  013322          MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  005156  026767  013326  013314          CMP     GOOD,BAD
   (1)  005164  001401                           BEQ     .+4
   (1)  005166  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  005170                                   LOOP    LP
   (2)  005210  000137  005142                   JMP     @#LP
   638  005214  005367  013266                   DEC     CNT1
   639  005220  001312                           BNE     P1D
   640                          ;READ CARD #10 OUT OF PRIMARY HOPPER
   641  005222  012777  025001  013310          MOV     #25001,@TCR     ;ISSUE HS0 + SS0 + GO
   642  005230                                   WTIDR
   (1)  005230  032777  000200  013316          BIT     #200,@TIR
   (1)  005236  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
   643  005240  004767  004572                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
   644  005244                                   CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
   (1)          005244                          LP=.
   (1)  005244  012767  004230  013236          MOV     #4230,GOOD
   (1)  005252  017767  013266  013220          MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  005260  026767  013224  013212          CMP     GOOD,BAD
   (1)  005266  001401                           BEQ     .+4
   (1)  005270  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  005272                                   LOOP    LP
   (2)  005312  000137  005244                   JMP     @#LP
   645  005316                                   CKTCR   #125200,LP      ;TEST FOR ERR + HS0 + SS0 + DONE
   (1)          005316                          LP=.
   (1)  005316  012767  125200  013164          MOV     #125200,GOOD
   (1)  005324  017767  013210  013146          MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  005332  026767  013152  013140          CMP     GOOD,BAD
   (1)  005340  001401                           BEQ     .+4
   (1)  005342  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  005344                                   LOOP    LP
   (2)  005364  000137  005316                   JMP     @#LP
   646                          ;READ CARD #11 OUT OF SECONDARY HOPPER
   647  005370  012777  045401  013142          MOV     #045401,@TCR    ;ISSUE CLEAR ERR + HS1 + SS0 + GO
   648  005376                                   WTIDR
   (1)  005376  032777  000200  013150          BIT     #200,@TIR
   (1)  005404  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
   649  005406                                   CKTSR   #4011,LP        ;TEST FOR CIW + RDR
   (1)          005406                          LP=.
   (1)  005406  012767  004011  013074          MOV     #4011,GOOD
   (1)  005414  017767  013124  013056          MOV     @TSR,BAD        ;READ STATUS REGISTER
```

```
(1)  005422  026767  013062  013050          CMP     GOOD,BAD
(1)  005430  001401                          BEQ     .+4
(1)  005432  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  005434                                   LOOP    LP
(2)  005454  000137  005406                   JMP     @#LP
650  005460  004767  004352                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
651  005464                                   CKTCR   #45200,LP       ;TEST FOR HS1 + SS0 + DONE
(1)          005464                          LP=.
(1)  005464  012767  045200  013016          MOV     #45200,GOOD
(1)  005472  017767  013042  013000          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  005500  026767  013004  012772          CMP     GOOD,BAD
(1)  005506  001401                          BEQ     .+4
(1)  005510  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  005512                                   LOOP    LP
(2)  005532  000137  005464                   JMP     @#LP
652                             ;READ CARDS #12-19 AND PUT THEM IN STACKER #2
653  005536  012767  000010  012742          MOV     #10,CNT1
654  005544  012777  055001  012766  P1E:     MOV     #55001,@TCR     ;ISSUE HS1 + SS1 + GO
655  005552                                   WTIDR
(1)  005552  032777  000200  012774          BIT     #200,@TIR
(1)  005560  001774                          BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
656  005562  004767  004250                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
657  005566                                   WTDONE
(1)  005566  032777  000200  012744          BIT     #200,@TCR
(1)  005574  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
658  005576                                   CKTSR   #4010,LP        ;TEST FOR CIW
(1)          005576                          LP=.
(1)  005576  012767  004010  012704          MOV     #4010,GOOD
(1)  005604  017767  012734  012666          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  005612  026767  012672  012660          CMP     GOOD,BAD
(1)  005620  001401                          BEQ     .+4
(1)  005622  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  005624                                   LOOP    LP
(2)  005644  000137  005576                   JMP     @#LP
659  005650                                   CKTCR   #55200,LP       ;TEST FOR HS1 + SS1 + DONE
(1)          005650                          LP=.
(1)  005650  012767  055200  012632          MOV     #55200,GOOD
(1)  005656  017767  012656  012614          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  005664  026767  012620  012606          CMP     GOOD,BAD
(1)  005672  001401                          BEQ     .+4
(1)  005674  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  005676                                   LOOP    LP
(2)  005716  000137  005650                   JMP     @#LP
660  005722  005367  012560                   DEC     CNT1
661  005726  001306                          BNE     P1E
662                             ;READ CARD #20 OUT OF SECONDARY HOPPER
663  005730  012777  051001  012602          MOV     #51001,@TCR     ;ISSUE HS1 + SS1 + GO
664  005736                                   WTIDR
(1)  005736  032777  000200  012610          BIT     #200,@TIR
(1)  005744  001774                          BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
665  005746  004767  004064                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
666  005752                                   CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
(1)          005752                          LP=.
(1)  005752  012767  004230  012530          MOV     #4230,GOOD
(1)  005760  017767  012560  012512          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  005766  026767  012516  012504          CMP     GOOD,BAD
```

```
(1)  005774  001401                        BEQ     .+4
(1)  005776  104003                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006000                                LOOP    LP
(2)  006020  000137  005752                JMP     @#LP
667  006024                                CKTCR   #151200,LP      ;TEST FOR ERR + HS1 + SS1 + DONE
(1)          006024                        LP=.
(1)  006024  012767  151200  012456        MOV     #151200,GOOD
(1)  006032  017767  012502  012440        MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  006040  026767  012444  012432        CMP     GOOD,BAD
(1)  006046  001401                        BEQ     .+4
(1)  006050  104003                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006052                                LOOP    LP
(2)  006072  000137  006024                JMP     @#LP
668                             ;PUT CARD #20 IN STACKER #2
669  006076  012777  011401  012434        MOV     #11401,@TCR     ;ISSUE CLEAR ERR + SS1 + GO
670  006104                                WTDONE
(1)  006104  032777  000200  012426        BIT     #200,@TCR
(1)  006112  001774                        BEQ     .-6             ;WAIT FOR DONE FLAG
671  006114                                CKTSR   #0,LP           ;TEST FOR STATUS=0
(1)          006114                        LP=.
(1)  006114  012767  000000  012366        MOV     #0,GOOD
(1)  006122  017767  012416  012350        MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  006130  026767  012354  012342        CMP     GOOD,BAD
(1)  006136  001401                        BEQ     .+4
(1)  006140  104003                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006142                                LOOP    LP
(2)  006162  000137  006114                JMP     @#LP
672  006166                                CKTCR   #11200,LP       ;TEST FOR SS1 + DONE
(1)          006166                        LP=.
(1)  006166  012767  011200  012314        MOV     #11200,GOOD
(1)  006174  017767  012340  012276        MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  006202  026767  012302  012270        CMP     GOOD,BAD
(1)  006210  001401                        BEQ     .+4
(1)  006212  104003                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006214                                LOOP    LP
(2)  006234  000137  006166                JMP     @#LP
673
674  006240  000137  006244                JMP     @#PASS2
675                             ;
676                             ;THIS TEST PUNCHES THE ENTIRE CHAR SET ON EACH CARD IN A PRECESSED
677                             ;       FASHION. THIS MEANS THE 1ST PUNCHED CHAR ON CARD #1 IS IN
678                             ;       COLUMN #11 AND THE 1ST PUNCHED CHAR OF CARD #20 IS IN COLUMN
679                             ;       #31. ALSO TEST THAT OUTPUT DATA REQUEST CAN CAUSE AN INTERRUPT.
680                             ;
681  006244  104400             PASS2:  TYPE
682  006246  016555                      P2MSG1                    ;'VERIFY THAT THERE ARE 10 BLANK CARDS IN EACH STACKER''
683  006250  104400                      TYPE
684  006252  016643                      P2MSG2                    ;'PLACE ALL 20 CARDS IN THE PRIMARY HOPPER''
685  006254  104400                      TYPE
686  006256  016715                      P2MSG3                    ;'PRESS STOP-RESET AND PRESS START ON THE 8035-8045 ''
687  006260  004767  003522              JSR     PC,CONTIN         ;'PRESS CR TO CONTINUE''
688  006264  004767  003700              JSR     PC,CLRBFS         ;PREPARE BUFFER FOR PUNCHING
689
690  006270  012777  021401  012242  ;FEED CARD #1
                                       MOV     #021401,@TCR      ;ISSUE CLEAR ERR + HS0 + GO
691  006276                                WTIDR
(1)  006276  032777  000200  012250        BIT     #200,@TIR
```

```
(1)  006304  001774                          BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
692  006306  004767  003524                  JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
693  006312                                  WTDONE
(1)  006312  032777  000200  012220          BIT     #200,@TCR
(1)  006320  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
694  006322                                  CKTSR   #4010,LP        ;TEST FOR CIW
(1)  006322                                  LP=.
(1)  006322  012767  004010  012160          MOV     #4010,GOOD
(1)  006330  017767  012210  012142          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  006336  026767  012146  012134          CMP     GOOD,BAD
(1)  006344  001401                          BEQ     .+4
(1)  006346  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006350                                  LOOP    LP
(2)  006370  000137  006322                  JMP     @#LP
695  006374                                  CKTCR   #21200,LP       ;TEST FOR HSO + DONE
(1)  006374                                  LP=.
(1)  006374  012767  021200  012106          MOV     #21200,GOOD
(1)  006402  017767  012132  012070          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  006410  026767  012074  012062          CMP     GOOD,BAD
(1)  006416  001401                          BEQ     .+4
(1)  006420  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006422                                  LOOP    LP
(2)  006442  000137  006374                  JMP     @#LP
696                             ;FEED CARDS #2-19, INHIBIT READ FOR THESE CARDS
697  006446  012767  017534  012054          MOV     #BF1,PTR
698  006454  012767  000022  012024          MOV     #22,CNT1        ;CNT1=18
699  006462  012777  021027  012050  P2A:    MOV     #21027,@TCR     ;ISSUE HSO + IR + PRI + PUN + GO
700  006470                                  WTODR
(1)  006470  132777  000200  012060          BITB    #200,@TIRHI
(1)  006476  001774                          BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
701  006500                                  CKTSR   #4410,LP        ;TEST FOR PUDR
(1)  006500                                  LP=.
(1)  006500  012767  004410  012002          MOV     #4410,GOOD
(1)  006506  017767  012032  011764          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  006514  026767  011770  011756          CMP     GOOD,BAD
(1)  006522  001401                          BEQ     .+4
(1)  006524  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006526                                  LOOP    LP
(2)  006546  000137  006500                  JMP     @#LP
702  006552  004767  003452                  JSR     PC,P2SEND       ;SEND DATA TO PUNCH
703  006556  162767  000002  011744          SUB     #2,PTR
704  006564                                  WTDONE
(1)  006564  032777  000200  011746          BIT     #200,@TCR
(1)  006572  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
705  006574                                  CKTSR   #4010,LP        ;TEST FOR CIW
(1)  006574                                  LP=.
(1)  006574  012767  004010  011706          MOV     #4010,GOOD
(1)  006602  017767  011736  011670          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  006610  026767  011674  011662          CMP     GOOD,BAD
(1)  006616  001401                          BEQ     .+4
(1)  006620  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006622                                  LOOP    LP
(2)  006642  000137  006574                  JMP     @#LP
706  006646                                  CKTCR   #21226,LP       ;TEST FOR HSO + DONE + IR + PRI + PUN
(1)  006646                                  LP=.
(1)  006646  012767  021226  011634          MOV     #21226,GOOD
```

```
(1)  006654  017767  011660  011616          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  006662  026767  011622  011610          CMP     GOOD,BAD
(1)  006670  001401                           BEQ     .+4
(1)  006672  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006674                                   LOOP    LP
(2)  006714  000137  006646                   JMP     @#LP
707  006720                                   CKTIR   #0,LP           ;TEST FOR 0
(1)       006720                              LP=.
(1)  006720  012767  000000  011562          MOV     #0,GOOD
(1)  006726  017767  011622  011544          MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
(1)  006734  026767  011550  011536          CMP     GOOD,BAD
(1)  006742  001401                           BEQ     .+4
(1)  006744  104003                           ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  006746                                   LOOP    LP
(2)  006766  000137  006720                   JMP     @#LP
708  006772  005367  011510                   DEC     CNT1            ;IF ALL CARDS ARE NOT DONE
709  006776  001231                           BNE     P2A             ;GO PUNCH ANOTHER CARD
710                                 ;FEED CARD #20
711  007000  012777  021007  011532          MOV     #21007,@TCR     ;ISSUE HSO + PRI + PUN + GO
712  007006                                   WTODR
(1)  007006  132777  000200  011542          BITB    #200,@TIRHI
(1)  007014  001774                           BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
713  007016  004767  003206                   JSR     PC,P2SEND       ;SEND DATA TO PUNCH
714  007022  162767  000002  011500          SUB     #2,PTR
715  007030                                   WTIDR
(1)  007030  032777  000200  011516          BIT     #200,@TIR
(1)  007036  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
716  007040  004767  002772                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
717  007044                                   WTDONE
(1)  007044  032777  000200  011466          BIT     #200,@TCR
(1)  007052  001774                           BEQ     .-6             ;WAIT FOR DONE FLAG
718  007054                                   CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
(1)       007054                              LP=.
(1)  007054  012767  004230  011426          MOV     #4230,GOOD
(1)  007062  017767  011456  011410          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  007070  026767  011414  011402          CMP     GOOD,BAD
(1)  007076  001401                           BEQ     .+4
(1)  007100  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  007102                                   LOOP    LP
(2)  007122  000137  007054                   JMP     @#LP
719  007126                                   CKTCR   #121206,LP      ;TEST FOR ERR + HSO + DONE + PRI + PUN
(1)       007126                              LP=.
(1)  007126  012767  121206  011354          MOV     #121206,GOOD
(1)  007134  017767  011400  011336          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  007142  026767  011342  011330          CMP     GOOD,BAD
(1)  007150  001401                           BEQ     .+4
(1)  007152  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  007154                                   LOOP    LP
(2)  007174  000137  007126                   JMP     @#LP
720                                 ;CARD #20 PUNCH AND PRINT
721  007200  012777  001407  011332          MOV     #1407,@TCR      ;ISSUE CLEAR ERR + PRI + PUN + GO
722  007206                                   WTODR
(1)  007206  132777  000200  011342          BITB    #200,@TIRHI
(1)  007214  001774                           BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
723  007216                                   CKTIR   #100000,LP      ;TEST FOR ODR
(1)       007216                              LP=.
```

```
(1)  007216  012767  100000  011264          MOV     #100000,GOOD
(1)  007224  017767  011324  011246          MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
(1)  007232  026767  011252  011240          CMP     GOOD,BAD
(1)  007240  001401                          BEQ     .+4
(1)  007242  104003                          ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  007244                                  LOOP    LP
(2)  007264  000137  007216                  JMP     @#LP
724  007270                                  CKTCR   #1006,LP                ;TEST FOR PRI + PUN
(1)            007270                        LP=.
(1)  007270  012767  001006  011212          MOV     #1006,GOOD
(1)  007276  017767  011236  011174          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  007304  026767  011200  011166          CMP     GOOD,BAD
(1)  007312  001401                          BEQ     .+4
(1)  007314  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  007316                                  LOOP    LP
(2)  007336  000137  007270                  JMP     @#LP
725                          ;SEE IF OUTPUT DATA REQUEST WILL CAUSE AN INTERRUPT
726  007342  052777  001100  011170          BIS     #1100,@TCR      ;ENABLE INTERRUPT
727  007350  012777  007434  011202          MOV     #P2INT,@PCV     ;SETUP PC VECTOR
728  007356  012777  000340  011176          MOV     #340,@PSV       ;SETUP PS VECTOR
729  007364  012767  000040  170404          MOV     #40,PS          ;SET BPU TO LEVEL #1
730  007372  000240                          NOP
731  007374  000240                          NOP
732  007376  012767  000340  170372          MOV     #340,PS         ;SET BPU TO LEVEL #7
733  007404  104000              P2B:        ERR                     ;OUTPUT DATA REQUEST DIDN'T CAUSE AN INTERRUPT
734  007406                                  LOOP    P2B
(1)  007426  000137  007404                  JMP     @#P2B
735  007432  000402                          BR      P2C
736  007434  062706  000004      P2INT:      ADD     #4,SP           ;HOUSE KEEPING ON STACK
737  007440  004767  002564      P2C:        JSR     PC,P2SEND       ;SEND DATA TO PUNCH
738  007444                                  WTDONE
(1)  007444  032777  000200  011066          BIT     #200,@TCR
(1)  007452  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
739  007454                                  CKTSR   #0,LP           ;TEST FOR STATUS =0
(1)            007454                        LP=.
(1)  007454  012767  000000  011026          MOV     #0,GOOD
(1)  007462  017767  011056  011010          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  007470  026767  011014  011002          CMP     GOOD,BAD
(1)  007476  001401                          BEQ     .+4
(1)  007500  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  007502                                  LOOP    LP
(2)  007522  000137  007454                  JMP     @#LP
740  007526                                  CKTCR   #1306,LP                ;TEST FOR DONE + INT EN + PRI + PUN
(1)            007526                        LP=.
(1)  007526  012767  001306  010754          MOV     #1306,GOOD
(1)  007534  017767  011000  010736          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  007542  026767  010742  010730          CMP     GOOD,BAD
(1)  007550  001401                          BEQ     .+4
(1)  007552  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  007554                                  LOOP    LP
(2)  007574  000137  007526                  JMP     @#LP
741
742  007600  000137  007604                  JMP     @#PASS3
743                                  ;
744                                  ;THIS TEST PERFORMS PUNCHING WITHOUT PRINTING. ONE ALL ONES CHAR IS
745                                  ;       PUNCHED IN COLUMN 1 OF CARD 1 AND SO ON UNTIL 20 ALL ONES CHARS
```

```
                                        ;       ARE PUNCHED IN THE 1ST 20 COLUMNS OF CARD 20. THIS TEST
                                        ;       VERIFIES THE DATA READ FROM PUNCHING DONE IN PASS 2.
                                        ;
007604  104400                          PASS3:  TYPE
007606  017044                                  P3MSG1                  ;'VERIFY THAT ALL CHARS WERE PUNCHED ON EACH CARD IN A
007610  104400                                  TYPE
007612  017132                                  P3MSG2                  ;PRECESS FASHION BEGINNING WITH COLUMN 11 OF 1ST CARD''
007614  104400                                  TYPE
007616  016643                                  P2MSG2                  ;'PLACE ALL 20 CARDS IN THE PRIMARY HOPPER''
007620  104400                                  TYPE
007622  016715                                  P2MSG3                  ;'PRESS STOP-RESET AND PRESS START ON THE 8035-8045 ''
007624  004767  002156                        JSR     PC,CONTIN       ;'PRESS CR TO CONTINUE''
                                        ;FEED CARD #1
007630  012767  017534  010656                MOV     #BF1,INPTR
007636  012777  021401  010674                MOV     #021401,@TCR    ;ISSUE CLEAR ERR + HSO + GO
007644                                          WTIDR
(1) 007644  032777  000200  010702              BIT     #200,@TIR
(1) 007652  001774                              BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
007654  004767  002624                        JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASS 2
007660  162767  000002  010626                SUB     #2,INPTR
007666                                          WTDONE
(1) 007666  032777  000200  010644              BIT     #200,@TCR
(1) 007674  001774                              BEQ     .-6             ;WAIT FOR DONE FLAG
007676                                          CKTSR   #4010,LP        ;TEST FOR CIW
(1)         007676                              LP=.
(1) 007676  012767  004010  010604              MOV     #4010,GOOD
(1) 007704  017767  010634  010566              MOV     @TSR,BAD        ;READ STATUS REGISTER
(1) 007712  026767  010572  010560              CMP     GOOD,BAD
(1) 007720  001401                              BEQ     .+4
(1) 007722  104003                              ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 007724                                      LOOP    LP
(2) 007744  000137  007676                      JMP     @#LP
007750                                          CKTCR   #21200,LP       ;TEST FOR HSO + DONE
(1)         007750                              LP=.
(1) 007750  012767  021200  010532              MOV     #21200,GOOD
(1) 007756  017767  010556  010514              MOV     @TCR,BAD        ;READ CMD REGISTER
(1) 007764  026767  010520  010506              CMP     GOOD,BAD
(1) 007772  001401                              BEQ     .+4
(1) 007774  104003                              ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 007776                                      LOOP    LP
(2) 010016  000137  007750                      JMP     @#LP
                                        ;FEED CARDS #2-19, CHECKING INPUT DATA AND PUNCHING 1 CHAR IN 1ST CARD,
                                        ;       2 IN 2ND ETC. UP TO 20 CHARS IN CARD 20
010022  012767  000022  010456                  MOV     #22,CNT1        ;CNT1=18
010030  012767  000001  010444                  MOV     #1,CARD         ;CARD=1
010036  012777  021003  010474          P3A:    MOV     #21003,@TCR     ;ISSUE HSO + PUN + GO
010044                                          WTODR
(1) 010044  132777  000200  010504              BITB    #200,@TIRHI
(1) 010052  001774                              BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
010054  004767  002224                        JSR     PC,P3SEND       ;SEND DATA TO PUNCH
010060  062767  000001  010414                ADD     #1,CARD
010066                                          WTIDR
(1) 010066  032777  000200  010460              BIT     #200,@TIR
(1) 010074  001774                              BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
010076  004767  002402                        JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASS 2
010102  162767  000002  010404                SUB     #2,INPTR
```

```
   778  010110
   (1)  010110  032777  000200  010422      WTDONE
   (1)  010116  001774                      BIT     #200,@TCR
   779  010120                               BEQ     .-6
   (1)          010120                      CKTCR   #21202,LP       ;WAIT FOR DONE FLAG
   (1)  010120  012767  021202  010362      LP=.                    ;TEST FOR HSO + DONE + PUN
   (1)  010126  017767  010406  010344      MOV     #21202,GOOD
   (1)  010134  026767  010350  010336      MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  010142  001401                      CMP     GOOD,BAD
   (1)  010144  104003                      BEQ     .+4
   (1)  010146                              ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (2)  010166  000137  010120              LOOP    LP
   780  010172                              JMP     @#LP
   (1)          010172                      CKTSR   #4010,LP        ;TEST FOR CIW
   (1)  010172  012767  004010  010310      LP=.
   (1)  010200  017767  010340  010272      MOV     #4010,GOOD
   (1)  010206  026767  010276  010264      MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  010214  001401                      CMP     GOOD,BAD
   (1)  010216  104003                      BEQ     .+4
   (1)  010220                              ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (2)  010240  000137  010172              LOOP    LP
   781  010244                              JMP     @#LP
   (1)          010244                      CKTIR   #0,LP           ;TEST INTERRUPT REG FOR 0
   (1)  010244  012767  000000  010236      LP=.
   (1)  010252  017767  010276  010220      MOV     #0,GOOD
   (1)  010260  026767  010224  010212      MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
   (1)  010266  001401                      CMP     GOOD,BAD
   (1)  010270  104003                      BEQ     .+4
   (1)  010272                              ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (2)  010312  000137  010244              LOOP    LP
   782  010316  005367  010164              JMP     @#LP
   783  010322  001245                      DEC     CNT1
   784                                      BNE     P3A             ;GO PUNCH AND READ MORE CARDS
                                    ;FEED CARD #20
   785  010324  012777  021003  010206      MOV     #21003,@TCR     ;ISSUE HSO + PUN + GO
   786  010332                              WTODR
   (1)  010332  132777  000200  010216      BITB    #200,@TIRHI
   (1)  010340  001774                      BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
   787  010342  004767  001736              JSR     PC,P3SEND       ;SEND DATA TO PUNCH
   788  010346  062767  000001  010126      ADD     #1,CARD
   789  010354                              WTIDR
   (1)  010354  032777  000200  010172      BIT     #200,@TIR
   (1)  010362  001774                      BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
   790  010364  004767  002114              JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASS 2
   791  010370                              WTDONE
   (1)  010370  032777  000200  010142      BIT     #200,@TCR
   (1)  010376  001774                      BEQ     .-6             ;WAIT FOR DONE FLAG
   792  010400                              CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
   (1)          010400                      LP=.
   (1)  010400  012767  004230  010102      MOV     #4230,GOOD
   (1)  010406  017767  010132  010064      MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  010414  026767  010070  010056      CMP     GOOD,BAD
   (1)  010422  001401                      BEQ     .+4
   (1)  010424  104003                      ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  010426                              LOOP    LP
   (2)  010446  000137  010400              JMP     @#LP
   793  010452                              CKTCR   #121202,LP      ;TEST FOR ERR + HSO + DONE + PUN
```

```
(1)             010452
(1)  010452  012767  121202  010030          LP=.
(1)  010460  017767  010054  010012          MOV     #121202,GOOD
(1)  010466  026767  010016  010004          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  010474  001401                          CMP     GOOD,BAD
(1)  010476  104003                          BEQ     .+4
(1)  010500                                  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(2)  010520  000137  010452                  LOOP    LP
794                                          JMP     @#LP
                                    ;PUNCH CARD #20
795  010524  012777  001003  010006          MOV     #1003,@TCR              ;ISSUE PUN + GO
796  010532                                  WTODR
(1)  010532  132777  000200  010016          BITB    #200,@TIRHI
(1)  010540  001774                          BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
797  010542  004767  001536                  JSR     PC,P3SEND       ;SEND DATA TO PUNCH
798  010546                                  WTDONE
(1)  010546  032777  000200  007764          BIT     #200,@TCR
(1)  010554  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
799  010556                                  CKTSR   #0,LP           ;TEST FOR STATUS=0
(1)             010556                       LP=.
(1)  010556  012767  000000  007724          MOV     #0,GOOD
(1)  010564  017767  007754  007706          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  010572  026767  007712  007700          CMP     GOOD,BAD
(1)  010600  001401                          BEQ     .+4
(1)  010602  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  010604                                  LOOP    LP
(2)  010624  000137  010556                  JMP     @#LP
800  010630                                  CKTCR   #001202,LP      ;TEST FOR ERR + DONE
(1)             010630                       LP=.
(1)  010630  012767  001202  007652          MOV     #001202,GOOD
(1)  010636  017767  007676  007634          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  010644  026767  007640  007626          CMP     GOOD,BAD
(1)  010652  001401                          BEQ     .+4
(1)  010654  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  010656                                  LOOP    LP
(2)  010676  000137  010630                  JMP     @#LP
801  010702  000137  010706                  JMP     @#PASS4
802                                 ;
803                                 ;THIS TEST PERFORMS A SEPARATE PRINT OPERATION ON ALL CARDS BY
804                                 ;       PRINTING ''CARD 01'' IN COLUMNS 1-7 OF CARD 1 UP TO ''CARD 20'' IN
805                                 ;       COLUMNS 1-7 OF CARD 20. VERIFY CORRECT DATA READ FROM PUNCHING
806                                 ;       DONE IN PASSES 2 + 3
807                                 ;
808  010706  104400                 PASS4:  TYPE
809  010710  017220                          P4MSG1                  ;''VERIFY THAT THERE IS 1 ALL ONES CHAR PUNCHED IN CARD 1
810  010712  104400                          TYPE
811  010714  017373                          P4MSG2                  ;UP TO 20 ALL ONES CHARS PUNCHED IN CARD 20''
812  010716  104400                          TYPE
813  010720  016643                          P2MSG2                  ;''PLACE ALL 20 CARDS IN THE PRIMARY HOPPER''
814  010722  104400                          TYPE
815  010724  016715                          P2MSG3                  ;''PRESS STOP-RESET AND PRESS START ON THE 8035-8045 ''
816  010726  004767  001054                  JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
817  010732  004767  001252                  JSR     PC,SETBFS       ;PREPARE BUFFER FOR DATA COMPARE
818                                 ;FEED CARD #1
819  010736  012767  017534  007550          MOV     #BF1,INPTR
820  010744  012777  021401  007566          MOV     #21401,@TCR     ;ISSUE CLEAR ERR + HSO + GO
821  010752                                  WTIDR
```

```
(1) 010752 032777 000200 007574          BIT     #200,@TIR
(1) 010760 001774                        BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
822 010762 004767 001516                 JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASSES 2 + 3
823 010766 162767 000002 007520          SUB     #2,INPTR
824 010774                               WTDONE
(1) 010774 032777 000200 007536          BIT     #200,@TCR
(1) 011002 001774                        BEQ     .-6             ;WAIT FOR DONE FLAG
825 011004                               CKTSR   #4010,LP        ;TEST FOR CIW
(1)        011004                        LP=.
(1) 011004 012767 004010 007476          MOV     #4010,GOOD
(1) 011012 017767 007526 007460          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1) 011020 026767 007464 007452          CMP     GOOD,BAD
(1) 011026 001401                        BEQ     .+4
(1) 011030 104003                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 011032                               LOOP    LP
(2) 011052 000137 011004                 JMP     @#LP
826 011056                               CKTCR   #21200,LP       ;TEST FOR HSO + DONE
(1)        011056                        LP=.
(1) 011056 012767 021200 007424          MOV     #21200,GOOD
(1) 011064 017767 007450 007406          MOV     @TCR,BAD        ;READ CMD REGISTER
(1) 011072 026767 007412 007400          CMP     GOOD,BAD
(1) 011100 001401                        BEQ     .+4
(1) 011102 104003                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 011104                               LOOP    LP
(2) 011124 000137 011056                 JMP     @#LP
827                         ;FEED CARDS #2-19, CHECKING INPUT DATA AND PRINTING IN COLUMNS 1-7
828 011130 012767 000022 007350          MOV     #22,CNT1        ;CNT1=18
829 011136 012767 020046 007360          MOV     #CARD01,OUTPTR  ;OUTPTR=CARD01
830 011144 012777 021015 007366 P4A:     MOV     #21015,@TCR     ;ISSUE HSO + SPD + PRI + GO
831 011152                               WTODR
(1) 011152 132777 000200 007376          BITB    #200,@TIRHI
(1) 011160 001774                        BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
832 011162 004767 001212                 JSR     PC,P4SEND       ;SEND DATA TO PRINTER
833 011166                               WTIDR
(1) 011166 032777 000200 007360          BIT     #200,@TIR
(1) 011174 001774                        BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
834 011176 004767 001302                 JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASSES 2 + 3
835 011202 162767 000002 007304          SUB     #2,INPTR
836 011210                               WTDONE
(1) 011210 032777 000200 007322          BIT     #200,@TCR
(1) 011216 001774                        BEQ     .-6             ;WAIT FOR DONE FLAG
837 011220                               CKTCR   #21214,LP       ;TEST FOR HSO + DONE + SPD + PRI
(1)        011220                        LP=.
(1) 011220 012767 021214 007262          MOV     #21214,GOOD
(1) 011226 017767 007306 007244          MOV     @TCR,BAD        ;READ CMD REGISTER
(1) 011234 026767 007250 007236          CMP     GOOD,BAD
(1) 011242 001401                        BEQ     .+4
(1) 011244 104003                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1) 011246                               LOOP    LP
(2) 011266 000137 011220                 JMP     @#LP
838 011272                               CKTSR   #4010,LP        ;TEST FOR CIW
(1)        011272                        LP=.
(1) 011272 012767 004010 007210          MOV     #4010,GOOD
(1) 011300 017767 007240 007172          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1) 011306 026767 007176 007164          CMP     GOOD,BAD
(1) 011314 001401                        BEQ     .+4
```

```
   (1)  011316  104003                          ERR+GB                ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  011320                                   LOOP    LP
   (2)  011340  000137  011272                   JMP     @#LP
  839   011344                                   CKTIR   #0,LP         ;TEST INTERRUPT REG FOR 0
   (1)          011344                          LP=.
   (1)  011344  012767  000000  007136           MOV     #0,GOOD
   (1)  011352  017767  007176  007120           MOV     @TIR,BAD      ;READ INTERRUPT REGISTER
   (1)  011360  026767  007124  007112           CMP     GOOD,BAD
   (1)  011366  001401                           BEQ     .+4
   (1)  011370  104003                           ERR+GB                ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  011372                                   LOOP    LP
   (2)  011412  000137  011344                   JMP     @#LP
  840   011416  005367  007064                   DEC     CNT1
  841   011422  001250                           BNE     P4A           ;GO PRINT AND READ MORE CARDS
  842                                ;FEED CARD #20
  843   011424  012777  021015  007106           MOV     #21015,@TCR   ;ISSUE HSO + SPD + PRI + GO
  844   011432                                   WTODR
   (1)  011432  132777  000200  007116           BITB    #200,@TIRHI
   (1)  011440  001774                           BEQ     .-6           ;WAIT FOR OUTPUT DATA REQUEST
  845   011442  004767  000732                   JSR     PC,P4SEND     ;SEND DATA TO PRINTER
  846   011446                                   WTIDR
   (1)  011446  032777  000200  007100           BIT     #200,@TIR
   (1)  011454  001774                           BEQ     .-6           ;WAIT FOR INPUT DATA REQUEST
  847   011456  004767  001022                   JSR     PC,CKDATA     ;CHECK INPUT DATA FROM PASSES 2 + 3
  848   011462  162767  000002  007024           SUB     #2,INPTR
  849   011470                                   WTDONE
   (1)  011470  032777  000200  007042           BIT     #200,@TCR
   (1)  011476  001774                           BEQ     .-6           ;WAIT FOR DONE FLAG
  850   011500                                   CKTSR   #4230,LP      ;TEST FOR CIW + IE + HE
   (1)          011500                          LP=.
   (1)  011500  012767  004230  007002           MOV     #4230,GOOD
   (1)  011506  017767  007032  006764           MOV     @TSR,BAD      ;READ STATUS REGISTER
   (1)  011514  026767  006770  006756           CMP     GOOD,BAD
   (1)  011522  001401                           BEQ     .+4
   (1)  011524  104003                           ERR+GB                ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  011526                                   LOOP    LP
   (2)  011546  000137  011500                   JMP     @#LP
  851   011552                                   CKTCR   #121214,LP    ;TEST FOR ERR + HSO  + SPD + PRI + DONE
   (1)          011552                          LP=.
   (1)  011552  012767  121214  006730           MOV     #121214,GOOD
   (1)  011560  017767  006754  006712           MOV     @TCR,BAD      ;READ CMD REGISTER
   (1)  011566  026767  006716  006704           CMP     GOOD,BAD
   (1)  011574  001401                           BEQ     .+4
   (1)  011576  104003                           ERR+GB                ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
   (1)  011600                                   LOOP    LP
   (2)  011620  000137  011552                   JMP     @#LP
  852                                ;PRINT CARD #20
  853   011624  012777  001415  006706           MOV     #01415,@TCR   ;ISSUE CLEAR ERR + SPD + PRI + GO
  854   011632                                   WTODR
   (1)  011632  132777  000200  006716           BITB    #200,@TIRHI
   (1)  011640  001774                           BEQ     .-6           ;WAIT FOR OUTPUT DATA REQUEST
  855   011642  004767  000532                   JSR     PC,P4SEND     ;SEND DATA TO PRINTER
  856   011646                                   WTDONE
   (1)  011646  032777  000200  006664           BIT     #200,@TCR
   (1)  011654  001774                           BEQ     .-6           ;WAIT FOR DONE FLAG
  857   011656                                   CKTSR   #0,LP         ;TEST FOR STATUS=0
```

```
(1)             011656
(1)  011656  012767  000000  006624          LP=.
(1)  011664  017767  006654  006606          MOV     #0,GOOD
(1)  011672  026767  006612  006600          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  011700  001401                           CMP     GOOD,BAD
(1)  011702  104003                           BEQ     .+4
(1)  011704                                   ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(2)  011724  000137  011656                   LOOP    LP
                                              JMP     @#LP
858  011730                                   CKTCR   #1214,LP                ;TEST FOR DONE + SPD + PRI
(1)             011730                        LP=.
(1)  011730  012767  001214  006552          MOV     #1214,GOOD
(1)  011736  017767  006576  006534          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  011744  026767  006540  006526          CMP     GOOD,BAD
(1)  011752  001401                           BEQ     .+4
(1)  011754  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  011756                                   LOOP    LP
(2)  011776  000137  011730                   JMP     @#LP
859
860  012002  000005                           RESET
861  012004  000207                           RTS     PC              ;EXIT CARD TEST
862
863                                   ;
                                      ;SUBROUTINE TO WAIT FOR CR
864                                   ;
865  012006  104400                   CONTIN: TYPE
866  012010  015307                           CONMSG                  ;'PRESS CR TO CONTINUE''
867  012012  004767  001666                   JSR     PC,CKSWR
868  012016  004767  001634           CONT1:  JSR     PC,TTYIN        ;GET A CHAR
869  012022  022704  000015                   CMP     #15,R4          ;WAS IT CR ?
870  012026  001373                           BNE     CONT1           ;NO - KEEP WAITING
871  012030  004767  001546                   JSR     PC,TTYOUT       ;YES - TYPE CR + LF
872  012034  000207                           RTS     PC
873
874                                   ;
                                      ;SUBROUTINE TO TEST FOR BLANK CARD (ALL SPACES)
875                                   ;
876  012036  012700  000001           BLKCRD: MOV     #1,R0
877  012042  004767  001636                   JSR     PC,CKSWR
878  012046  032777  000200  006500   BC1:    BIT     #200,@TIR
879  012054  001774                           BEQ     BC1             ;WAIT FOR INPUT DATA REQUEST
880  012056  012767  000040  006424          MOV     #40,GOOD
881  012064  017767  006460  006406          MOV     @TDR,BAD        ;READ DATA REGISTER
882  012072  026767  006412  006400          CMP     GOOD,BAD
883  012100  001401                           BEQ     .+4
884  012102  104007                           ERR+GB+COL              ;DATA REG - COMPARE ERROR AT INDICATED COLUMN
885  012104  005200                           INC     R0
886  012106  022700  000121                   CMP     #121,R0
887  012112  001355                           BNE     BC1             ;IF MORE COLUMNS TO READ GO DO IT
888  012114                                   CKTIR   #0,LP           ;TEST THAT IDR IS 0
(1)             012114                        LP=.
(1)  012114  012767  000000  006366          MOV     #0,GOOD
(1)  012122  017767  006426  006350          MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
(1)  012130  026767  006354  006342          CMP     GOOD,BAD
(1)  012136  001401                           BEQ     .+4
(1)  012140  104003                           ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO EXPECTED VALUE
(1)  012142                                   LOOP    LP
(2)  012162  000137  012114                   JMP     @#LP
889  012166  000207                           RTS     PC              ;EXIT
```

```
                                        ;
                                        ;PUT SPACE CHAR IN ALL LOCS BF20 - BF1+10
                                        ;
012170  012702  017466          CLRBFS: MOV     #BF20,R2
012174  012722  000040          CB1:    MOV     #40,(R2)+
012200  022702  017560                  CMP     #BF1+24,R2
012204  001373                          BNE     CB1
012206  000207                          RTS     PC
                                        ;
                                        ;PUT 1 CHAR (A) IN LOCS BF20 - BF1
                                        ;
012210  012702  017466          SETBFS: MOV     #BF20,R2
012214  012722  000101          SB1:    MOV     #101,(R2)+
012220  022702  017536                  CMP     #BF1+2,R2
012224  001373                          BNE     SB1
012226  000207                          RTS     PC
                                        ;
                                        ;SEND 80 CHARS BEGINNING AT LOC POINTED TO BY PTR
                                        ;
012230  016700  006274          P2SEND: MOV     PTR,R0
012234  004767  001444                  JSR     PC,CKSWR
012240  012701  000120                  MOV     #120,R1
012244  132777  000200  006304  P2SD1:  BITB    #200,@TIRHI
012252  001774                          BEQ     P2SD1           ;WAIT FOR OUTPUT DATA REQUEST
012254  011077  006270                  MOV     (R0),@TDR       ;SEND A CHAR
012260  062700  000002                  ADD     #2,R0
012264  005301                          DEC     R1              ;ALL 80 CHARS SENT ?
012266  001366                          BNE     P2SD1           ;NO - GO SEND ANOTHER ONE
012270  132777  000200  006260          BITB    #200,@TIRHI
012276  001401                          BEQ     .+4
012300  000000                          XX                      ;ERROR - GOT MORE THAN 80 ODR'S
012302  000207                          RTS     PC
                                        ;
                                        ;SEND 1 CHAR TO CARD #1 UP TO 20 CHARS TO CARD #20
                                        ;
012304  016767  006172  006224  P3SEND: MOV     CARD,TOG1
012312  012700  000001                  MOV     #1,R0
012316  004767  001362                  JSR     PC,CKSWR
012322  132777  000200  006226  P3SD1:  BITB    #200,@TIRHI
012330  001774                          BEQ     P3SD1           ;WAIT FOR OUTPUT DATA REQUEST
012332  012777  000101  006210          MOV     #101,@TDR       ;SEND "A" CHARACTER
012340  005200                          INC     R0
012342  005367  006170                  DEC     TOG1            ;HAVE ALL CHARS BEEN SENT
012346  001365                          BNE     P3SD1           ;NO - GO SEND ANOTHER ONE
012350  132777  000200  006200  P3SD2:  BITB    #200,@TIRHI
012356  001774                          BEQ     P3SD2           ;WAIT FOR OUTPUT DATA REQUEST
012360  012777  000040  006162          MOV     #40,@TDR        ;SEND SPACE CHAR
012366  005200                          INC     R0
012370  022700  000121                  CMP     #121,R0         ;HAVE ALL 80 CHARS BEEN SENT
012374  001365                          BNE     P3SD2           ;NO - GO SEND MORE
012376  000207                          RTS     PC
                                        ;
                                        ;SEND 7 CHARS TO BE PRINTED "CARD XX" AND THEN ALL SPACES
                                        ;
012400  005067  006132          P4SEND: CLR     TOG1
012404  004767  001274                  JSR     PC,CKSWR
```

```
 946  012410  132777  000200  006140  P4SD1:  BITB    #200,@TIRHI
 947  012416  001774                  BEQ     P4SD1           ;WAIT FOR OUTPUT DATA REQUEST
 948  012420  017777  006100  006122          MOV     @OUTPTR,@TDR    ;SEND A CHAR
 949  012426  062767  000002  006070          ADD     #2,OUTPTR
 950  012434  005267  006076                  INC     TOG1
 951  012440  022767  000007  006070          CMP     #7,TOG1         ;HAVE ALL PRINTING CHARS BEEN SENT
 952  012446  001360                  BNE     P4SD1           ;NO - GO PRINT MORE
 953  012450  132777  000200  006100  P4SD2:  BITB    #200,@TIRHI
 954  012456  001774                  BEQ     P4SD2           ;WAIT FOR OUTPUT DATA REQUEST
 955  012460  012777  000040  006062          MOV     #40,@TDR        ;SEND SPACE CHAR
 956  012466  005267  006044                  INC     TOG1
 957  012472  022767  000120  006036          CMP     #120,TOG1       ;HAVE ALL 80 CHARS BEEN SENT
 958  012500  001363                  BNE     P4SD2           ;NO - GO SEND MORE CHARS
 959  012502  000207                  RTS     PC
 960                          ;
 961                          ;COMPARE INPUT DATA PATTERN BEGINNING AT INPTR
 962                          ;
 963  012504  012700  000001  CKDATA: MOV     #1,R0
 964  012510  016767  006000  006012          MOV     INPTR,PTR
 965  012516  032777  000200  006030  CD1:    BIT     #200,@TIR
 966  012524  001774                  BEQ     CD1             ;WAIT FOR INPUT DATA REQUEST
 967  012526  017767  005776  005754          MOV     @PTR,GOOD
 968  012534  017767  006010  005736          MOV     @TDR,BAD        ;READ DATA REG
 969  012542  026767  005742  005730          CMP     GOOD,BAD
 970  012550  001401                  BEQ     .+4
 971  012552  104007                  ERR+GB+COL              ;DATA REG - COMPARE ERROR AT INDICATED COLUMN
 972  012554  005200                  INC     R0
 973  012556  062767  000002  005744          ADD     #2,PTR
 974  012564  022700  000121                  CMP     #121,R0
 975  012570  001352                  BNE     CD1             ;IF MORE COLUMNS TO READ GO DO IT
 976  012572  000207                  RTS     PC              ;EXIT
 977                          ;
 978                          ;CONVERT OCTAL # IN R0 TO DECIMAL # AND TYPE OUT DECIMAL #
 979                          ;
 980  012574  010067  000110  OCTDEC: MOV     R0,TOG
 981  012600  005067  000100          CLR     ONES
 982  012604  005067  000076          CLR     TENS
 983  012610  005267  000070  OD1:    INC     ONES
 984  012614  022767  000012  000062          CMP     #12,ONES
 985  012622  001404                  BEQ     OD2
 986  012624  005367  000060          DEC     TOG
 987  012630  001367                  BNE     OD1
 988  012632  000407                  BR      OD3
 989  012634  005067  000044  OD2:    CLR     ONES
 990  012640  005267  000042          INC     TENS
 991  012644  005367  000040          DEC     TOG
 992  012650  001357                  BNE     OD1
 993  012652  016704  000030  OD3:    MOV     TENS,R4
 994  012656  062704  000060          ADD     #60,R4
 995  012662  004767  000714          JSR     PC,TTYOUT       ;TYPEOUT TENS
 996  012666  016704  000012          MOV     ONES,R4
 997  012672  062704  000060          ADD     #60,R4
 998  012676  004767  000700          JSR     PC,TTYOUT       ;TYPEOUT ONES
 999  012702  000207                  RTS     PC
1000  012704  000000  ONES:   XX
1001  012706  000000  TENS:   XX
```

F 3

SEQ 0031

  1002 012710 000000   TOG: XX

```
1004
1005  012712  104400                ;INPUT ROUTINE FOR DEVICE ADDRESS, INT VECTOR ADDRESS AND BR LEVEL
1006  012714  014654          DEVCOD: TYPE
1007  012716  104400                  DCMSG1                  ;''THE FOLLOWING VALUES ARE BEING USED''
1008  012720  014722                  TYPE
1009  012722  104406                  DCMSG2                  ;''    DEVICE ADDRESS = ''
1010  012724  020540                  TYPE+6
1011  012726  104400                  TCR                     ;TYPE TCR ADDRESS
1012  012730  014751                  TYPE
1013  012732  104403                  DCMSG3                  ;''    PC INTERRUPT VECTOR ADDRESS = ''
1014  012734  020560                  TYPE+3
1015  012736  104400                  PCV                     ;TYPE PCV ADDRESS
1016  012740  015015                  TYPE
1017  012742  104401                  DCMSG4                  ;''    BUS REQUEST LEVEL = ''
1018  012744  020564                  TYPE+1
1019                                  BRLV                    ;TYPE BRLV
1020  012746  104400          ;TEST TO SEE IF VALUES ARE TO BE CHANGED
1021  012750  015047                  TYPE
1022  012752  004767  000700          DCMSG5                  ;''DO YOU WANT TO CHANGE ANY OF THESE VALUES (Y OR N)?''
1023  012756  022704  000131  DC1:    JSR     PC,TTYIN        ;INPUT A CHAR
1024  012762  001004                  CMP     #131,R4         ;IS IT Y ?
1025  012764  004767  000612          BNE     DC2             ;NO - GO CHECK FOR N
1026  012770  000167  000014          JSR     PC,TTYOUT       ;YES - ECHO Y
1027  012774  022704  000116          JMP     DC3             ;GO GET NEW VALUES
1028  013000  001364  DC2:    CMP     #116,R4         ;IS IT N ?
1029  013002  004767  000574          BNE     DC1             ;NO - GO BACK TO INPUT
1030  013006  000207                  JSR     PC,TTYOUT       ;YES - ECHO N
1031                                  RTS     PC              ;EXIT
1032  013010  104400          ;VALUES ARE TO BE CHANGED
1033  013012  015135          DC3:    TYPE
1034  013014  104400                  DCMSG6                  ;''  ENTER VALUE AFTER (=) IS TYPED''
1035  013016  014722                  TYPE
1036  013020  012705  000006          DCMSG2                  ;''    DEVICE ADDRESS = ''
1037  013024  004767  000464          MOV     #6,R5
1038  013030  042704  000007          JSR     PC,RDOCT        ;GET NEW DEVICE ADDRESS
1039  013034  012705  020540          BIC     #7,R4
1040  013040  042715  177770          MOV     #TCR,R5
1041  013044  050425          DC4:    BIC     #177770,(R5)
1042  013046  020527  020560          BIS     R4,(R5)+        ;STORE 4 NEW DEVICE ADDRESSES
1043  013052  001372                  CMP     R5,#PCV
1044  013054  104400                  BNE     DC4
1045  013056  014751                  TYPE
1046  013060  012705  000003          DCMSG3                  ;''    PC INTERRUPT VECTOR ADDRESS = ''
1047  013064  004767  000424          MOV     #3,R5
1048  013070  010467  005464          JSR     PC,RDOCT        ;GET NEW INT VECTOR ADDRESS
1049  013074  062704  000002          MOV     R4,PCV          ;SET NEW PC VECTOR ADDRESS
1050  013100  010467  005456          ADD     #2,R4
1051  013104  104400                  MOV     R4,PSV          ;SET NEW PS VECTOR ADDRESS
1052  013106  015015                  TYPE
1053  013110  012705  000001          DCMSG4                  ;''    BUS REQUEST LEVEL = ''
1054  013114  004767  000374          MOV     #1,R5
1055  013120  010467  005440          JSR     PC,RDOCT        ;GET NEW BR LEVEL
1056  013124  000207                  MOV     R4,BRLV         ;SET NEW BR LEVEL
1057                                  RTS     PC              ;EXIT
1058                          ;
1059                          ;ERROR SUBROUTINE
                              ;
```

```
1060  013126  004767  000552          ERROR:  JSR     PC,CKSWR
1061  013132  032777  020000  165340          BIT     #20000,@SWR
1062  013140  001403                          BEQ     .+10            ;IS LOOP ON ERROR SELECTED ?
1063  013142  012767  000001  005362          MOV     #1,SCOPE        ;YES - SET SCOPE TO A 1
1064  013150  011667  005342                  MOV     (SP),LASTPC     ;PUT ADDRESS OF ERROR IN LASTPC
1065  013154  032777  040000  165316          BIT     #40000,@SWR     ;IF SR14=1 DELETE TYPEOUT
1066  013162  001041                          BNE     E3              ;EXIT NOW
1067  013164  104400                          TYPE
1068  013166  015335                          PCMSG                   ;''ERROR AT ADDRESS''
1069  013170  011667  005302                  MOV     (SP),ADR
1070  013174  162767  000002  005274          SUB     #2,ADR
1071  013202  104406                          TYPE+6
1072  013204  020476                          ADR                     ;TYPE CONTENTS OF ADR
1073  013206  032777  000002  005262          BIT     #2,@ADR         ;TYPE GOOD ?
1074  013214  001404                          BEQ     E1              ;NO
1075  013216  104400                          TYPE
1076  013220  015361                          GDMSG                   ;''GOOD=''
1077  013222  104406                          TYPE+6
1078  013224  020510                          GOOD                    ;TYPE CONTENTS OF GOOD
1079  013226  032777  000001  005242  E1:     BIT     #1,@ADR         ;TYPE BAD ?
1080  013234  001404                          BEQ     E2              ;NO
1081  013236  104400                          TYPE
1082  013240  015370                          BDMSG                   ;''BAD =''
1083  013242  104406                          TYPE+6
1084  013244  020500                          BAD                     ;TYPE CONTENTS OF BAD
1085  013246  032777  000004  005222  E2:     BIT     #4,@ADR
1086  013254  001404                          BEQ     E3
1087  013256  104400                          TYPE
1088  013260  015377                          COLMSG                  ;''COLUMN #=''
1089  013262  004767  177306                  JSR     PC,OCTDEC       ;TYPEOUT DECIMAL COLUMN # IN ERROR
1090  013266  012704  000015          E3:     MOV     #15,R4
1091  013272  004767  000304                  JSR     PC,TTYOUT       ;TYPE CR + LF
1092  013276  032777  100000  165174          BIT     #100000,@SWR    ;IF SR15=1
1093  013304  001001                          BNE     E4              ;BYPASS THE ERROR HALT
1094  013306  000000          ERRHLT: XX                              ;ERROR HALT
1095  013310  000002          E4:     RTI                             ;EXIT
1096                           ;
1097                           ;DETERMINE IF THIS TYPEOUT IS FOR MESSAGE OR OCTAL VALUE AND BRANCH ACCORDINGLY
1098                           ;
1099  013312  011603          TYPOUT: MOV     (SP),R3         ;GET PC FROM STACK
1100  013314  162703  000002          SUB     #2,R3           ;SUBTRACT 2
1101  013320  032713  000007          BIT     #7,(R3)         ;TEST LOW ORDER 3 BITS OF TRAP INSTRUCTION
1102  013324  001401                  BEQ     TYPMSG          ;IF 0'S GO TO TYPMSG
1103  013326  000417                  BR      TYPOCT          ;IF NOT 0'S GO TO TYPOCT
1104                           ;
1105                           ;MESSAGE TYPEOUT - R5=ADDRESS OF BEGINNING OF MESSAGE
1106                           ;
1107  013330  011604          TYPMSG: MOV     (SP),R4
1108  013332  011405                  MOV     (R4),R5         ;R5=ADDRESS OF MESSAGE
1109  013334  112504          TM1:    MOVB    (R5)+,R4        ;PUT CHAR IN R4
1110  013336  001410                  BEQ     TM3             ;IF IT EQUALS 0 TERMINATE
1111  013340  022704  000046          CMP     #46,R4          ;DOES CHAR = CRLF FLAG ?
1112  013344  001002                  BNE     TM2             ;NO - GO TYPE CHAR
1113  013346  012704  000015          MOV     #15,R4          ;YES - TYPE CR+LF
1114  013352  004767  000224  TM2:    JSR     PC,TTYOUT       ;GO TYPE CONTENTS OF R4
1115  013356  000766                  BR      TM1             ;GO GET NEXT CHAR
```

```
1116  013360  062716  000002          TM3:    ADD     #2,(SP)         ;ADD 2 TO SP FOR RETURN
1117  013364  000002                          RTI                     ;EXIT
1118                                  ;
1119                                  ;OCTAL TYPEOUT - R4=VALUE  R5=# OF DIGITS TO TYPE
1120                                  ;
1121  013366  011305                  TYPOCT: MOV     (R3),R5
1122  013370  042705  177770                  BIC     #177770,R5      ;R5 = # OF DIGITS TO BE TYPED
1123  013374  062703  000002                  ADD     #2,R3
1124  013400  011367  005122                  MOV     (R3),PT
1125  013404  017704  005116                  MOV     @PT,R4          ;R4 = VALUE TO BE TYPED
1126  013410  010567  005070                  MOV     R5,CNT          ;SET CNT FOR STACKING
1127  013414  010467  005114          TO.1:   MOV     R4,TEMP         ;PUT VALUE IN TEMP
1128  013420  042767  177770  005106        BIC     #177770,TEMP    ;MASK OUT ALL BUT BITS 2,1+0
1129  013426  062767  000060  005100        ADD     #60,TEMP        ;MAKE ASCII NUMBER
1130  013434  016746  005074                MOV     TEMP,-(SP)      ;STORE NUMBER ON STACK
1131  013440  006004                        ROR     R4
1132  013442  006004                        ROR     R4
1133  013444  006004                        ROR     R4
1134  013446  005367  005032                DEC     CNT             ;IF CNT IS NOT ZERO
1135  013452  001360                        BNE     TO.1            ;GO STACK ANOTHER NUMBER
1136  013454  022705  000006                CMP     #6,R5
1137  013460  001002                        BNE     TO.2            ;IF 6 NUMBERS WERE SELECTED
1138  013462  042716  000006                BIC     #6,(SP)         ;CLEAR BITS 1+0 OF LAST NUMBER STACKED
1139  013466  010567  005012          TO.2: MOV     R5,CNT          ;SET CNT FOR TYPING
1140  013472  012604                  TO.3: MOV     (SP)+,R4        ;GET A NUMBER FROM THE STACK
1141  013474  004767  000102                JSR     PC,TTYOUT       ;TYPE THE NUMBER
1142  013500  005367  005000                DEC     CNT             ;IF COUNT IS NOT ZERO
1143  013504  001372                        BNE     TO.3            ;GO TYPE ANOTHER NUMBER
1144  013506  062716  000002                ADD     #2,(SP)         ;ADD 2 TO SP FOR RETURN
1145  013512  000002                        RTI                     ;EXIT
1146                                  ;
1147                                  ;OCTAL TYPEIN - R5=#OF DIGITS TO BE INPUT INTO R4
1148                                  ;
1149  013514  010567  004764          RDOCT:  MOV     R5,CNT          ;CNT=# OF DIGITS TO BE INPUT
1150  013520  005067  005010                  CLR     TEMP
1151  013524  004767  000126          RD.1:   JSR     PC,TTYIN        ;GET CHAR FROM TTY
1152  013530  004767  000046                  JSR     PC,TTYOUT       ;ECHO IT
1153  013534  042704  177770                  BIC     #177770,R4      ;MAKE CHAR OCTAL
1154  013540  006167  004770                  ROL     TEMP
1155  013544  006167  004764                  ROL     TEMP
1156  013550  006167  004760                  ROL     TEMP
1157  013554  042767  000007  004752        BIC     #7,TEMP         ;CLEAR TEMP BITS 2,1+0
1158  013562  060467  004746                  ADD     R4,TEMP         ;ADD INPUT CHAR TO OTHERS
1159  013566  005367  004712                  DEC     CNT             ;IF ALL CHARS NOT READ IN
1160  013572  001354                        BNE     RD.1            ;GO READ ANOTHER ONE
1161  013574  016704  004734                MOV     TEMP,R4         ;PUT FINAL OCTAL VALUE IN R4
1162  013600  000207                        RTS     PC              ;EXIT
1163                                  ;
1164                                  ;TYPE THE CHARACTER IN R4
1165                                  ;
1166  013602  032777  040000  164670  TTYOUT: BIT     #40000,@SWR     ;IF SR14=1 DELETE TYPEOUT
1167  013610  001021                        BNE     TTO.3
1168  013612  010467  163750                MOV     R4,TPB          ;TYPE CHAR IN R4
1169  013616  105767  163742          TTO.1:  TSTB    TPS
1170  013622  100375                        BPL     TTO.1           ;WAIT FOR DONE
1171  013624  022704  000015                CMP     #15,R4          ;WAS IT A CR ?
```

```
1172  013630  001003                          BNE     TTO.2           ;IF NO - GO EXIT
1173  013632  012704  000012                  MOV     #12,R4          ;IF YES - PUT LF IN R4
1174  013636  000761                          BR      TTYOUT          ;GO TYPE THE LF
1175  013640  012767  040000  004654  TTO.2:  MOV     #40000,LOC
1176  013646  005367  004650                  DEC     LOC
1177  013652  001375                          BNE     .-4
1178  013654  000207                  TTO.3:  RTS     PC              ;EXIT
1179                                  ;
1180                                  ;READ CHAR FROM TTY INTO R4
1181                                  ;
1182  013656  005267  163676          TTYIN:  INC     TKS             ;FETCH A CHAR
1183  013662  032767  000200  163670  TTI.1:  BIT     #200,TKS
1184  013670  001774                          BEQ     TTI.1           ;WAIT FOR DONE FLAG
1185  013672  016704  163664                  MOV     TKB,R4          ;READ CHAR INTO R4
1186  013676  042704  177600                  BIC     #177600,R4      ;MASK R4 WITH ALL BUT 177
1187  013702  000207                          RTS     PC              ;EXIT
1188                                  ;
1189                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
1190                                  ;*ROUTINE IS ENTERED FROM TRAP HANDLERR,AND WILL
1191                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
1192                                  ;*WHEN OPERATING IN TTY FLAG MODE.
1193  013704  022767  000176  164566  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT-SWR SELECTED?
1194  013712  001075                          BNE     K15             ;BRANCH IF NO
1195  013714  105767  163640                  TSTB    TKS             ;CHAR THERE?
1196  013720  100072                          BPL     K15             ;IF NO,DON'T WAIT AROUND
1197  013722  116746  163634                  MOVB    TKB,-(SP)       ;SAVE THE CHAR
1198  013726  042716  177600                  BIC     #^C177,(SP)     ;STRIP OFF THE ASCII
1199  013732  022726  000007                  CMP     #7,(SP)+        ;IS IT A CONTROL G?
1200  013736  001063                          BNE     K15             ;NO RETURN TO USER
1201  013740  126727  164546  000001          CMPB    AUTOB,#1        ;ARE WE RUNNING IN AUTO-MODE?
1202  013746  001457                          BEQ     K15             ;BRANCH IF YES
1203  013750  104400  014211                  TYPE    ,CNTLG          ;ECHO THE CONTROL-G (^G)
1204  013754  104400  014216  GTSWR:  TYPE    ,MSWR           ;TYPE CURRENT CONTENTS
1205  013760  016746  164212                  MOV     SWREG,-(SP)     ;SAVE SWREG FOR TYPEOUT
1206  013764  004767  000300                  JSR     PC,TYPOC        ;GO TYPE-OCTAL ASCII(ALL DIGITS)
1207  013770  104400  014226                  TYPE    ,MNEW           ;PROMPT FOR NEW SWR
1208  013774  005046          K19:    CLR     -(SP)           ;CLEAR COUNTER
1209  013776  005046                  CLR     -(SP)           ;NEW SWR
1210  014000  105767  163554          K7:     TSTB    TKS             ;CHAR THERE?
1211  014004  100375                          BPL     K7              ;IF NOT TRY AGAIN
1212  014006  116746  163550                  MOVB    TKB,-(SP)       ;PICK UP CHAR
1213  014012  042716  177600                  BIC     #^C177,(SP)     ;MAKE IT 7-BIT ASCII
1214  014016  021627  000025          K9:     CMP     (SP),#25        ;IT IS A CONTROL-U?
1215  014022  001005                          BNE     K10             ;BRANCH IF NOT
1216  014024  104400  014204                  TYPE    ,CNTLU          ;YES ECHO CONTROL - U(^U)
1217  014030  062706  000006          K20:    ADD     #6,SP           ;IGNORE PREVIOUS INPUT
1218  014034  000757                          BR      K19             ;LET'S TRY IT AGAIN
1219  014036  021627  000015          K10:    CMP     (SP),#15        ;IS IT A <CR>?
1220  014042  001022                          BNE     K16             ;BRANCH IF NO
1221  014044  005766  000004                  TST     4(SP)           ;YES,IS IT THE FIRST CHAR?
1222  014050  001403                          BEQ     K11             ;BRANCH IF YES
1223  014052  016677  000002  164420          MOV     2(SP),@SWR      ;SAVE NEW SWR
1224  014060  062706  000010          K11:    ADD     #10,SP          ;CLEAR UP STACK
1225                                  ;TTY INPUT ROUTINE
1226                                  ;
1227  014064  104400  014240          K14:    TYPE    ,CRLF           ;ECHO <CR> AND <LF>
```

```
1228  014070  126727  164420  000001                CMPB    INTAG,#1        ;RE-ENABLE TTY KBD INTERRUPTS
1229  014076  001003                                BNE     K15             ;BRANCH IF NOT
1230  014100  012767  000100  163452                MOV     #100,TKS       ;RE-ENABLE TTY KBD INTERRUPTS
1231  014106  000207                        K15:    RTS     PC              ;RETURN
1232  014110  004767  000356                K16:    JSR     PC,TYPEC        ;ECHO CHAR
1233  014114  021627  000060                        CMP     (SP),#60        ;CHAR < 0?
1234  014120  002420                                BLT     K18             ;BRANCH IF YES
1235  014122  021627  000067                        CMP     (SP),#67        ;CHAR > 7?
1236  014126  003015                                BGT     K18             ;BRANCH IF YES
1237  014130  042726  000060                        BIC     #60,(SP)+       ;STRIP-OFF ASCII
1238  014134  005766  000002                        TST     2(SP)           ;IS THIS THE FIRST CHAR
1239  014140  001403                                BEQ     K17             ;BRANCH IF YES
1240  014142  006316                                ASL     (SP)            ;NO SHIFT PRESENT
1241  014144  006316                                ASL     (SP)            ;CHAR OVER TO MAKE
1242  014146  006316                                ASL     (SP)            ;ROOM FOR NEW ONE
1243  014150  005266  000002                K17:    INC     2(SP)           ;KEEP COUNT OF CHARACTER
1244  014154  056616  177776                        BIS     -2(SP),(SP)     ;SET IN NEW CHAR
1245  014160  000707                                BR      K7              ;GET THE NEXT ONE
1246  014162  104400  014237                K18:    TYPE    ,QUES           ;TYPE ?<CR><LF>
1247  014166  000720                                BR      K20             ;SIMULATE CONTROL-U
1248  014170  000014                        TTYN:   .BLKB   12.              ;RESERVE 12 BYTE FOR TTYT
1249                                                                          ;Y INPUT
1250
1251  014204  052536  005015     000       CNTLU:  .ASCIZ  /^U/<15><12>    ;CONTROL 'U'
1252  014211     136  006507  000012       CNTLG:  .ASCIZ  /^G/<15><12>    ;CONTROL 'G'
1253  014216  005015  053523  036522       MSWR:   .ASCIZ  <15><12>/SWR= /
      014224  000040
1254
1255  014226  020040  042516  020127       MNEW:   .ASCIZ  /  NEW = /
      014234  020075     000
1256
1257  014237     077                        QUES:   .ASCII  /?/             ;QUESTION MARK
1258  014240     015                        CRLF:   .ASCII  <15>            ;CARRIAGE RETURN
1259  014241     012     000                LF:     .ASCIZ  <12>            ;LINE FEED
1260                                        ;*******************************************************
1261
1262                                        ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO
1263                                        ;*A 6-DIGIT OCTAL (ASCII) NUMBER AND TYPE IT
1264                                        ;*TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF
1265                                        ;*DIGITS TO TYPE
1266                                        ;*CALL:
1267                                        ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
1268                                        ;*      TYPOS                   ;CALL FOR TYPE OUT
1269                                        ;*      .BYTE   N               ;N=1 TO 6 FOR NUMBER OF DIGITS
1270                                        ;*TO TYPE
1271                                        ;*      .BYTE   M                       ;M=1  OR 0
1272                                                                        ;1=TYPE LEADING ZEROS
1273                                                                        ;0=SUPPRESS LEADING ZEROS
1274                                        ;*
1275                                        ;*TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS
1276                                        ;*THE LAST
1277                                        ;*TYPOS OR TYPOC
1278                                        ;*CALL:
1279                                        ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
1280                                        ;*      TYPON                   ;CALL FOR TYPEOUT
1281                                        ;*
```

```
1282                                                    ;*TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1283                                                    ;*CALL:
1284                                                    ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
1285                                                    ;*      TYPOC                   ;CALL FOR TYPEOUT
1286            014244                          .EVEN
1287    014244  017646  000000                  TYPOS: MOV     @(SP),-(SP)     ;PICKUP THE MODE
1288    014250  116667  000001  000211                MOVB    1(SP),FILL0     ;LOAD ZERO FILL SWITCH
1289    014256  112667  000207                        MOVB    (SP)+,MODE0+1   ;NUMBER OF DIGIT TO TYPE
1290    014262  062716  000002                        ADD     #2,(SP)          ;ADJUST RETURN ADDRESS
1291    014266  000406                                BR      TYPON
1292    014270  112767  000001  000171          TYPOC: MOVB    #1,FILL0         ;SET THE ZERO FILL SWITCH
1293    014276  112767  000006  000165                MOVB    #6,MODE0+1      ;SET FOR SIX(6) DIGITS
1294    014304  112767  000005  000154          TYPON: MOVB    #5,CNT0         ;SET THE ITERATION COUNT
1295    014312  010346                                MOV     R3,-(SP)        ;SAVR R3
1296    014314  010446                                MOV     R4,-(SP)        ;SAVE R4
1297    014316  010546                                MOV     R5,-(SP)        ;SAVE R5
1298    014320  116704  000145                        MOVB    MODE0+1,R4      ;GET THE NUMBER OF DIGIT TO TYPE
1299    014324  005404                                NEG     R4
1300    014326  062704  000006                        ADD     #6,R4           ;SUBSTRACT IT FOR MAX.ALLOWED
1301    014332  110467  000132                        MOVB    R4,MODE0        ;SAVE IT FOR USE
1302    014336  116704  000125                        MOVB    FILL0,R4        ;GET THE ZERO FILL SWITCH
1303    014342  016605  000010                        MOV     10(SP),R5       ;PICK UP THE INPUT NUMBER
1304    014346  005003                                CLR     R3              ;CLEAR THE OUTPUT WORD
1305    014350  006105                          A1:   ROL     R5              ;ROTATE MSB INTO ''C''
1306    014352  000404                                BR      A3              ;GO TO MSB
1307    014354  006105                          A2:   ROL     R5              ;FORM THIS DIGIT
1308    014356  006105                                ROL     R5
1309    014360  006105                                ROL     R5
1310    014362  010503                                MOV     R5,R3
1311    014364  006103                          A3:   ROL     R3               ;GET LSB OF THIS DIGIT
1312    014366  105367  000076                        DECB    MODE0            ;TYPE THIS DIGIT?
1313    014372  100022                                BPL     A7              ;BR IF NO
1314    014374  042703  177770                        BIC     #177770,R3       ;GET RID OF JUNK
1315    014400  001002                                BNE     A4              ;TEST FOR 0
1316    014402  005704                                TST     R4                      ;SUPPRESS THIS 0?
1317    014404  001403                                BEQ     A5              ;BR IF YES
1318    014406  005204                          A4:   INC     R4              ;DON'T SUPPRESS ANYMORE 0'S
1319    014410  052703  000060                        BIS     #'0,R3          ;MAKE THIS DIGIT ASCII
1320    014414  052703  000040                  A5:   BIS     #' ,R3          ;MAKE ASCII IF NOT ALREA
1321    014420  110367  000040                        MOVB    R3,A8            ;SAVE FOR TYPING
1322    014424  010446                                MOV     R4,-(SP)
1323    014426  010546                                MOV     R5,-(SP)         ;
1324    014430  104400  014464                        TYPE    ,A8              ;GO TYPE THIS DIGIT
1325    014434  012605                                MOV     (SP)+,R5
1326    014436  012604                                MOV     (SP)+,R4
1327    014440  105367  000022                  A7:   DECB    CNT0             ;COUNT BY ONE
1328    014444  003343                                BGT     A2              ;BRANCH IF MORE TO DO
1329    014446  002402                                BLT     A6              ;BRANCH IF DONE
1330    014450  005204                                INC     R4              ;INSURE LAST DIGIT IS NOT A BLANK
1331    014452  000740                                BR      A2
1332    014454  012605                          A6:   MOV     (SP)+,R5        ;RESTORE R5
1333    014456  012604                                MOV     (SP)+,R4        ;RESTORE R4
1334    014460  012603                                MOV     (SP)+,R3        ;RESTORE R3
1335    014462  000207                                RTS     PC
1336    014464     000                  A8:   .BYTE    0               ;STORAGE FOR ASCII DIGIT
1337    014465     000                        .BYTE    0               ;TERMINATOR FOR TYPE ROUTINE
```

```
1338  014466  000
1339  014467  000
1340  014470  000000
1341
1342
1343
1344  014472  105767  163066
1345  014476  100375
1346  014500  116667  000002  163060
1347  014506  122766  000015  000002
1348  014514  001003
1349  014516  105067  000014
1350  014522  000406
1351  014524  122766  014241  000002
1352  014532  001402
1353  014534  105227
1354  014536  000000
1355  014540  000207
1356
1357
1358
1359  014542  013746  000004
1360  014546  012737  014602  000004
1361  014554  012767  177570  163716
1362  014562  012767  177570  163712
1363  014570  022777  177777  163702
1364  014576  001012
1365
1366  014600  000403
1367  014602  012716  014610
1368  014606  000002
1369  014610  012767  000176  163662
1370  014616  012767  000174  163656
1371  014624  012637  000004
1372  014630  005067  163650
1373  014634  132767  000200  163644
1374  014642  001403
1375  014644  012767  000510  163626
1376  014652  000207
1377
1378  014654  023046  044124  020105  DCMSG1: .ASCIZ /&&THE FOLLOWING VALUES ARE BEING USED/
      014662  047506  046114  053517
      014670  047111  020107  040526
      014676  052514  051505  040440
      014704  042522  041040  044505
      014712  043516  052440  042523
      014720  000104
1379  014722  020046  020040  042040  DCMSG2: .ASCIZ  /&    DEVICE ADDRESS = /
      014730  053105  041511  020105
      014736  042101  051104  051505
      014744  020123  020075     000
1380  014751     046  020040  020040  DCMSG3: .ASCIZ  /&    PC INTERRUPT VECTOR ADDRESS = /
      014756  041520  044440  052116
      014764  051105  052522  052120
      014772  053040  041505  047524
      015000  020122  042101  051104

CNTO:    .BYTE   0              ;OCTAL DIGIT COUNTER
FILLO:   .BYTE   0              ;ZERO FILL SWITCH
MODEO:   .WORD   0              ;NUMBER OF DIGIT TO TYPE
;
;
;
TYPEC:   TSTB    TPS            ;WAIT UNTIL THE PRINTER IS READY
         BPL     TYPEC
         MOVB    2(SP),TPB      ;LOAD CHARACTER TO BE TYPED INTO DATA REG
         CMPB    #15,2(SP)       ;IS CHAR CARRIAGE RETURN?
         BNE     B1
         CLRB    CHARCNT        ;YES CLEAR CHAR COUNT
         BR      TYPEX          ;EXIT
B1:      CMPB    #LF,2(SP)      ;IS CHAR LINE FEED?
         BEQ     TYPEX          ;BRANCH IF YES
         INCB    (PC)+          ;COUNT THE CHARS
CHARCNT:.WORD    0              ;CHARACTER COUNT SPACE
TYPEX:   RTS     PC

;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
;EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
SWRCK:   MOV     @#ERRVEC,-(SP)  ;SAVE ERROR VECTOR
         MOV     #RT1,@#ERRVEC    ;SET UP ERROR VECTOR
         MOV     #DSWR,SWR       ;SET UP FOR HARDWARE SWR
         MOV     #DDISP,DISPLAY  ;AND A HARDWARE DISPLAY REGISTER
         CMP     #-1,@SWR        ;TRY TO REFERENCE HARDWARE SWR
         BNE     RT3             ;BRANCH IF NO TIME OUT TRAP OCCURED
                                 ;AND THE HARDWARE SWR IS NOT = -1
         BR      RT2             ;BRANCH IF NO TIME OUT
RT1:     MOV     #RT2,(SP)       ;SET UP FOR TRAP RETURN
         RTI
RT2:     MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWR
         MOV     #DISPREG,DISPLAY
RT3:     MOV     (SP)+,@#ERRVEC  ;RESTORE ERROR VECTOR
         CLR     APASS           ;CLEAR PASS COUNT
         BITB    #200,AENVM      ;TEST USER SIZE UNDER APT
         BEQ     RT4             ;YES,USE NON-APT SWITCH
         MOV     #ASWREG,SWR     ;NO,USE APT SWITCH REGISTER
RT4:     RTS     PC
```

```
                015006  051505  020123  020075
                015014     000
1381            015015     046  020040  020040  DCMSG4: .ASCIZ  /&    BUS REQUEST LEVEL = /
                015022  052502  020123  042522
                015030  052521  051505  020124
                015036  042514  042526  020114
                015044  020075     000
1382            015047     046  042046  020117  DCMSG5: .ASCIZ  /&&DO YOU WANT TO CHANGE ANY OF THESE VALUES (Y OR N)?/
                015054  047531  020125  040527
                015062  052116  052040  020117
                015070  044103  047101  042507
                015076  040440  054516  047440
                015104  020106  044124  051505
                015112  020105  040526  052514
                015120  051505  024040  020131
                015126  051117  047040  037451
                015134     000
1383            015135     046  020040  047105  DCMSG6: .ASCIZ  /&  ENTER VALUE AFTER (=) IS TYPED/
                015142  042524  020122  040526
                015150  052514  020105  043101
                015156  042524  020122  036450
                015164  020051  051511  052040
                015172  050131  042105     000
1384            015177     046  025046  025052  LOGTST: .ASCIZ  /&&***** LOGIC TESTS *****&&/
                015204  025052  046040  043517
                015212  041511  052040  051505
                015220  051524  025040  025052
                015226  025052  023046     000
1385            015233     046  050046  052125  INITM1: .ASCIZ  /&&PUT 8035-8045  OFF-LINE, REMOVE ALL CARDS/
                015240  034040  031460  026465
                015246  030070  032464  020040
                015254  043117  026506  044514
                015262  042516  020054  042522
                015270  047515  042526  040440
                015276  046114  041440  051101
                015304  051504     000
1386            015307     046  051120  051505  CONMSG: .ASCIZ  /&PRESS CR TO CONTINUE/
                015314  020123  051103  052040
                015322  020117  047503  052116
                015330  047111  042525     000
1387            015335     046  042446  051122  PCMSG:  .ASCIZ  /&&ERROR AT ADDRESS /
                015342  051117  040440  020124
                015350  042101  051104  051505
                015356  020123     000
1388            015361     046  047507  042117  GDMSG:  .ASCIZ  /&GOOD=/
                015366  000075
1389            015370  041046  042101  036440  BDMSG:  .ASCIZ  /&BAD =/
                015376     000
1390            015377     046  047503  052514  COLMSG: .ASCIZ  /&COLUMN #=/
                015404  047115  021440  000075
1391            015412  023046  052520  020124  T2MSG1: .ASCIZ  /&&PUT 8035-8045  ON-LINE AND PRESS START ON THE 8035-8045/
                015420  030070  032463  034055
                015426  032060  020065  047440
                015434  026516  044514  042516
                015442  040440  042116  050040
                015450  042522  051523  051440
```

```
                015456  040524  052122  047440
                015464  020116  044124  020105
                015472  030070  032463  034055
                015500  032060  000065
 1392           015504  023046  052520  020124  T5M1:   .ASCIZ  /&&PUT 8035-8045  OFF-LINE/
                015512  030070  032463  034055
                015520  032060  020065  047440
                015526  043106  046055  047111
                015534  000105
 1393           015536  023046  051120  051505  T5M2:   .ASCIZ  /&&PRESS START ON THE 8035-8045 /
                015544  020123  052123  051101
                015552  020124  047117  052040
                015560  042510  034040  031460
                015566  026465  030070  032464
                015574  000040
 1394           015576  023046  052520  046114  T5MSG1: .ASCIZ  /&&PULL CARD PLATE OF STACKER #1 BACK TO POSITION WHICH/
                015604  041440  051101  020104
                015612  046120  052101  020105
                015620  043117  051440  040524
                015626  045503  051105  021440
                015634  020061  040502  045503
                015642  052040  020117  047520
                015650  044523  044524  047117
                015656  053440  044510  044103
                015664     000
 1395           015665     046  044527  046114  T5MSG2: .ASCIZ  /&WILL SIMULATE STACKER FULL AND RELEASE/
                015672  051440  046511  046125
                015700  052101  020105  052123
                015706  041501  042513  020122
                015714  052506  046114  040440
                015722  042116  051040  046105
                015730  040505  042523     000
 1396           015735     046  051120  043517  T5MSG3: .ASCIZ  /&PROGRAM WILL WAIT FOR STACKER FULL FLAG/
                015742  040522  020115  044527
                015750  046114  053440  044501
                015756  020124  047506  020122
                015764  052123  041501  042513
                015772  020122  052506  046114
                016000  043040  040514  000107
 1397           016006  023046  051120  051505  T5MSG4: .ASCIZ  /&&PRESS START ON THE 8035-8045 /
                016014  020123  052123  051101
                016022  020124  047117  052040
                016030  042510  034040  031460
                016036  026465  030070  032464
                016044  000040
 1398           016046  023046  052520  046114  T5MSG5: .ASCIZ  /&&PULL CARD PLATE OF STACKER #2 BACK TO POSITION WHICH/
                016054  041440  051101  020104
                016062  046120  052101  020105
                016070  043117  051440  040524
                016076  045503  051105  021440
                016104  020062  040502  045503
                016112  052040  020117  047520
                016120  044523  044524  047117
                016126  053440  044510  044103
                016134     000
 1399           016135     046  050046  042522  T5MSG6: .ASCIZ  /&&PRESS START ON THE 8035-8045 /
```

```
                016142  051523  051440  040524
                016150  052122  047440  020116
                016156  044124  020105  030070
                016164  032463  034055  032060
                016172  020065     000
1400            016175     046  050046  042522  T5MSG7: .ASCIZ  /&&PRESS START ON THE 8035-8045 /
                016202  051523  051440  040524
                016210  052122  047440  020116
                016216  044124  020105  030070
                016224  032463  034055  032060
                016232  020065     000
1401            016235     046  051120  051505  T6MSG1: .ASCIZ  /&PRESS START ON THE 8035-8045 /
                016242  020123  052123  051101
                016250  020124  047117  052040
                016256  042510  034040  031460
                016264  026465  030070  032464
                016272  000040
1402            016274  023046  025052  025052  CRDTST: .ASCIZ  /&&***** CARD TESTS *****&&/
                016302  020052  040503  042122
                016310  052040  051505  051524
                016316  025040  025052  025052
                016324  023046     000
1403            016327     046  051046  046505  P1MSG1: .ASCIZ  /&&REMOVE ALL CARDS AND PUT THE 8035-8045  OFF-LINE/
                016334  053117  020105  046101
                016342  020114  040503  042122
                016350  020123  047101  020104
                016356  052520  020124  044124
                016364  020105  030070  032463
                016372  034055  032060  020065
                016400  047440  043106  046055
                016406  047111  000105
1404            016412  023046  052520  020124  P1MSG2: .ASCIZ  /&&PUT 10 BLANK CARDS IN EACH INPUT HOPPER/
                016420  030061  041040  040514
                016426  045516  041440  051101
                016434  051504  044440  020116
                016442  040505  044103  044440
                016450  050116  052125  044040
                016456  050117  042520  000122
1405            016464  050046  052125  034040  P1MSG3: .ASCIZ  /&PUT 8035-8045  ON-LINE AND PRESS START ON THE 8035-8045/
                016472  031460  026465  030070
                016500  032464  020040  047117
                016506  046055  047111  020105
                016514  047101  020104  051120
                016522  051505  020123  052123
                016530  051101  020124  047117
                016536  052040  042510  034040
                016544  031460  026465  030070
                016552  032464     000
1406            016555     046  042526  044522  P2MSG1: .ASCIZ  /&VERIFY THAT THERE ARE 10 BLANK CARDS IN EACH STACKER/
                016562  054506  052040  040510
                016570  020124  044124  051105
                016576  020105  051101  020105
                016604  030061  041040  040514
                016612  045516  041440  051101
                016620  051504  044440  020116
                016626  040505  044103  051440
```

```
                016634  040524  045503  051105
                016642     000
1407            016643     046  046120  041501  P2MSG2: .ASCIZ  /&PLACE ALL 20 CARDS IN THE PRIMARY HOPPER/
                016650  020105  046101  020114
                016656  030062  041440  051101
                016664  051504  044440  020116
                016672  044124  020105  051120
                016700  046511  051101  020131
                016706  047510  050120  051105
                016714     000
1408            016715     046  051120  051505  P2MSG3: .ASCIZ  /&PRESS STOP-RESET AND PRESS START ON THE 8035-8045 /
                016722  020123  052123  050117
                016730  051055  051505  052105
                016736  040440  042116  050040
                016744  042522  051523  051440
                016752  040524  052122  047440
                016760  020116  044124  020105
                016766  030070  032463  034055
                016774  032060  020065     000
1409            017001     046  051120  051505  P2MSG4: .ASCIZ  /&PRESS STOP-RESET ON THE 8035-8045/
                017006  020123  052123  050117
                017014  051055  051505  052105
                017022  047440  020116  044124
                017030  020105  030070  032463
                017036  034055  032060  000065
1410            017044  053046  051105  043111  P3MSG1: .ASCIZ  /&VERIFY THAT ALL CHARS WERE PUNCHED ON EACH CARD IN A/
                017052  020131  044124  052101
                017060  040440  046114  041440
                017066  040510  051522  053440
                017074  051105  020105  052520
                017102  041516  042510  020104
                017110  047117  042440  041501
                017116  020110  040503  042122
                017124  044440  020116  000101
1411            017132  050046  042522  042503  P3MSG2: .ASCIZ  /&PRECESS FASHION BEGINNING WITH COLUMN 11 OF 1ST CARD/
                017140  051523  043040  051501
                017146  044510  047117  041040
                017154  043505  047111  044516
                017162  043516  053440  052111
                017170  020110  047503  052514
                017176  047115  030440  020061
                017204  043117  030440  052123
                017212  041440  051101  000104
1412            017220  053046  051105  043111  P4MSG1: .ASCIZ  /&VERIFY THAT THERE IS 1 ''A'' CHAR PUNCHED IN CARD 1/
                017226  020131  044124  052101
                017234  052040  042510  042522
                017242  044440  020123  020061
                017250  040442  020042  044103
                017256  051101  050040  047125
                017264  044103  042105  044440
                017272  020116  040503  042122
                017300  030440     000
1413            017303     046  025052  020052  PROGNM:.ASCIZ /&*** DECSPEC-11-CZCTAA0-CTS11-JC-7 BIT ASCII-TEST ****&/
                017310  042504  051503  042520
                017316  026503  030461  041455
                017324  041532  040524  030101
```

```
                017332  041455  051524  030461
                017340  045055  026503  020067
                017346  044502  020124  051501
                017354  044503  026511  042524
                017362  052123  025040  025052
                017370  023052     000
1414            017373     046  050125  052040  P4MSG2: .ASCIZ  /&UP TO 20 ''A'' CHARS PUNCHED IN CARD 20/
                017400  020117  030062  021040
                017406  021101  041440  040510
                017414  051522  050040  047125
                017422  044103  042105  044440
                017430  020116  040503  042122
                017436  031040  000060
1415            017442  023046  047105  020104  ENDMSG: .ASCIZ  /&&END OF TESTING&&/
                017450  043117  052040  051505
                017456  044524  043516  023046
                017464     000
1416                    017466                       .EVEN
1417            017466  000040               BF20:   40
1418            017470  000040               BF19:   40
1419            017472  000040               BF18:   40
1420            017474  000040               BF17:   40
1421            017476  000040               BF16:   40
1422            017500  000040               BF15:   40
1423            017502  000040               BF14:   40
1424            017504  000040               BF13:   40
1425            017506  000040               BF12:   40
1426            017510  000040               BF11:   40
1427            017512  000040               BF10:   40
1428            017514  000040               BF9:    40
1429            017516  000040               BF8:    40
1430            017520  000040               BF7:    40
1431            017522  000040               BF6:    40
1432            017524  000040               BF5:    40
1433            017526  000040               BF4:    40
1434            017530  000040               BF3:    40
1435            017532  000040               BF2:    40
1436            017534  000040               BF1:    40
1437            017536  000040                       40
1438            017540  000040                       40
1439            017542  000040                       40
1440            017544  000040                       40
1441            017546  000040                       40
1442            017550  000040                       40
1443            017552  000040                       40
1444            017554  000040                       40
1445            017556  000040                       40
1446            017560  000041                       41
1447            017562  000042                       42
1448            017564  000043                       43
1449            017566  000044                       44
1450            017570  000045                       45
1451            017572  000046                       46
1452            017574  000047                       47
1453            017576  000050                       50
1454            017600  000051                       51
```

```
 1455  017602  000052                          52
 1456  017604  000053                          53
 1457  017606  000054                          54
 1458  017610  000055                          55
 1459  017612  000056                          56
 1460  017614  000057                          57
 1461  017616  000060                          60
 1462  017620  000061                          61
 1463  017622  000062                          62
 1464  017624  000063                          63
 1465  017626  000064                          64
 1466  017630  000065                          65
 1467  017632  000066                          66
 1468  017634  000067                          67
 1469  017636  000070                          70
 1470  017640  000071                          71
 1471  017642  000072                          72
 1472  017644  000073                          73
 1473  017646  000074                          74
 1474  017650  000075                          75
 1475  017652  000076                          76
 1476  017654  000077                          77
 1477  017656  000100                         100
 1478  017660  000101                         101
 1479  017662  000102                         102
 1480  017664  000103                         103
 1481  017666  000104                         104
 1482  017670  000105                         105
 1483  017672  000106                         106
 1484  017674  000107                         107
 1485  017676  000110                         110
 1486  017700  000111                         111
 1487  017702  000112                         112
 1488  017704  000113                         113
 1489  017706  000114                         114
 1490  017710  000115                         115
 1491  017712  000116                         116
 1492  017714  000117                         117
 1493  017716  000120                         120
 1494  017720  000121                         121
 1495  017722  000122                         122
 1496  017724  000123                         123
 1497  017726  000124                         124
 1498  017730  000125                         125
 1499  017732  000126                         126
 1500  017734  000127                         127
 1501  017736  000130                         130
 1502  017740  000131                         131
 1503  017742  000132                         132
 1504  017744  000133                         133
 1505  017746  000134                         134
 1506  017750  000135                         135
 1507  017752  000136                         136
 1508  017754  000137                         137
 1509  017756  000173                         173
 1510  017760  000175
```

```
 1511  017762  000040                          40
 1512  017764  000040                          40
 1513  017766  000040                          40
 1514  017770  000040                          40
 1515  017772  000040                          40
 1516  017774  000040                          40
 1517  017776  000040                          40
 1518  020000  000040                          40
 1519  020002  000040                          40
 1520  020004  000040                          40
 1521  020006  000040                          40
 1522  020010  000040                          40
 1523  020012  000040                          40
 1524  020014  000040                          40
 1525  020016  000040                          40
 1526  020020  000040                          40
 1527  020022  000040                          40
 1528  020024  000040                          40
 1529  020026  000040                          40
 1530  020030  000040                          40
 1531  020032  000040                          40
 1532  020034  000040                          40
 1533  020036  000040                          40
 1534  020040  000040                          40
 1535  020042  000040                          40
 1536  020044  000040                          40
 1537  020046  000103              CARD01:     103
 1538  020050  000101                          101
 1539  020052  000122                          122
 1540  020054  000104                          104
 1541  020056  000040                          40
 1542  020060  000060                          60
 1543  020062  000061                          61
 1544  020064  000103              CARD02:     103
 1545  020066  000101                          101
 1546  020070  000122                          122
 1547  020072  000104                          104
 1548  020074  000040                          40
 1549  020076  000060                          60
 1550  020100  000062                          62
 1551  020102  000103              CARD03:     103
 1552  020104  000101                          101
 1553  020106  000122                          122
 1554  020110  000104                          104
 1555  020112  000040                          40
 1556  020114  000060                          60
 1557  020116  000063                          63
 1558  020120  000103              CARD04:     103
 1559  020122  000101                          101
 1560  020124  000122                          122
 1561  020126  000104                          104
 1562  020130  000040                          40
 1563  020132  000060                          60
 1564  020134  000064                          64
 1565  020136  000103              CARD05:     103
 1566  020140  000101                          101
```

```
 1567  020142  000122                          122
 1568  020144  000104                          104
 1569  020146  000040                          40
 1570  020150  000060                          60
 1571  020152  000065                          65
 1572  020154  000103                CARD06:   103
 1573  020156  000101                          101
 1574  020160  000122                          122
 1575  020162  000104                          104
 1576  020164  000040                          40
 1577  020166  000060                          60
 1578  020170  000066                          66
 1579  020172  000103                CARD07:   103
 1580  020174  000101                          101
 1581  020176  000122                          122
 1582  020200  000104                          104
 1583  020202  000040                          40
 1584  020204  000060                          60
 1585  020206  000067                          67
 1586  020210  000103                CARD08:   103
 1587  020212  000101                          101
 1588  020214  000122                          122
 1589  020216  000104                          104
 1590  020220  000040                          40
 1591  020222  000060                          60
 1592  020224  000070                          70
 1593  020226  000103                CARD09:   103
 1594  020230  000101                          101
 1595  020232  000122                          122
 1596  020234  000104                          104
 1597  020236  000040                          40
 1598  020240  000060                          60
 1599  020242  000071                          71
 1600  020244  000103                CARD10:   103
 1601  020246  000101                          101
 1602  020250  000122                          122
 1603  020252  000104                          104
 1604  020254  000040                          40
 1605  020256  000061                          61
 1606  020260  000060                          60
 1607  020262  000103                CARD11:   103
 1608  020264  000101                          101
 1609  020266  000122                          122
 1610  020270  000104                          104
 1611  020272  000040                          40
 1612  020274  000061                          61
 1613  020276  000061                          61
 1614  020300  000103                CARD12:   103
 1615  020302  000101                          101
 1616  020304  000122                          122
 1617  020306  000104                          104
 1618  020310  000040                          40
 1619  020312  000061                          61
 1620  020314  000062                          62
 1621  020316  000103                CARD13:   103
 1622  020320  000101                          101
```

```
 1623  020322  000122                          122
 1624  020324  000104                          104
 1625  020326  000040                          40
 1626  020330  000061                          61
 1627  020332  000063                          63
 1628  020334  000103                CARD14:   103
 1629  020336  000101                          101
 1630  020340  000122                          122
 1631  020342  000104                          104
 1632  020344  000040                          40
 1633  020346  000061                          61
 1634  020350  000064                          64
 1635  020352  000103                CARD15:   103
 1636  020354  000101                          101
 1637  020356  000122                          122
 1638  020360  000104                          104
 1639  020362  000040                          40
 1640  020364  000061                          61
 1641  020366  000065                          65
 1642  020370  000103                CARD16:   103
 1643  020372  000101                          101
 1644  020374  000122                          122
 1645  020376  000104                          104
 1646  020400  000040                          40
 1647  020402  000061                          61
 1648  020404  000066                          66
 1649  020406  000103                CARD17:   103
 1650  020410  000101                          101
 1651  020412  000122                          122
 1652  020414  000104                          104
 1653  020416  000040                          40
 1654  020420  000061                          61
 1655  020422  000067                          67
 1656  020424  000103                CARD18:   103
 1657  020426  000101                          101
 1658  020430  000122                          122
 1659  020432  000104                          104
 1660  020434  000040                          40
 1661  020436  000061                          61
 1662  020440  000070                          70
 1663  020442  000103                CARD19:   103
 1664  020444  000101                          101
 1665  020446  000122                          122
 1666  020450  000104                          104
 1667  020452  000040                          40
 1668  020454  000061                          61
 1669  020456  000071                          71
 1670  020460  000103                CARD20:   103
 1671  020462  000101                          101
 1672  020464  000122                          122
 1673  020466  000104                          104
 1674  020470  000040                          40
 1675  020472  000062                          62
 1676  020474  000060                          60
 1677                                          .EVEN
 1678
```

```
1679  020476  000000                  ADR:    XX
1680  020500  000000                  BAD:    XX
1681  020502  000000                  CARD:   XX
1682  020504  000000                  CNT:    XX
1683  020506  000000                  CNT1:   XX
1684  020510  000000                  GOOD:   XX
1685  020512  000000                  HOLD:   XX
1686  020514  000000                  INPTR:  XX
1687  020516  000000                  LASTPC: XX
1688  020520  000000                  LEVEL:  XX
1689  020522  000000                  LOC:    XX
1690  020524  000000                  OUTPTR: XX
1691  020526  000000                  PT:     XX
1692  020530  000000                  PTR:    XX
1693  020532  000000                  SCOPE:  XX
1694  020534  000000                  TEMP:   XX
1695  020536  000000                  TOG1:   XX
1696
1697                                  ;
1698                                  ;******** DEVICE ADDRESSES ********
1699                                  ;
1700  020540  177160                  TCR:    177160
1701  020542  177161                  TCRHI:  177161
1702  020544  177162                  TSR:    177162
1703  020546  177163                  TSRHI:  177163
1704  020550  177164                  TDR:    177164
1705  020552  177165                  TDRHI:  177165
1706  020554  177166                  TIR:    177166
1707  020556  177167                  TIRHI:  177167
1708                                  ;
1709                                  ;******** INTERRUPT VECTOR ADDRESSES ********
1710                                  ;
1711  020560  000230                  PCV:    230
1712  020562  000232                  PSV:    232
1713                                  ;
1714                                  ;******** BUS REQUEST LEVEL ********
1715                                  ;
1716  020564  000004                  BRLV:   4
1717
1718          000001                          .END
```

```
ADR     020476  1069* 1070* 1072  1073  1079  1085  1679#
AENVM   000506   334# 1373
APASS   000504   333# 1372*
ASWREG  000510   335# 1375
AUTOB   000512   336# 1201
A1      014350  1305#
A2      014354  1307# 1328  1331
A3      014364  1306  1311#
A4      014406  1315  1318#
A5      014414  1317  1320#
A6      014454  1329  1332#
A7      014440  1313  1327#
A8      014464  1321* 1324  1336#
B     = 000001   244#
BAD     020500   378*  379*  380*  389*  395*  419*  420*  421*  444*  445   450*  460*  461*
                 464*  467*  468*  481*  484*  485*  509*  510*  515*  516*  527*  532*  557*
                 558*  579*  605*  608*  614*  615*  629*  630*  635*  637*  644*  645*  649*
                 651*  658*  659*  666*  667*  671*  672*  694*  695*  701*  705*  706*  707*
                 718*  719*  723*  724*  739*  740*  765*  766*  779*  780*  781*  792*  793*
                 799*  800*  825*  826*  837*  838*  839*  850*  851*  857*  858*  881*  882
                 888*  968*  969  1084  1680#
BC1     012046   878#  879   887
BDMSG   015370  1082  1389#
BF1     017534   697   759   819   895   903  1436#
BF10    017512  1427#
BF11    017510  1426#
BF12    017506  1425#
BF13    017504  1424#
BF14    017502  1423#
BF15    017500  1422#
BF16    017476  1421#
BF17    017474  1420#
BF18    017472  1419#
BF19    017470  1418#
BF2     017532  1435#
BF20    017466   893   901  1417#
BF3     017530  1434#
BF4     017526  1433#
BF5     017524  1432#
BF6     017522  1431#
BF7     017520  1430#
BF8     017516  1429#
BF9     017514  1428#
BLKCRD  012036   628   636   643   650   656   665   692   716   876#
BRLV    020564   443  1018  1055* 1716#
B1      014524  1348  1351#
CARD    020502   770*  774*  788*  925  1681#
CARD01  020046   829  1537#
CARD02  020064  1544#
CARD03  020102  1551#
CARD04  020120  1558#
CARD05  020136  1565#
CARD06  020154  1572#
CARD07  020172  1579#
CARD08  020210  1586#
CARD09  020226  1593#
```

```
CARD10  020244          1600#
CARD11  020262          1607#
CARD12  020300          1614#
CARD13  020316          1621#
CARD14  020334          1628#
CARD15  020352          1635#
CARD16  020370          1642#
CARD17  020406          1649#
CARD18  020424          1656#
CARD19  020442          1663#
CARD20  020460          1670#
CB1     012174           894#   896
CD1     012516           965#   966    975
CHARCN  014536          1349*  1354#
CKDATA  012504           762    776    790    822    834    847    963#
CKSWR   013704           595    867    877    910    927    945   1060   1193#
CLRBFS  012170           688    893#
CNT     020504           570*   571*   573*   575*   577*   609*   610*   612*  1126*  1134*  1139*  1142*  1149*
                        1159*  1682#
CNTLG   014211          1203   1252#
CNTLU   014204          1216   1251#
CNT0    014466          1294*  1327*  1338#
CNT1    020506           632*   638*   653*   660*   698*   708*   769*   782*   828*   840*  1683#
COL   = 000004           241#   884    971
COLMSG  015377          1088   1390#
CONMSG  015307           866   1386#
CONTIN  012006           376    417    495    500    536    548    568    594    603    687    757    816    865#
CONT1   012016           868#   870
CRDTST  016274           591   1402#
CRLF    014240          1227   1258#
DCMSG1  014654          1006   1378#
DCMSG2  014722          1008   1035   1379#
DCMSG3  014751          1012   1045   1380#
DCMSG4  015015          1016   1052   1381#
DCMSG5  015047          1021   1382#
DCMSG6  015135          1033   1383#
DC1     012752          1022#  1028
DC2     012774          1024   1027#
DC3     013010          1026   1032#
DC4     013040          1040#  1043
DDISP = 177570           329#   332    1362
DEVCOD  012712           353   1005#
DISPLA  000502           332#  1362*  1370*
DISPRE  000174           325#  1370
DSWR  = 177570           328#   331    1361
ENDMSG  017442           364   1415#
ERR   = 104000           239#   378    379    380    389    395    419    420    421    438    447    450    460
                         461    464    467    468    477    481    484    485    509    510    515    516    527
                         532    545    557    558    579    605    608    614    615    624    629    630    635
                         637    644    645    649    651    658    659    666    667    671    672    694    695
                         701    705    706    707    718    719    723    724    733    739    740    765    766
                         779    780    781    792    793    799    800    825    826    837    838    839    850
                         851    857    858    884    888    971
ERRHLT  013306          1094#
ERROR   013126           318   1060#
ERRVEC= 000004           327#  1359   1360*  1371*
```

```
E1      013226          1074    1079#
E2      013246          1080    1085#
E3      013266          1066    1086    1090#
E4      013310          1093    1095#
FILLO   014467          1288*   1292*   1302    1339#
G     = 000002           243#
GB    = 000003           242#    378     379     380     389     395     419     420     421     447     450     460     461
                         464     467     468     481     484     485     509     510     515     516     527     532     557
                         558     579     605     608     614     615     629     630     635     637     644     645     649
                         651     658     659     666     667     671     672     694     695     701     705     706     707
                         718     719     723     724     739     740     765     766     779     780     781     792     793
                         799     800     825     826     837     838     839     850     851     857     858     884     888
                         971
GDMSG   015361          1076    1388#
GOOD    020510           378*    379*    380*    389*    395*    419*    420*    421*    443*    445     450*    460*    461*
                         464*    467*    468*    481*    484*    485*    509*    510*    515*    516*    527*    532*    557*
                         558*    579*    605*    608*    614*    615*    629*    630*    635*    637*    644*    645*    649*
                         651*    658*    659*    666*    667*    671*    672*    694*    695*    701*    705*    706*    707*
                         718*    719*    723*    724*    739*    740*    765*    766*    779*    780*    781*    792*    793*
                         799*    800*    825*    826*    837*    838*    839*    850*    851*    857*    858*    880*    882
                         888*    967*    969     1078    1684#
GTSWR   013754           349     1204#
HOLD    020512           387*    388     389     1685#
INIT    001110           357     370#
INITM1  015233           373     1385#
INIT1   001130           377#    378     379     380
INPTR   020514           759*    763*    777*    819*    823*    835*    848*    964     1686#
INTAG   000514           337#    1228
K10     014036          1215    1219#
K11     014060          1222    1224#
K14     014064          1227#
K15     014106          1194    1196    1200    1202    1229    1231#
K16     014110          1220    1232#
K17     014150          1239    1243#
K18     014162          1234    1236    1246#
K19     013774          1208#   1218
K20     014030          1217#   1247
K7      014000          1210#   1211    1245
K9      014016          1214#
LASTPC  020516           378     379     380     389     395     419     420     421     439     448     450     460     461
                         464     467     468     478     481     484     485     509     510     515     516     527     532
                         547     557     558     579     605     608     614     615     625     629     630     635     637
                         644     645     649     651     658     659     666     667     671     672     694     695     701
                         705     706     707     718     719     723     724     734     739     740     765     766     779
                         780     781     792     793     799     800     825     826     837     838     839     850     851
                         857     858     888     1064*   1687#
LEVEL   020520           428*    436*    444     1688#
LF      014241          1259#   1351
LOC     020522          1175*   1176*   1689#
LOGTST  015177           371     1384#
LP    = 012114           378#    379#    380#    389#    395#    419#    420#    421#    450#    460#    461#    464#    467#
                         468#    481#    484#    485#    509#    510#    515#    516#    527#    532#    557#    558#    579#
                         605#    608#    614#    615#    629#    630#    635#    637#    644#    645#    649#    651#    658#
                         659#    666#    667#    671#    672#    694#    695#    701#    705#    706#    707#    718#    719#
                         723#    724#    739#    740#    765#    766#    779#    780#    781#    792#    793#    799#    800#
                         825#    826#    837#    838#    839#    850#    851#    857#    858#    888#
```

```
MNEW     014226            1207    1255#
MODE0    014470            1289*   1293*   1298    1301*   1312*   1340#
MSWR     014216            1204    1253#
OCTDEC   012574             980#   1089
OD1      012610             983#    987     992
OD2      012634             985     989#
OD3      012652             988     993#
ONES     012704             981*    983*    984     989*    996     1000#
OUTPTR   020524             829*    948     949*   1690#
PASS1    004242             361     589#
PASS2    006244             674     681#
PASS3    007604             742     749#
PASS4    010706             801     808#
PCMSG    015335            1068    1387#
PCV      020560             429*    471*    618*    727*   1014     1042    1048*   1711#
PROGNM   017303             348    1413#
PS     = 177776             224#    343*    427*    433*    473*    476*    620*    623*    729*    732*
PSV      020562             430*    472*    619*    728*   1050*   1712#
PT       020526            1124*   1125    1691#
PTR      020530             697*    703*    714*    909     964*    967     973*   1692#
P1A      004634             620#    625
P1B      004710             626     628#
P1D      005046             633#    639
P1E      005544             654#    661
P1INT    004704             618     627#
P1MSG1   016327             593    1403#
P1MSG2   016412             599    1404#
P1MSG3   016464             601    1405#
P2A      006462             699#    709
P2B      007404             733#    734
P2C      007440             735     737#
P2INT    007434             727     736#
P2MSG1   016555             682    1406#
P2MSG2   016643             684     754     813    1407#
P2MSG3   016715             512     529     567     686     756     815    1408#
P2MSG4   017001             375    1409#
P2SD1    012244             912#    913     917
P2SEND   012230             702     713     737     909#
P3A      010036             771#    783
P3MSG1   017044             750    1410#
P3MSG2   017132             752    1411#
P3SD1    012322             928#    929     933
P3SD2    012350             934#    935     939
P3SEND   012304             773     787     797     925#
P4A      011144             830#    841
P4MSG1   017220             809    1412#
P4MSG2   017373             811    1414#
P4SD1    012410             946#    947     952
P4SD2    012450             953#    954     958
P4SEND   012400             832     845     855     944#
QUES     014237            1246    1257#
RDOCT    013514            1037    1047    1054    1149#
RD.1     013524            1151#   1160
RT1      014602            1360    1367#
RT2      014610            1366    1367    1369#
RT3      014624            1364    1371#
```

```
RT4     014652          1374    1376#
SB1     012214           902#    904
SCOPE   020532           345*    378     379     380     389     395     419     420     421     439     448     450     460
                         461     464     467     468     478     481     484     485     509     510     515     516     527
                         532     547     557     558     579     605     608     614     615     625     629     630     635
                         637     644     645     649     651     658     659     666     667     671     672     694     695
                         701     705     706     707     718     719     723     724     734     739     740     765     766
                         779     780     781     792     793     799     800     825     826     837     838     839     850
                         851     857     858     888    1063*   1693#
SETBFS  012210           817     901#
SR      = 177570         225#
START   001000           340     343#
ST1     001046           352     355#    366
ST2     001062           356     359#
ST3     001076           360     363#
SWR     000500           331#    351     355     359    1061    1065    1092    1166    1193    1223*   1361*   1363    1369*
                        1375*
SWRCK   014542           346    1359#
SWREG   000176           326#   1193    1205    1369
TCR     020540           378     388*    389     393*    395     419     432*    450     461     464     468     470*    481
                         483*    484     510     516     538*    558     569*    605     607*    608     617*    630     633*
                         637     641*    645     647*    651     654*    657     659     663*    667     669*    670     672
                         690*    693     695     699*    704     706     711*    717     719     721*    724     726*    738
                         740     760*    764     766     771*    778     779     785*    791     793     795*    798     800
                         820*    824     826     830*    836     837     843*    849     851     853*    856     858    1010
                        1039    1700#
TCRHI   020542           459*    466*    497*    518*    597*   1701#
TDR     020550           881     914*    930*    936*    948*    955*    968    1704#
TDRHI   020552          1705#
TEMP    020534          1127*   1128*   1129*   1130    1150*   1154*   1155*   1156*   1157*   1158*   1161    1694#
TENS    012706           982*    990*    993    1001#
TEST1   001334           382     386#
TEST2   001552           397     415#
TEST3   001764           423     427#    439     448
TEST4   002230           452     459#    460     461
TEST5   003076           487     493#
TEST6   004100           561     565#
TIR     020554           380     421     615     634     642     648     655     664     691     707     715     723     761
                         775     781     789     821     833     839     846     878     888     965    1706#
TIRHI   020556           700     712     722     772     786     796     831     844     854     912     918     928     934
                         946     953    1707#
TKB     = 177562         227#   1185    1197    1212
TKS     = 177560         226#   1182*   1183    1195    1210    1230*
TM1     013334          1109#   1115
TM2     013352          1112    1114#
TM3     013360          1110    1116#
TOG     012710           980*    986*    991*   1002#
TOG1    020536           925*    932*    944*    950*    951     956*    957    1695#
TO.1    013414          1127#   1135
TO.2    013466          1137    1139#
TO.3    013472          1140#   1143
TPB     = 177566         229#   1168*   1346*
TPS     = 177564         228#   1169    1344
TSR     020544           379     420     460     467     485     509     515     527     532     557     579     614     629
                         635     644     649     658     666     671     694     701     705     718     739     765     780
                         792     799     825     838     850     857    1702#
```

```
TSRHI   020546          507   513   525   530   543   555  1703#
TTI.1   013662         1183#  1184
TTO.1   013616         1169#  1170
TTO.2   013640         1172   1175#
TTO.3   013654         1167   1178#
TTYIN   013656          868   1022  1151  1182#
TTYN    014170         1248#
TTYOUT  013602          871   995   998   1025  1029  1091  1114  1141  1152  1166# 1174
TYPE  = 104400          240#  347   363   370   372   374   415   493   498   501   503   505   511
                        519   521   523   528   534   539   541   549   551   553   566   590   592
                        598   600   681   683   685   749   751   753   755   808   810   812   814
                        865   1005  1007  1009  1011  1013  1015  1017  1020  1032  1034  1044  1051
                       1067   1071  1075  1077  1081  1083  1087  1203  1204  1207  1216  1227  1246
                       1324
TYPEC   014472         1232   1344# 1345
TYPEX   014540         1350   1352  1355#
TYPMSG  013330         1102   1107#
TYPOC   014270         1206   1292#
TYPOCT  013366         1103   1121#
TYPON   014304         1291   1294#
TYPOS   014244         1287#
TYPOUT  013312          322   1099#
T1A     001340          387#   391
T1B     001344          388#   389
T1C     001432          393#   395
T1END   001550          390    411#
T1LST   001520          386    399#
T2MSG1  015412          416    499   1391#
T3A     002022          433#   437
T3B     002150          440    449#  450
T3INT   002072          429    442#
T4A     002362          463#   464
T4B     002436          466#   467   468
T4C     002612          473#   478
T4D     002666          479    481#
T4E     002740          483#   484   485
T4INT   002662          471    480#
T5A     003142          507#   508   509
T5B     003302          513#   514   515
T5C     003460          525#   526   527
T5D     003546          530#   531   532
T5E     003660          543#   547
T5F     003736          555#   556   557
T5MSG1  015576          502    540   1394#
T5MSG2  015665          504    522   542   552   1395#
T5MSG3  015735          506    524   554   1396#
T5MSG4  016006         1397#
T5MSG5  016046          520    550   1398#
T5MSG6  016135         1399#
T5MSG7  016175         1400#
T5M1    015504          494   1392#
T5M2    015536          535   1393#
T6MSG1  016235         1401#
XX    = 000000          238#  365   920   1000  1001  1002  1094  1679  1680  1681  1682  1683  1684
                       1685   1686  1687  1688  1689  1690  1691  1692  1693  1694  1695
.     = 020566          309#   314   317#  321#  324#  330#  339#  341#  378   379   380   389   395
```

```
                419     420     421     439     446     448     450     460     461     464     467     468     478
                481     484     485     509     510     515     516     527     532     544     547     557     558
                572     574     576     578     579     605     608     611     613     614     615     625     629
                630     634     635     637     642     644     645     648     649     651     655     657     658
                659     664     666     667     670     671     672     691     693     694     695     700     701
                704     705     706     707     712     715     717     718     719     722     723     724     734
                738     739     740     761     764     765     766     772     775     778     779     780     781
                786     789     791     792     793     796     798     799     800     821     824     825     826
                831     833     836     837     838     839     844     846     849     850     851     854     856
                857     858     883     888     919     970    1062    1177   1248#   1286#   1416#
```

CZCTAA.SRC 19-MAY-80 14:57 CROSS REFERENCE TABLE -- MACRO NAMES SEQ 0056

```
CKTCR    270#  378  389  395  419  450  461  464  468  481  484  510  516  558  605
         608   630  637  645  651  659  667  672  695  706  719  724  740  766  779
         793   800  826  837  851  858
CKTDR    290#
CKTIR    300#  380  421  615  707  723  781  839  888
CKTSR    280#  379  420  460  467  485  509  515  527  532  557  579  614  629  635
         644   649  658  666  671  694  701  705  718  739  765  780  792  799  825
         838   850  857
LOOP     260#  378  379  380  389  395  419  420  421  439  448  450  460  461  464
         467   468  478  481  484  485  509  510  515  516  527  532  547  557  558
         579   605  608  614  615  625  629  630  635  637  644  645  649  651  658
         659   666  667  671  672  694  695  701  705  706  707  718  719  723  724
         734   739  740  765  766  779  780  781  792  793  799  800  825  826  837
         838   839  850  851  857  858  888
WTDONE   246#  657  670  693  704  717  738  764  778  791  798  824  836  849  856
WTIDR    256#  634  642  648  655  664  691  715  761  775  789  821  833  846
WTODR    251#  700  712  722  772  786  796  831  844  854
```

. ABS. 020566 000

ERRORS DETECTED: 0

CZCTAA,CZCTAA/CRF=CZCTAA.SRC
RUN-TIME: 9 18 2 SECONDS
RUN-TIME RATIO: 157/30=5.2
CORE USED: 8K (15 PAGES)