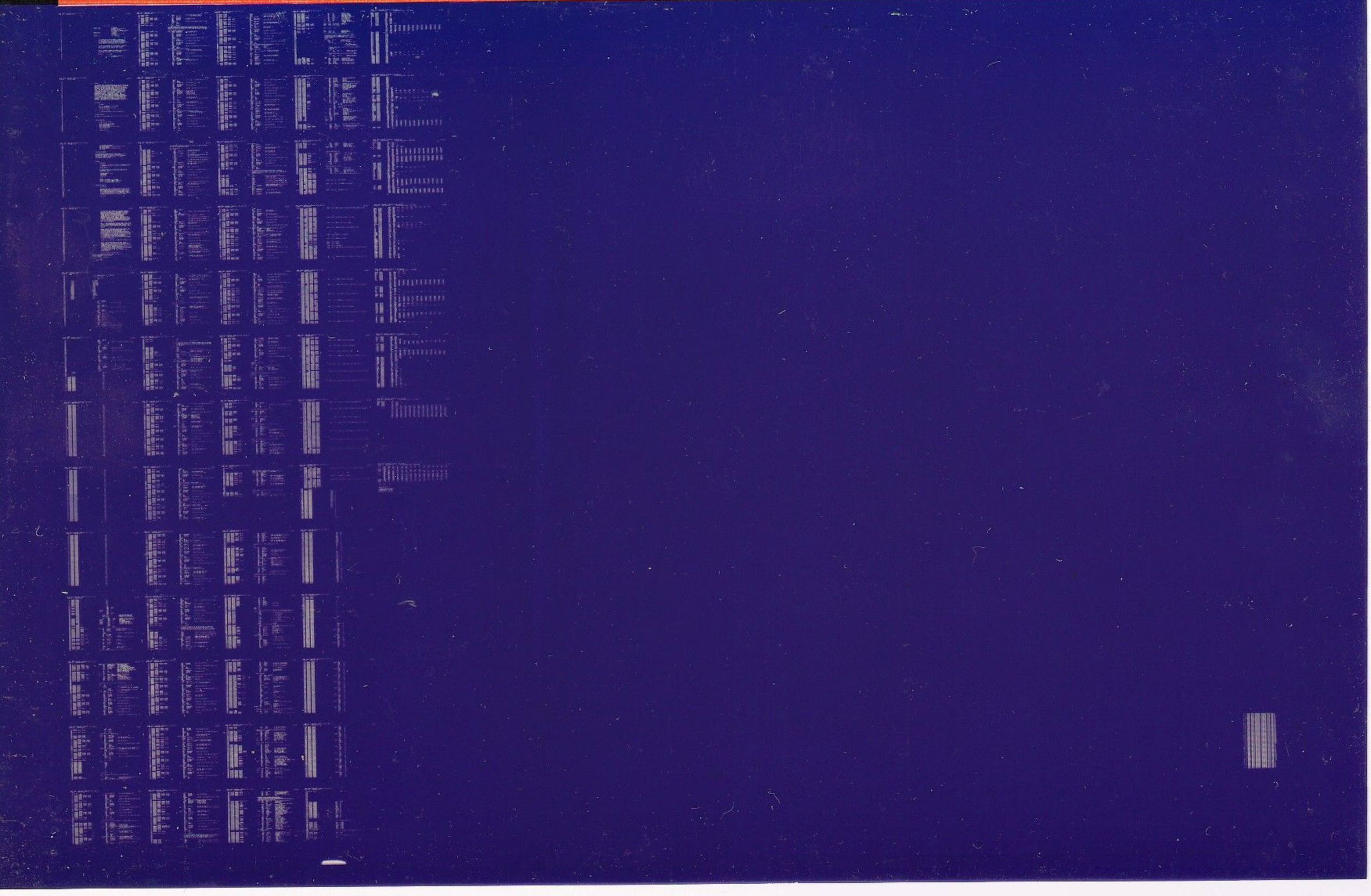
CTS11-JC CTS11-JC HOLLORITH AH-S076A-MC
CZCTBAO FICHE 1 OF 1



```
                        .REM _
```

IDENTIFICATION

PRODUCT CODE:	AC-S068A-MC
PRODUCT NAME:	CZCTBA0 CTS11-JC HOLLORITH 8035-8045 (11/70) 80 COLUMN CARD TERMINAL CONTROL DATA FORMAT: 8 BIT HOLLERITH
UPDATED DATE	20 MARCH 80
MAINTAINER:	J. 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 NORMAI  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 READIN
                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 PER
                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 INT
                        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:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXY

.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 THE LETTER " A "  PUNCHED
                        CARD #2 WILL HAVE 2 CHARACTERS (LETTER A) UP TO
                        CARD #20 WHICH WILL HAVE 20 CHARACTERS (LETTER A) PUNCHE
.                       ALL THE CHARACTERS PUNCHED IN THIS TEST WILL BE CAPITAL
.                         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 HOLERITH. (8 BIT) FORMAT
;DECSPEC-CZCTBA
;       ***** MODEL 8035-8045 *****
;AUTHORS:
;       P. W. DUKE
;       A. L. UNSER
;20-DEC-73
```

```
                                        ;UPDATE AUTHOR: VIJAY ANANDWALA
                                ;[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 EXPEC
                                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 EX
                                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 EXPE
                                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
                                LOOP    B
                                .ENDM
                 000000                 .=0
                 000140                 .REPT    140
                                        .+2
                                        HALT
                                        .ENDR
(1)   000000     000002                 .+2
(1)   000002     000000                 HALT
(1)   000004     000006                 .+2
(1)   000006     000000                 HALT
(1)   000010     000012                 .+2
(1)   000012     000000                 HALT
(1)   000014     000016                 .+2
(1)   000016     000000                 HALT
(1)   000020     000022                 .+2
(1)   000022     000000                 HALT
(1)   000024     000026                 .+2
(1)   000026     000000                 HALT
(1)   000030     000032                 .+2
(1)   000032     000000                 HALT
(1)   000034     000036                 .+2
(1)   000036     000000                 HALT
```

```
    (1)  000040  000042                .+2
    (1)  000042  000000                HALT
    (1)  000044  000046                .+2
    (1)  000046  000000                HALT
    (1)  000050  000052                .+2
    (1)  000052  000000                HALT
    (1)  000054  000056                .+2
    (1)  000056  000000                HALT
    (1)  000060  000062                .+2
    (1)  000062  000000                HALT
    (1)  000064  000066                .+2
    (1)  000066  000000                HALT
    (1)  000070  000072                .+2
    (1)  000072  000000                HALT
    (1)  000074  000076                .+2
    (1)  000076  000000                HALT
    (1)  000100  000102                .+2
    (1)  000102  000000                HALT
    (1)  000104  000106                .+2
    (1)  000106  000000                HALT
    (1)  000110  000112                .+2
    (1)  000112  000000                HALT
    (1)  000114  000116                .+2
    (1)  000116  000000                HALT
    (1)  000120  000122                .+2
    (1)  000122  000000                HALT
    (1)  000124  000126                .+2
    (1)  000126  000000                HALT
    (1)  000130  000132                .+2
    (1)  000132  000000                HALT
    (1)  000134  000136                .+2
    (1)  000136  000000                HALT
    (1)  000140  000142                .+2
    (1)  000142  000000                HALT
    (1)  000144  000146                .+2
    (1)  000146  000000                HALT
    (1)  000150  000152                .+2
    (1)  000152  000000                HALT
    (1)  000154  000156                .+2
    (1)  000156  000000                HALT
    (1)  000160  000162                .+2
    (1)  000162  000000                HALT
    (1)  000164  000166                .+2
    (1)  000166  000000                HALT
    (1)  000170  000172                .+2
    (1)  000172  000000                HALT
    (1)  000174  000176                .+2
    (1)  000176  000000                HALT
    (1)  000200  000202                .+2
    (1)  000202  000000                HALT
    (1)  000204  000206                .+2
    (1)  000206  000000                HALT
    (1)  000210  000212                .+2
    (1)  000212  000000                HALT
    (1)  000214  000216                .+2
    (1)  000216  000000                HALT
```

I 1

SEQ 0008

```
(1)  000220  000222                .+2
(1)  000222  000000                HALT
(1)  000224  000226                .+2
(1)  000226  000000                HALT
(1)  000230  000232                .+2
(1)  000232  000000                HALT
(1)  000234  000236                .+2
(1)  000236  000000                HALT
(1)  000240  000242                .+2
(1)  000242  000000                HALT
(1)  000244  000246                .+2
(1)  000246  000000                HALT
(1)  000250  000252                .+2
(1)  000252  000000                HALT
(1)  000254  000256                .+2
(1)  000256  000000                HALT
(1)  000260  000262                .+2
(1)  000262  000000                HALT
(1)  000264  000266                .+2
(1)  000266  000000                HALT
(1)  000270  000272                .+2
(1)  000272  000000                HALT
(1)  000274  000276                .+2
(1)  000276  000000                HALT
(1)  000300  000302                .+2
(1)  000302  000000                HALT
(1)  000304  000306                .+2
(1)  000306  000000                HALT
(1)  000310  000312                .+2
(1)  000312  000000                HALT
(1)  000314  000316                .+2
(1)  000316  000000                HALT
(1)  000320  000322                .+2
(1)  000322  000000                HALT
(1)  000324  000326                .+2
(1)  000326  000000                HALT
(1)  000330  000332                .+2
(1)  000332  000000                HALT
(1)  000334  000336                .+2
(1)  000336  000000                HALT
(1)  000340  000342                .+2
(1)  000342  000000                HALT
(1)  000344  000346                .+2
(1)  000346  000000                HALT
(1)  000350  000352                .+2
(1)  000352  000000                HALT
(1)  000354  000356                .+2
(1)  000356  000000                HALT
(1)  000360  000362                .+2
(1)  000362  000000                HALT
(1)  000364  000366                .+2
(1)  000366  000000                HALT
(1)  000370  000372                .+2
(1)  000372  000000                HALT
(1)  000374  000376                .+2
(1)  000376  000000                HALT
```

J 1

SEQ 0009

```
(1)  000400  000402                .+2
(1)  000402  000000                HALT
(1)  000404  000406                .+2
(1)  000406  000000                HALT
(1)  000410  000412                .+2
(1)  000412  000000                HALT
(1)  000414  000416                .+2
(1)  000416  000000                HALT
(1)  000420  000422                .+2
(1)  000422  000000                HALT
(1)  000424  000426                .+2
(1)  000426  000000                HALT
(1)  000430  000432                .+2
(1)  000432  000000                HALT
(1)  000434  000436                .+2
(1)  000436  000000                HALT
(1)  000440  000442                .+2
(1)  000442  000000                HALT
(1)  000444  000446                .+2
(1)  000446  000000                HALT
(1)  000450  000452                .+2
(1)  000452  000000                HALT
(1)  000454  000456                .+2
(1)  000456  000000                HALT
(1)  000460  000462                .+2
(1)  000462  000000                HALT
(1)  000464  000466                .+2
(1)  000466  000000                HALT
(1)  000470  000472                .+2
(1)  000472  000000                HALT
(1)  000474  000476                .+2
(1)  000476  000000                HALT
(1)  000500  000502                .+2
(1)  000502  000000                HALT
(1)  000504  000506                .+2
(1)  000506  000000                HALT
(1)  000510  000512                .+2
(1)  000512  000000                HALT
(1)  000514  000516                .+2
(1)  000516  000000                HALT
(1)  000520  000522                .+2
(1)  000522  000000                HALT
(1)  000524  000526                .+2
(1)  000526  000000                HALT
(1)  000530  000532                .+2
(1)  000532  000000                HALT
(1)  000534  000536                .+2
(1)  000536  000000                HALT
(1)  000540  000542                .+2
(1)  000542  000000                HALT
(1)  000544  000546                .+2
(1)  000546  000000                HALT
(1)  000550  000552                .+2
(1)  000552  000000                HALT
(1)  000554  000556                .+2
(1)  000556  000000                HALT
```

```
(1)  000560  000562                                    .+2
(1)  000562  000000                                    HALT
(1)  000564  000566                                    .+2
(1)  000566  000000                                    HALT
(1)  000570  000572                                    .+2
(1)  000572  000000                                    HALT
(1)  000574  000576                                    .+2
(1)  000576  000000                                    HALT
315          000030                                    .=30
316  000030  013476                                    .WORD   ERROR
317  000032  000340                                    .WORD   340
318          000174                                    .=174
319  000174  000000                            DISPREG:.WORD   0
320  000176  000000                            SWREG:  .WORD   0
321          000034                                    .=34
322  000034  013662                                    .WORD   TYPOUT
323  000036  000340                                    .WORD   340
324          000004                        ERRVEC =4                ;TIME OUT AND OTHER ERROR
325          177570                        DSWR  =177570            ;HARDWARE SWITCH REGISTER
326          177570                        DDISP=177570             ;HARDWARE DISPLAY REGISTER
327          000700                                    .=700
328  000700  177570                            SWR:    .WORD   DSWR     ;ADDRESS OF SWITCH REGISTER
329  000702  177570                        DISPLAY:    .WORD   DDISP    ;ADDRESS OF DISPLAY REGISTER
330  000704  000000                        APASS:      .WORD   0        ;PASS COUNT
331  000706  000000                        AENVM:      .WORD   0        ;ENVIRONMENT REGISTER
332  000710  000000                        ASWREG:     .WORD   0        ;APT SWITCH REGISTER
333  000712     000                        AUTOB:      .BYTE   0        ;AUTOMATIC MODE INDICATOR
334  000713     000                        INTAG:      .BYTE   0        ;INTERRUPT MODE INDICATOR
335  000714  000000                                    .WORD
336          000200                                    .=200
337  000200  000167  000574                            JMP     START
338          001000                                    .=1000
339  001000  012767  000340  176770    START:  MOV     #340,PS          ;SET BPU TO LEVEL #7
340  001006  012706  001000                    MOV     #1000,SP         ;INITIALIZE THE STACK POINTER
341  001012  005067  020052                    CLR     SCOPE            ;INITIALIZE SCOPE
342  001016  004767  000066                    JSR     PC,SWRCK
343  001022  104400                            TYPE
344  001024  017745                            PROGNM                   ;'' DECSPEC-11-CZCTBA0-CARD READER - TEST
345  001026  004767  013174                    JSR     PC,GTSWR
346  001032  032777  000004  177640            BIT     #4,@SWR
347  001040  001002                            BNE     ST1              ;IF SR2=0
348  001042  004767  012214                    JSR     PC,DEVCOD        ;GO TO DEVICE CODE ENTRY ROUTINE
349
350  001046  032777  000002  177624    ST1:    BIT     #2,@SWR
351  001054  001402                            BEQ     ST2              ;IF SR1=1
352  001056  004767  000140                    JSR     PC,INIT          ;GO DO LOGIC TESTS
353
354  001062  032777  000001  177610    ST2:    BIT     #1,@SWR
355  001070  001402                            BEQ     ST3              ;IF SR0=1
356  001072  004767  003266                    JSR     PC,PASS1         ;GO DO CARD TESTS
357
358  001076  104400                    ST3:    TYPE
359  001100  017722                            ENDMSG                   ;''END OF TESTING''
360  001102  000000                            XX                       ;HALT AT END
361  001104  000137  001046                    JMP     @#ST1
362                                    ;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

;
;3 REGISTERS SHOULD EQUAL ZERO AT INITIALIZATION
;
INIT:   TYPE
        LOGTST                  ;"***** LOGIC TESTS *****"
        TYPE
        INITM1                  ;"PUT THE 8035-8045 OFF-LINE, REMOVE ALL
                TYPE
                P2MSG4                  ;PRESS STOP RESET ON 8035-8045
        JSR     PC,CONTIN       ;"PRESS CR TO CONTINUE"
INIT1:  RESET
        CKTCR   #0,INIT1        ;TEST CMD REG FOR 0
        LP=.
        MOV     #0,GOOD
        MOV     @TCR,BAD        ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    INIT1
        JMP     @#INIT1
        CKTSR   #0,INIT1        ;TEST STATUS REG FOR 0
        LP=.
        MOV     #0,GOOD
        MOV     @TSR,BAD        ;READ STATUS REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
        LOOP    INIT1
        JMP     @#INIT1
        CKTIR   #0,INIT1        ;TEST INTERRUPT REG FOR 0
        LP=.
        MOV     #0,GOOD
        MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
```

```
363
364  001110  013746  000004
365  001114  012737  001150  000004
366  001122  012767  177570  177550
367  001130  012767  177570  177544
368  001136  022777  177777  177534
369  001144  001012
370
371  001146  000403
372  001150  012716  001156
373  001154  000002
374  001156  012767  000176  177514
375  001164  012767  000174  177510
376  001172  012637  000004
377  001176  005067  177502
378  001202  132767  000200  177476
379  001210  001403
380  001212  012767  000710  177460
381  001220  000207
382
383
384
385
386  001222  104400
387  001224  015441
388  001226  104400
389  001230  015475
390  001232  104400
391  001234  017274
392  001236  004767  010750
393  001242  000005
394  001244
(1)          001244
(1)  001244  012767  000000  017574
(1)  001252  017767  017622  017556
(1)  001260  026767  017562  017550
(1)  001266  001401
(1)  001270  104003
(1)  001272
(2)  001312  000137  001242
395  001316
(1)          001316
(1)  001316  012767  000000  017522
(1)  001324  017767  017554  017504
(1)  001332  026767  017510  017476
(1)  001340  001401
(1)  001342  104003
(1)  001344
(2)  001364  000137  001242
396  001370
(1)          001370
(1)  001370  012767  000000  017450
(1)  001376  017767  017512  017432
(1)  001404  026767  017436  017424
(1)  001412  001401
(1)  001414  104003
```

```
(1)  001416
(2)  001436  000137  001242
397
398  001442  000137  001446
399
400
401
402  001446  012702  001632
403  001452  012267  017372
404  001456  016777  017366  017414
405  001464
(1)          001464
(1)  001464  016767  017360  017354
(1)  001472  017767  017402  017336
(1)  001500  026767  017342  017330
(1)  001506  001401
(1)  001510  104003
(1)  001512
(2)  001532  000137  001456
406  001536  020227  001664
407  001542  001343
408
409  001544  012777  074136  017326
410  001552  000005
411  001554
(1)          001554
(1)  001554  012767  000000  017264
(1)  001562  017767  017312  017246
(1)  001570  026767  017252  017240
(1)  001576  001401
(1)  001600  104003
(1)  001602
(2)  001622  000137  001544
412
413  001626  000137  001664
414
415  001632  000000
416  001634  040000
417  001636  020000
418  001640  010000
419  001642  004000
420  001644  000100
421  001646  000020
422  001650  000010
423  001652  000004
424  001654  000002
425  001656  000000
426  001660  074136
427  001662  000000
428
429
430
431  001664  104400
432  001666  015653
433  001670  004767  010316
434
```

```
        LOOP    INIT1
        JMP     @#INIT1

        JMP     @#TEST1
        .
;LOAD-READ TEST OF COMMAND REGISTER
;
TEST1:  MOV     #T1LST,R2
T1A:    MOV     (R2)+,HOLD
T1B:    MOV     HOLD,@TCR       ;LOAD CMD REGISTER
        CKTCR   HOLD,T1B        ;TEST CMD REG FOR CORRECT VALUE
        LP=.
        MOV     HOLD,GOOD
        MOV     @TCR,BAD        ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    T1B
        JMP     @#T1B
        CMP     R2,#T1END+2
        BNE     T1A
;NOW SEE IF RESET CLEARS ALL BITS THAT CAN BE SET IN THE COMMAND REGISTE
T1C:    MOV     #74136,@TCR     ;LOAD COMMAND REG WITH ALL BITS
        RESET
        CKTCR   #0,T1C          ;TEST CMD REG FOR 0
        LP=.
        MOV     #0,GOOD
        MOV     @TCR,BAD        ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    T1C
        JMP     @#T1C

        JMP     @#TEST2

T1LST:  0
        40000
        20000
        10000
        4000
        100
        20
        10
        4
        2
        0
        74136
T1END:  0
        .
;TEST THAT DONE BIT IS ONLY BIT SET IN 3 REGISTERS
;
TEST2:  TYPE
        T2MSG1                  ;"PUT 8035-8045 ON-LINE AND PRESS START
        JSR     PC,CONTIN       ;"PRESS CR TO CONTINUE"
;DONE SHOULD BE THE ONLY BIT SET IN 3 REGISTERS
```

```
   435  001674
   (1)            001674
   (1)  001674  012767  000200  017144
   (1)  001702  017767  017172  017126
   (1)  001710  026767  017132  017120
   (1)  001716  001401
   (1)  001720  104003
   (1)  001722
   (2)  001742  000137  001674
   436  001746
   (1)            001746
   (1)  001746  012767  000000  017072
   (1)  001754  017767  017124  017054
   (1)  001762  026767  017060  017046
   (1)  001770  001401
   (1)  001772  104003
   (1)  001774
   (2)  002014  000137  001746
   437  002020
   (1)            002020
   (1)  002020  012767  000000  017020
   (1)  002026  017767  017062  017002
   (1)  002034  026767  017006  016774
   (1)  002042  001401
   (1)  002044  104003
   (1)  002046
   (2)  002066  000137  002020
   438
   439  002072  000137  002076
   440
   441
   442
   443  002076  012767  000340  175672
   444  002104  012767  000007  016744
   445  002112  012777  002204  017000
   446  002120  012777  000340  016774
   447
   448  002126  012777  000100  016744
   449  002134  162767  000040  175634
   450  002142  000240
   451  002144  000240
   452  002146  005367  016704
   453  002152  100370
   454  002154  104000
   455  002156
   (1)  002176  000137  002076
   456  002202  000427
   457
   458  002204  062706  000004
   459  002210  016767  016710  016630
   460  002216  016767  016634  016612
   461  002224  026767  016616  016604
   462  002232  001401
   463  002234  104003
   464  002236
   (1)  002256  000137  002076
```

```
CKTCR   #200,LP         ;TEST CMD REG FOR DONE FLAG
LP=.
MOV     #200,GOOD
MOV     @TCR,BAD        ;READ CMD REGISTER
CMP     GOOD,BAD
BEQ     .+4
ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPcC
LOOP    LP
JMP     @#LP
CKTSR   #0,LP           ;TEST STATUS REG FOR 0
LP=.
MOV     #0,GOOD
MOV     @TSR,BAD        ;READ STATUS REGISTER
CMP     GOOD,BAD
BEQ     .+4
ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
LOOP    LP
JMP     @#LP
CKTIR   #0,LP           ;TEST INTERRUPT REG FOR 0
LP=.
MOV     #0,GOOD
MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
CMP     GOOD,BAD
BEQ     .+4
ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
LOOP    LP
JMP     @#LP

        JMP     @#TEST3
        :
;TEST THAT DONE BIT CAUSES INTERRUPT TO CORRECT ADDRESS AND BR LEVEL
;
TEST3:  MOV     #340,PS
        MOV     #7,LEVEL
        MOV     #T3INT,@PCV     ;SETUP PC VECTOR
        MOV     #340,@PSV       ;SETUP PS VECTOR
;SHOULD GET AN INTERRUPT WHEN BPU IS SET TO ONE LEVEL BELOW BR LEVEL FOR
        MOV     #100,@TCR       ;ENABLE INTERRUPT IN CMD REG
T3A:    SUB     #40,PS          ;DROP BPU LEVEL BY 1
        NOP
        NOP
        DEC     LEVEL           ;IF LEVEL IS 0 OR HIGHER
        BPL     T3A             ;GO WAIT FOR INT
        ERR                     ;GOT NO INTERRUPT WITH BPU AT LEVEL 0
        LOOP    TEST3
        JMP     @#TEST3
        BR      T3B
;GOT INTERRUPT, NOW TEST TO SEE THAT IT OCCURRED AT CORRECT LEVEL
T3INT:  ADD     #4,SP           ;HOUSE KEEPING OF STACK
        MOV     BRLV,GOOD       ;GOOD=BR LEVEL OF DEVICE
        MOV     LEVEL,BAD       ;BAD=LEVEL AT WHICH INTERRUPT OCCURRED
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;INTERRUPT OCCURRED AT WRONG LEVEL
        LOOP    TEST3
        JMP     @#TEST3
```

```
 465  002262  000005
 466  002264
(1)             002264
(1)  002264  012767  000200  016554
(1)  002272  017767  016602  016536
(1)  002300  026767  016542  016530
(1)  002306  001401
(1)  002310  104003
(1)  002312
(2)  002332  000137  002262
 467
 468  002336  000137  002342
 469
 470
 471
 472
 473
 474
 475  002342  112777  000040  016532
 476  002350
(1)             002350
(1)  002350  012767  000220  016470
(1)  002356  017767  016522  016452
(1)  002364  026767  016456  016444
(1)  002372  001401
(1)  002374  104003
(1)  002376
(2)  002416  000137  002342
 477  002422
(1)             002422
(1)  002422  012767  120200  016416
(1)  002430  017767  016444  016400
(1)  002436  026767  016404  016372
(1)  002444  001401
(1)  002446  104003
(1)  002450
(2)  002470  000137  002342
 478
 479  002474  000005
 480  002476
(1)             002476
(1)  002476  012767  000200  016342
(1)  002504  017767  016370  016324
(1)  002512  026767  016330  016316
(1)  002520  001401
(1)  002522  104003
(1)  002524
(2)  002544  000137  002474
 481
 482  002550  112777  000100  016324
 483  002556
(1)             002556
(1)  002556  012767  000220  016262
(1)  002564  017767  016314  016244
(1)  002572  026767  016250  016236
(1)  002600  001401
```

```
T3B:    RESET                           ;CLEAR OUT CMD REG WITH RESET
        CKTCR   #200,T3B                        ;TEST CMD REG FOR DONE
        LP=.
        MOV     #200,GOOD
        MOV     @TCR,BAD                ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                          ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    T3B
        JMP     @#T3B

        JMP     @#TEST4
        .
;TEST THAT HOPPER EMPTY CAN BE DETECTED WHEN HOPPERS ARE SELECTED, THAT
;HOPPER EMPTY BEING SET CAUSES INPUT ERROR TO SET AND THIS IN TURN CAUSE
;ERROR BIT TO SET IN THE CMD REG. TEST THAT ERROR BIT CAUSES AN INTERRUP
;
;TEST FOR PRIMARY HOPPER EMPTY
TEST4:  MOVB    #40,@TCRHI              ;SELECT PRIMARY HOPPER
        CKTSR   #220,TEST4              ;TEST FOR IE + HE
        LP=.
        MOV     #220,GOOD
        MOV     @TSR,BAD                ;READ STATUS REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                          ;STATUS REG - VALUE READ NOT EQUAL TO EX
        LOOP    TEST4
        JMP     @#TEST4
        CKTCR   #120200,TEST4           ;TEST FOR ERR + HSO + DONE
        LP=.
        MOV     #120200,GOOD
        MOV     @TCR,BAD                ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                          ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    TEST4
        JMP     @#TEST4
;TEST THAT RESET WILL CLEAR OUT CMD REG
T4A:    RESET                           ;CLEAR OUT CMD REG WITH RESET
        CKTCR   #200,T4A                        ;TEST CMD REG FOR DONE
        LP=.
        MOV     #200,GOOD
        MOV     @TCR,BAD                ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                          ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    T4A
        JMP     @#T4A
        ;TEST FOR SECONDARY HOPPER EMPTY
T4B:    MOVB    #100,@TCRHI             ;SELECT SECONDARY HOPPER
        CKTSR   #220,T4B                ;TEST FOR IE + HE
        LP=.
        MOV     #220,GOOD
        MOV     @TSR,BAD                ;READ STATUS REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
```

```
    (1)  002602  104003
    (1)  002604
    (2)  002624  000137  002550
    484  002630
    (1)          002630
    (1)  002630  012767  140200  016210
    (1)  002636  017767  016236  016172
    (1)  002644  026767  016176  016164
    (1)  002652  001401
    (1)  002654  104003
    (1)  002656
    (2)  002676  000137  002550
    485
    486  002702  112777  000100  016170
    487  002710  012777  002774  016202
    488  002716  012777  000340  016176
    489  002724  012767  000040  175044
    490  002732  000240
    491  002734  000240
    492  002736  012767  000340  175032
    493  002744  104000
    494  002746
    (1)  002766  000137  002724
    495  002772  000402
    496  002774  062706  000004
    497  003000
    (1)          003000
    (1)  003000  012767  140300  016040
    (1)  003006  017767  016066  016022
    (1)  003014  026767  016026  016014
    (1)  003022  001401
    (1)  003024  104003
    (1)  003026
    (2)  003046  000137  003000
    498
    499  003052  012777  000400  016020
    500  003060
    (1)          003060
    (1)  003060  012767  000200  015760
    (1)  003066  017767  016006  015742
    (1)  003074  026767  015746  015734
    (1)  003102  001401
    (1)  003104  104003
    (1)  003106
    (2)  003126  000137  003052
    501  003132
    (1)          003132
    (1)  003132  012767  000000  015706
    (1)  003140  017767  015740  015670
    (1)  003146  026767  015674  015662
    (1)  003154  001401
    (1)  003156  104003
    (1)  003160
    (2)  003200  000137  003052
    502
    503  003204  000137  003210
```

```
                    ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                    LOOP    T4B
                    JMP     @#T4B
                    CKTCR   #140200,T4B     ;TEST FOR ERR + HS1 + DONE
                    LP=.
                    MOV     #140200,GOOD
                    MOV     @TCR,BAD        ;READ CMD REGISTER
                    CMP     GOOD,BAD
                    BEQ     .+4
                    ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                    LOOP    T4B
                    JMP     @#T4B
                    ;TEST THAT ERROR BIT IN CMD REG CAUSES AN INT
                    MOVB    #100,@TCR       ;ENABLE INT
                    MOV     #T4INT,@PCV     ;SETUP PC VECTOR
                    MOV     #340,@PSV       ;SETUP PS VECTOR
T4C:                MOV     #40,PS          ;SET BPU TO LEVEL #1
                    NOP
                    NOP
                    MOV     #340,PS         ;SET BPU TO LEVEL #7
                    ERR                     ;WITH ERROR BIT SET IN CMD REG INT DIDN'
                    LOOP    T4C
                    JMP     @#T4C
                    BR      T4D
T4INT:              ADD     #4,SP           ;HOUSE KEEPING ON STACK
T4D:                CKTCR   #140300,T4D     ;TEST FOR ERR + HS1 + IE + DONE
                    LP=.
                    MOV     #140300,GOOD
                    MOV     @TCR,BAD        ;READ CMD REGISTER
                    CMP     GOOD,BAD
                    BEQ     .+4
                    ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                    LOOP    T4D
                    JMP     @#T4D
                ;TEST THAT WRITING A 1 TO ERROR BIT CLEAR WILL CLEAR IT
T4E:                MOV     #400,@TCR       ;CLEAR ERROR BIT BY WRITING A 1 TO IT
                    CKTCR   #200,T4E                ;TEST CMD REG FOR DONE
                    LP=.
                    MOV     #200,GOOD
                    MOV     @TCR,BAD        ;READ CMD REGISTER
                    CMP     GOOD,BAD
                    BEQ     .+4
                    ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                    LOOP    T4E
                    JMP     @#T4E
                    CKTSR   #0,T4E          ;TEST STATUS REG FOR 0
                    LP=.
                    MOV     #0,GOOD
                    MOV     @TSR,BAD        ;READ STATUS REGISTER
                    CMP     GOOD,BAD
                    BEQ     .+4
                    ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                    LOOP    T4E
                    JMP     @#T4E

                    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
                                                        JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
                                                        TYPE
                                                        T5MSG1                  ;''PULL CARD PLATE OF STACKER #1 BACK TO
                                                        TYPE
                                                        T5MSG2                  ;WILL SIMULATE STACKER FULL AND HOLD IT
                                                        TYPE
                                                        T5MSG3                  ;''PROGRAM WILL WAIT FOR STACKER FULL FLA
                                                T5A:    BITB    #20,@TSRHI
                                                        BEQ     T5A             ;WAIT FOR STACKER FULL FLAG
                                                        CKTSR   #110000,T5A     ;TEST FOR OE + SF
                                                        LP=.
                                                        MOV     #110000,GOOD
                                                        MOV     @TSR,BAD        ;READ STATUS REGISTER
                                                        CMP     GOOD,BAD
                                                        BEQ     .+4
                                                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                                                        LOOP    T5A
                                                        JMP     @#T5A
                                                        CKTCR   #104000,LP      ;TEST FOR ERR + SSO + DONE
                                                        LP=.
                                                        MOV     #104000,GOOD
                                                        MOV     @TCR,BAD        ;READ CMD REGISTER
                                                        CMP     GOOD,BAD
                                                        BEQ     .+4
                                                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                                                        LOOP    LP
                                                        JMP     @#LP
                                                        TYPE
                                                        P2MSG3                  ;'' PRESS STOP-RESET AND PRESS START ON T
                                                T5B:    BITB    #20,@TSRHI
                                                        BNE     T5B             ;WAIT FOR STACKER FULL TO GO AWAY
                                                        CKTSR   #0,T5B          ;TEST STATUS REG FOR 0
                                                        LP=.
                                                        MOV     #0,GOOD
                                                        MOV     @TSR,BAD        ;READ STATUS REGISTER
                                                        CMP     GOOD,BAD
                                                        BEQ     .+4
                                                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                                                        LOOP    T5B
                                                        JMP     @#T5B
                                                        CKTCR   #104000,LP      ;TEST FOR ERR + SSO
                                                        LP=.
                                                        MOV     #104000,GOOD
                                                        MOV     @TCR,BAD        ;READ CMD REGISTER
```

```
  504
  505
  506
  507
  508
  509  003210  104400
  510  003212  015744
  511  003214  004767  006772
  512  003220  000005
  513  003222  112777  000010  015652
  514  003230  104400
  515  003232  015653
  516  003234  004767  006752
  517  003240  104400
  518  003242  016075
  519  003244  104400
  520  003246  016164
  521  003250  104400
  522  003252  016234
  523  003254  132777  000020  015624
  524  003262  001774
  525  003264
  (1)          003264
  (1)  003264  012767  110000  015554
  (1)  003272  017767  015606  015536
  (1)  003300  026767  015542  015530
  (1)  003306  001401
  (1)  003310  104003
  (1)  003312
  (2)  003332  000137  003254
  526  003336
  (1)          003336
  (1)  003336  012767  104000  015502
  (1)  003344  017767  015530  015464
  (1)  003352  026767  015470  015456
  (1)  003360  001401
  (1)  003362  104003
  (1)  003364
  (2)  003404  000137  003336
  527  003410  104400
  528  003412  017210
  529  003414  132777  000020  015464
  530  003422  001374
  531  003424
  (1)          003424
  (1)  003424  012767  000000  015414
  (1)  003432  017767  015446  015376
  (1)  003440  026767  015402  015370
  (1)  003446  001401
  (1)  003450  104003
  (1)  003452
  (2)  003472  000137  003414
  532  003476
  (1)          003476
  (1)  003476  012767  104000  015342
  (1)  003504  017767  015370  015324
```

```
(1)  003512  026767  015330  015316                    CMP     GOOD,BAD
(1)  003520  001401                                    BEQ     .+4
(1)  003522  104003                                    ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
(1)  003524                                            LOOP    LP
(2)  003544  000137  003476                            JMP     @#LP
533                                                    ;TEST FOR STACKER #2 FULL
534  003550  112777  000031  015324                    MOVB    #31,@TCRHI      ;SELECT STACKER #2 + CLEAR ERR
535  003556  104400                                    TYPE
536  003560  016344                                    T5MSG5                  ;''PULL CARD PLATE OF STACKER #2 BACK TO
537  003562  104400                                    TYPE
538  003564  016164                                    T5MSG2                  ;WILL SIMULATE STACKER FULL AND HOLD IT
539  003566  104400                                    TYPE
540  003570  016234                                    T5MSG3                  ;''PROGRAM WILL WAIT FOR STACKER FULL FLA
541  003572  132777  000020  015306          T5C:    BITB    #20,@TSRHI
542  003600  001774                                    BEQ     T5C             ;WAIT FOR STACKER FULL FLAG
543  003602                                            CKTSR   #110000,T5C     ;TEST FOR OE + SF
(1)            003602                                  LP=.
(1)  003602  012767  110000  015236                    MOV     #110000,GOOD
(1)  003610  017767  015270  015220                    MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  003616  026767  015224  015212                    CMP     GOOD,BAD
(1)  003624  001401                                    BEQ     .+4
(1)  003626  104003                                    ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
(1)  003630                                            LOOP    T5C
(2)  003650  000137  003572                            JMP     @#T5C
544  003654  104400                                    TYPE
545  003656  017210                                    P2MSG3          ;''PRESS STOP-RESET AND PRESS START ON THE 8035-8
546  003660  132777  000020  015220          T5D:    BITB    #20,@TSRHI
547  003666  001374                                    BNE     T5D             ;WAIT FOR STACKER FULL TO GO AWAY
543  003670                                            CKTSR   #0,T5D          ;TEST STATUS REG FOR 0
(1)            003670                                  LP=.
(1)  003670  012767  000000  015150                    MOV     #0,GOOD
(1)  003676  017767  015202  015132                    MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  003704  026767  015136  015124                    CMP     GOOD,BAD
(1)  003712  001401                                    BEQ     .+4
(1)  003714  104003                                    ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
(1)  003716                                            LOOP    T5D
(2)  003736  000137  003660                            JMP     @#T5D
549                                                    ;TEST FOR STACKER FULL IN STACKER OVERFLOW MODE
550  003742  104400                                    TYPE
551  003744  016036                                    T5M2                    ;''PRESS START ON THE 8035-8045''
552  003746  004767  006240                            JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE
553  003752  000005                                    RESET                   ;CLEAR OUT CMD REG
554  003754  012777  000001  015116                    MOV     #1,@TCR         ;SET STACKER OVERFLOW MODE + GO
555  003762  104400                                    TYPE
556  003764  016075                                    T5MSG1                  ;''PULL CARD PLATE OF STACKER #1 BACK TO
557  003766  104400                                    TYPE
558  003770  016164                                    T5MSG2                  ;WILL SIMULATE STACKER FULL AND RELEASE''
559  003772  004767  006214                            JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
560  003776  132777  000020  015102          T5E:    BITB    #20,@TSRHI
561  004004  001401                                    BEQ     .+4
562  004006  104000                                    ERR                     ;STACKER FULL FLAG SET WITH ONLY STACKER
563                                                    ;FULL WHEN IN OVERFLOW MODE
564  004010                                            LOOP    T5E
(1)  004030  000137  003776                            JMP     @#T5E
565  004034  104400                                    TYPE
566  004036  016344                                    T5MSG5                  ;''PULL CARD PLATE OF STACKER #2 BACK TO
```

```
   567  004040  104400
   568  004042  016164
   569  004044  104400
   570  004046  016234
   571  004050  132777  000020  015030          T5F:  BITB    #20,@TSRHI
   572  004056  001774                                BEQ     T5F
   573  004060                                       CKTSR   #110000,T5F
   (1)          004060                               LP=.
   (1)  004060  012767  110000  014760              MOV     #110000,GOOD
   (1)  004066  017767  015012  014742              MOV     @TSR,BAD
   (1)  004074  026767  014746  014734              CMP     GOOD,BAD
   (1)  004102  001401                              BEQ     .+4
   (1)  004104  104003                              ERR+GB
   (1)  004106                                      LOOP    T5F
   (2)  004126  000137  004050                      JMP     @#T5F
   574  004132                                      CKTCR   #100000,LP
   (1)          004132                              LP=.
   (1)  004132  012767  100000  014706              MOV     #100000,GOOD
   (1)  004140  017767  014734  014670              MOV     @TCR,BAD
   (1)  004146  026767  014674  014662              CMP     GOOD,BAD
   (1)  004154  001401                              BEQ     .+4
   (1)  004156  104003                              ERR+GB
   (1)  004160                                      LOOP    LP
   (2)  004200  000137  004132                      JMP     @#LP
   575  004204  104400                              TYPE
   576  004206  016472                              T5MSG7
   577  004210  004767  005776                      JSR     PC,CONTIN
   578  004214  000005                              RESET
   579
   580  004216  000137  004222                      JMP     @#TEST6
   581                                              ;
   582
   583                                  ;FORCE INPUT CHECK ERROR BY TRYING TO FEED A CARD FROM AN EMPTY HOPPER
   584  004222  000005                              ;
   585  004224  104400                  TEST6:  RESET
   586  004226  016531                          TYPE
   587  004230  004767  005756                  T6MSG1
   588  004234  012777  020001  014636          JSR     PC,CONTIN
   589  004242  005067  014574                  MOV     #20001,@TCR
   590  004246  162767  000001  014566          CLR     CNT
   591  004254  001374                          SUB     #1,CNT
   592  004256  162767  000001  014556          BNE     .-6
   593  004264  001374                          SUB     #1,CNT
   594  004266  162767  000001  014546          BNE     .-6
   595  004274  001374                          SUB     #1,CNT
   596  004276  162767  000001  014536          BNE     .-6
   597  004304  001374                          CKTSR   #320,LP
   598  004306                                  LP=.
   (1)          004306                          MOV     #320,GOOD
   (1)  004306  012767  000320  014532          MOV     @TSR,BAD
   (1)  004314  017767  014564  014514          CMP     GOOD,BAD
   (1)  004322  026767  014520  014506          BEQ     .+4
   (1)  004330  001401                          ERR+GB
   (1)  004332  104003                          LOOP    LP
   (1)  004334                                  JMP     @#LP
   (2)  004354  000137  004306
```

```
  599  004360  000005                          RESET
  600
  601  004362  000207                          RTS     PC              ;EXIT LOGIC TEST
  602                                          .
  603                                  ;THIS TEST FEEDS 20 BLANK CARDS, 10 FROM EACH HOPPER, INTO THE STACKERS
  604                                  ;       (10 INTO EACH STACKER). THE CARDS ARE TESTED FOR ALL SPACES AND
  605                                  ;       TESTING IS DONE TO INSURE THAT THE INPUT DATA REQUEST CAN CAUSE
  606                                  ;       AN INTERRUPT.
  607                                  ;
  608  004364  000005                  PASS1:  RESET
  609  004366  104400                          TYPE
  610  004370  016570                          CRDTST                  ;"***** CARD TESTS *****"
  611  004372  104400                          TYPE
  612  004374  016623                          P1MSG1                  ;"REMOVE ALL CARDS AND PUT THE 8035-8045
  613  004376  004767  005610                  JSR     PC,CONTIN       ;"PRESS CR TO CONTINUE"
  614  004402  004767  007550                          JSR     PC,CKSWR
  615  004406  000005                          RESET
  616  004410  112777  000010  014464         MOVB    #10,@TCRHI      ;SELECT STACKER #1
  617  004416  104400                          TYPE
  618  004420  016705                          P1MSG2                  ;"PUT 10 BLANK CARDS IN EACH INPUT HOPPE
  619  004422  104400                          TYPE
  620  004424  016757                          P1MSG3                  ;"PUT 8035-8045 ON-LINE AND PRESS
  621                                                                  ; START ON THE 8035-8045"
  622  004426  004767  005560                  JSR     PC,CONTIN       ;"PRESS CR TO CONTINUE"
  623                                  ;DONE + SSO SHOULD BE THE ONLY BITS SET IN THE CMD REG
  624  004432                          LP:     CKTCR   #4200,LP        ;TEST FOR SSO + DONE FLAG
  (1)          004432                         LP=.
  (1)  004432  012767  004200  014406         MOV     #4200,GOOD
  (1)  004440  017767  014434  014370         MOV     @TCR,BAD        ;READ CMD REGISTER
  (1)  004446  026767  014374  014362         CMP     GOOD,BAD
  (1)  004454  001401                         BEQ     .+4
  (1)  004456  104003                         ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
  (1)  004460                                 LOOP    LP
  (2)  004500  000137  004432                 JMP     @#LP
  625                                  ;SEE IF ISSUING GO CLEARS THE DONE FLAG (READ 1ST CARD)
  626  004504  012777  024001  014366         MOV     #24001,@TCR     ;ISSUE HSO + SSO + GO
  627  004512                          LP:     CKTCR   #24000,LP       ;TEST FOR HSO + SSO
  (1)          004512                         LP=.
  (1)  004512  012767  024000  014326         MOV     #24000,GOOD
  (1)  004520  017767  014354  014310         MOV     @TCR,BAD        ;READ CMD REGISTER
  (1)  004526  026767  014314  014302         CMP     GOOD,BAD
  (1)  004534  001401                         BEQ     .+4
  (1)  004536  104003                         ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
  (1)  004540                                 LOOP    LP
  (2)  004560  000137  004512                 JMP     @#LP
  628  004564  005067  014252                 CLR     CNT
  629  004570  162767  000001  014244         SUB     #1,CNT
  630  004576  001374                         BNE     .-6
  631  004600  162767  000001  014234         SUB     #1,CNT
  632  004606  001374                         BNE     .-6             ;DELAY 1 SECOND
  633  004610                          LP:     CKTSR   #4011,LP        ;TEST FOR CIW + RDR
  (1)          004610                         LP=.
  (1)  004610  012767  004011  014230         MOV     #4011,GOOD
  (1)  004616  017767  014262  014212         MOV     @TSR,BAD        ;READ STATUS REGISTER
  (1)  004624  026767  014216  014204         CMP     GOOD,BAD
  (1)  004632  001401                         BEQ     .+4
```

```
   (1)  004634  104003
   (1)  004636
   (2)  004656  000137  004610
   634  004662
   (1)          004662
   (1)  004662  012767  000200  014156          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
   (1)  004670  017767  014220  014140          LOOP    LP
   (1)  004676  026767  014144  014132          JMP     @#LP
   (1)  004704  001401                                  CKTIR   #200,LP         ;TEST FOR IDR
   (1)  004706  104003                          LP=.
   (1)  004710                                  MOV     #200,GOOD
   (2)  004730  000137  004662                  MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
   635                                          CMP     GOOD,BAD
   636  004734  112777  000100  014136          BEQ     .+4
   637  004742  012777  005026  014150          ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
   638  004750  012777  000340  014144          LOOP    LP
   639  004756  012767  000040  173012          JMP     @#LP
   640  004764  000240          ;SEE IF INPUT DATA REQUEST WILL CAUSE AN INTERRUPT
   641  004766  000240          MOVB    #100,@TCR       ;ENABLE INT
   642  004770  012767  000340  173000          MOV     #P1INT,@PCV     ;SETUP PC VECTOR
   643  004776  104000          MOV     #340,@PSV       ;SETUP PS VECTOR
   644  005000          P1A:    MOV     #40,PS          ;SET BPU TO LEVEL #1
   (1)  005020  000137  004756  NOP
   645  005024  000402          NOP
   646  005026  062706  000004  MOV     #340,PS         ;SET BPU TO LEVEL #7
   647  005032  004767  005204  ERR                     ;INPUT DATA REQUEST DIDN'T CAUSE AN INTE
   648  005036                  LOOP    P1A
   (1)          005036          JMP     @#P1A
   (1)  005036  012767  004010  014002  BR      P1B
   (1)  005044  017767  014034  013764  P1INT:  ADD     #4,SP           ;HOUSE KEEPING ON STACK
   (1)  005052  026767  013770  013756  P1B:    JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
   (1)  005060  001401          CKTSR   #4010,LP        ;TEST FOR CIW
   (1)  005062  104003          LP=.
   (1)  005064                  MOV     #4010,GOOD
   (2)  005104  000137  005036  MOV     @TSR,BAD        ;READ STATUS REGISTER
   649  005110                  CMP     GOOD,BAD
   (1)          005110          BEQ     .+4
   (1)  005110  012767  024300  013730  ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
   (1)  005116  017767  013756  013712  LOOP    LP
   (1)  005124  026767  013716  013704  JMP     @#LP
   (1)  005132  001401          CKTCR   #24300,LP       ;TEST FOR HSO + SSO + DONE + INT EN
   (1)  005134  104003          LP=.
   (1)  005136                  MOV     #24300,GOOD
   (2)  005156  000137  005110  MOV     @TCR,BAD        ;READ CMD REGISTER
   650                          CMP     GOOD,BAD
   651  005162  012767  000010  013654  BEQ     .+4
   652  005170  012777  024001  013702  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
   653  005176                  LOOP    LP
   (1)  005176  032777  000200  013710  JMP     @#LP
   (1)  005204  001774          ;READ CARDS #2-9 FROM PRIMARY HOPPER
   654  005206                  MOV     #10,CNT1
   (1)          005206  P1D:    MOV     #24001,@TCR     ;ISSUE HSO + SSO + GO
   (1)  005206  012767  004011  013632  WTIDR
   (1)  005214  017767  013664  013614  BIT     #200,@TIR
   (1)  005222  026767  013620  013606  BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
   (1)  005230  001401          CKTSR   #4011,LP        ;TEST FOR CIW + RDR
                                LP=.
                                MOV     #4011,GOOD
                                MOV     @TSR,BAD        ;READ STATUS REGISTER
                                CMP     GOOD,BAD
                                BEQ     .+4
```

```
    (1)  005232  104003
    (1)  005234
    (2)  005254  000137  005206
655 005260  004767  004756
656 005264
    (1)          005264
    (1)  005264  012767  024200  013554
    (1)  005272  017767  013602  013536
    (1)  005300  026767  013542  013530
    (1)  005306  001401
    (1)  005310  104003
    (1)  005312
    (2)  005332  000137  005264
657 005336  005367  013502
658 005342  001312
659
660 005344  012777  024001  013526
661 005352
    (1)  005352  032777  000200  013534
    (1)  005360  001774
662 005362  004767  004654
663 005366
    (1)          005366
    (1)  005366  012767  004230  013452
    (1)  005374  017767  013504  013434
    (1)  005402  026767  013440  013426
    (1)  005410  001401
    (1)  005412  104003
    (1)  005414
    (2)  005434  000137  005366
664 005440
    (1)          005440
    (1)  005440  012767  124200  013400
    (1)  005446  017767  013426  013362
    (1)  005454  026767  013366  013354
    (1)  005462  001401
    (1)  005464  104003
    (1)  005466
    (2)  005506  000137  005440
665
666 005512  012777  044401  013360
667 005520
    (1)  005520  032777  000200  013366
    (1)  005526  001774
668 005530
    (1)          005530
    (1)  005530  012767  004011  013310
    (1)  005536  017767  013342  013272
    (1)  005544  026767  013276  013264
    (1)  005552  001401
    (1)  005554  104003
    (1)  005556
    (2)  005576  000137  005530
669 005602  004767  004434
670 005606
    (1)          005606
```

```
ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
LOOP    LP
JMP     @#LP
JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
CKTCR   #24200,LP       ;TEST FOR HS0 + SS0 + DONE
LP=.
MOV     #24200,GOOD
MOV     @TCR,BAD        ;READ CMD REGISTER
CMP     GOOD,BAD
BEQ     .+4
ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
LOOP    LP
JMP     @#LP
DEC     CNT1
BNE     P1D
;READ CARD #10 OUT OF PRIMARY HOPPER
MOV     #24001,@TCR     ;ISSUE HS0 + SS0 + GO
WTIDR
BIT     #200,@TIR
BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
LP=.
MOV     #4230,GOOD
MOV     @TSR,BAD        ;READ STATUS REGISTER
CMP     GOOD,BAD
BEQ     .+4
ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
LOOP    LP
JMP     @#LP
CKTCR   #124200,LP      ;TEST FOR ERR + HS0 + SS0 + DONE
LP=.
MOV     #124200,GOOD
MOV     @TCR,BAD        ;READ CMD REGISTER
CMP     GOOD,BAD
BEQ     .+4
ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
LOOP    LP
JMP     @#LP
;READ CARD #11 OUT OF SECONDARY HOPPER
MOV     #44401,@TCR     ;ISSUE CLEAR ERR + HS1 + SS0 + GO
WTIDR
BIT     #200,@TIR
BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
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 EX
LOOP    LP
JMP     @#LP
JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
CKTCR   #44200,LP       ;TEST FOR HS1 + SS0 + DONE
LP=.
```

```
    (1)  005606  012767  044200  013232          MOV     #44200,GOOD
    (1)  005614  017767  013260  013214          MOV     @TCR,BAD        ;READ CMD REGISTER
    (1)  005622  026767  013220  013206          CMP     GOOD,BAD
    (1)  005630  001401                           BEQ     .+4
    (1)  005632  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
    (1)  005634                                   LOOP    LP
    (2)  005654  000137  005606                   JMP     @#LP
    671                             ;READ CARDS #12-19 AND PUT THEM IN STACKER #2
    672  005660  012767  000010  013156          MOV     #10,CNT1
    673  005666  012777  054001  013204  P1E:    MOV     #54001,@TCR     ;ISSUE HS1 + SS1 + GO
    674  005674                                   WTIDR
    (1)  005674  032777  000200  013212          BIT     #200,@TIR
    (1)  005702  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
    675  005704  004767  004332                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
    676  005710                                   WTDONE
    (1)  005710  032777  000200  013162          BIT     #200,@TCR
    (1)  005716  001774                           BEQ     .-6             ;WAIT FOR DONE FLAG
    677  005720                                   CKTSR   #4010,LP        ;TEST FOR CIW
    (1)          005720                           LP=.
    (1)  005720  012767  004010  013120          MOV     #4010,GOOD
    (1)  005726  017767  013152  013102          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  005734  026767  013106  013074          CMP     GOOD,BAD
    (1)  005742  001401                           BEQ     .+4
    (1)  005744  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
    (1)  005746                                   LOOP    LP
    (2)  005766  000137  005720                   JMP     @#LP
    678  005772                                   CKTCR   #54200,LP       ;TEST FOR HS1 + SS1 + DONE
    (1)          005772                           LP=.
    (1)  005772  012767  054200  013046          MOV     #54200,GOOD
    (1)  006000  017767  013074  013030          MOV     @TCR,BAD        ;READ CMD REGISTER
    (1)  006006  026767  013034  013022          CMP     GOOD,BAD
    (1)  006014  001401                           BEQ     .+4
    (1)  006016  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
    (1)  006020                                   LOOP    LP
    (2)  006040  000137  005772                   JMP     @#LP
    679  006044  005367  012774                   DEC     CNT1
    680  006050  001306                           BNE     P1E
    681                             ;READ CARD #20 OUT OF SECONDARY HOPPER
    682  006052  012777  050001  013020          MOV     #50001,@TCR     ;ISSUE HS1 + SS1 + GO
    683  006060                                   WTIDR
    (1)  006060  032777  000200  013026          BIT     #200,@TIR
    (1)  006066  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
    684  006070  004767  004146                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
    685  006074                                   CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
    (1)          006074                           LP=.
    (1)  006074  012767  004230  012744          MOV     #4230,GOOD
    (1)  006102  017767  012776  012726          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  006110  026767  012732  012720          CMP     GOOD,BAD
    (1)  006116  001401                           BEQ     .+4
    (1)  006120  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
    (1)  006122                                   LOOP    LP
    (2)  006142  000137  006074                   JMP     @#LP
    686  006146                                   CKTCR   #150200,LP      ;TEST FOR ERR + HS1 + SS1 + DONE
    (1)          006146                           LP=.
    (1)  006146  012767  150200  012672          MOV     #150200,GOOD
    (1)  006154  017767  012720  012654          MOV     @TCR,BAD        ;READ CMD REGISTER
```

```
    (1)  006162  026767  012660  012646          CMP     GOOD,BAD
    (1)  006170  001401                          BEQ     .+4
    (1)  006172  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
    (1)  006174                                  LOOP    LP
    (2)  006214  000137  006146                  JMP     @#LP
    687                                  ;PUT CARD #20 IN STACKER #2
    688  006220  012777  010401  012652          MOV     #10401,@TCR     ;ISSUE CLEAR ERR + SS1 + GO
    689  006226                                  WTDONE
    (1)  006226  032777  000200  012644          BIT     #200,@TCR
    (1)  006234  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
    690  006236                                  CKTSR   #0,LP           ;TEST FOR STATUS=0
    (1)          006236                          LP=.
    (1)  006236  012767  000000  012602          MOV     #0,GOOD
    (1)  006244  017767  012634  012564          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  006252  026767  012570  012556          CMP     GOOD,BAD
    (1)  006260  001401                          BEQ     .+4
    (1)  006262  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
    (1)  006264                                  LOOP    LP
    (2)  006304  000137  006236                  JMP     @#LP
    691  006310                                  CKTCR   #10200,LP       ;TEST FOR SS1 +DONE
    (1)          006310                          LP=.
    (1)  006310  012767  010200  012530          MOV     #10200,GOOD
    (1)  006316  017767  012556  012512          MOV     @TCR,BAD        ;READ CMD REGISTER
    (1)  006324  026767  012516  012504          CMP     GOOD,BAD
    (1)  006332  001401                          BEQ     .+4
    (1)  006334  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
    (1)  006336                                  LOOP    LP
    (2)  006356  000137  006310                  JMP     @#LP
    692
    693  006362  000137  006366                  JMP     @#PASS2
    694                                          ;
    695                                  ;THIS TEST PUNCHES THE ENTIRE CHAR SET ON EACH CARD IN A PRECESSED
    696                                  ;       FASHION. THIS MEANS THE 1ST PUNCHED CHAR ON CARD #1 IS IN
    697                                  ;       COLUMN #11 AND THE 1ST PUNCHED CHAR OF CARD #20 IS IN COLUMN
    698                                  ;       #31. ALSO TEST THAT OUTPUT DATA REQUEST CAN CAUSE AN INTERRUPT.
    699                                  ;
    700  006366  104400                  PASS2:  TYPE
    701  006370  017050                          P2MSG1                  ;''VERIFY THAT THERE ARE 10 BLANK CARDS I
    702  006372  104400                          TYPE
    703  006374  017136                          P2MSG2                  ;''PLACE ALL 20 CARDS IN THE PRIMARY HOPP
    704  006376  104400                          TYPE
    705  006400  017210                          P2MSG3                  ;''PRESS STOP-RESET  AND PRESS START ON T
    706  006402  004767  003604                  JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
    707  006406  004767  003762                  JSR     PC,CLRBFS       ;PREPARE BUFFER FOR PUNCHING
    708                                  ;FEED CARD #1
    709  006412  012777  020401  012460          MOV     #20401,@TCR     ;ISSUE CLEAR ERR + HSO + GO
    710  006420                                  WTIDR
    (1)  006420  032777  000200  012466          BIT     #200,@TIR
    (1)  006426  001774                          BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
    711  006430  004767  003606                  JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
    712  006434                                  WTDONE
    (1)  006434  032777  000200  012436          BIT     #200,@TCR
    (1)  006442  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
    713  006444                                  CKTSR   #4010,LP        ;TEST FOR CIW
    (1)          006444                          LP=.
    (1)  006444  012767  004010  012374          MOV     #4010,GOOD
```

```
   (1)  006452  017767  012426  012356          MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  006460  026767  012362  012350          CMP     GOOD,BAD
   (1)  006466  001401                           BEQ     .+4
   (1)  006470  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
   (1)  006472                                   LOOP    LP
   (2)  006512  000137  006444                   JMP     @#LP
   714  006516                                   CKTCR   #20200,LP       ;TEST FOR HSO + DONE
   (1)               006516                      LP=.
   (1)  006516  012767  020200  012322          MOV     #20200,GOOD
   (1)  006524  017767  012350  012304          MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  006532  026767  012310  012276          CMP     GOOD,BAD
   (1)  006540  001401                           BEQ     .+4
   (1)  006542  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
   (1)  006544                                   LOOP    LP
   (2)  006564  000137  006516                   JMP     @#LP
   715                                  ;FEED CARDS #2-19, INHIBIT READ FOR THESE CARDS
   716  006570  012767  020104  012270          MOV     #BF1,PTR
   717  006576  012767  000022  012240          MOV     #22,CNT1        ;CNT1=18
   718  006604  012777  020027  012266  P2A:    MOV     #20027,@TCR     ;ISSUE HSO + IR + PRI + PUN + GO
   719  006612                                   WTODR
   (1)  006612  132777  000200  012276          BITB    #200,@TIRHI
   (1)  006620  001774                           BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
   720  006622                                           CKTSR   #4410,LP        ;TEST FOR PUDR
   (1)               006622                      LP=.
   (1)  006622  012767  004410  012216          MOV     #4410,GOOD
   (1)  006630  017767  012250  012200          MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  006636  026767  012204  012172          CMP     GOOD,BAD
   (1)  006644  001401                           BEQ     .+4
   (1)  006646  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
   (1)  006650                                   LOOP    LP
   (2)  006670  000137  006622                   JMP     @#LP
   721  006674  004767  003534                   JSR     PC,P2SEND       ;SEND DATA TO PUNCH
   722  006700  162767  000002  012160          SUB     #2,PTR
   723  006706                                   WTDONE
   (1)  006706  032777  000200  012164          BIT     #200,@TCR
   (1)  006714  001774                           BEQ     .-6             ;WAIT FOR DONE FLAG
   724  006716                                   CKTSR   #4010,LP        ;TEST FOR CIW
   (1)               006716                      LP=.
   (1)  006716  012767  004010  012122          MOV     #4010,GOOD
   (1)  006724  017767  012154  012104          MOV     @TSR,BAD        ;READ STATUS REGISTER
   (1)  006732  026767  012110  012076          CMP     GOOD,BAD
   (1)  006740  001401                           BEQ     .+4
   (1)  006742  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
   (1)  006744                                   LOOP    LP
   (2)  006764  000137  006716                   JMP     @#LP
   725  006770                                   CKTCR   #20226,LP       ;TEST FOR HSO + DONE + IR + PRI + PUN
   (1)               006770                      LP=.
   (1)  006770  012767  020226  012050          MOV     #20226,GOOD
   (1)  006776  017767  012076  012032          MOV     @TCR,BAD        ;READ CMD REGISTER
   (1)  007004  026767  012036  012024          CMP     GOOD,BAD
   (1)  007012  001401                           BEQ     .+4
   (1)  007014  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
   (1)  007016                                   LOOP    LP
   (2)  007036  000137  006770                   JMP     @#LP
   726  007042                                   CKTIR   #0,LP           ;TEST FOR 0
   (1)               007042                      LP=.
```

```
    (1)  007042  012767  000000  011776          MOV     #0,GOOD
    (1)  007050  017767  012040  011760          MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
    (1)  007056  026767  011764  011752          CMP     GOOD,BAD
    (1)  007064  001401                           BEQ     .+4
    (1)  007066  104003                           ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
    (1)  007070                                   LOOP    LP
    (2)  007110  000137  007042                   JMP     @#LP
   727   007114  005367  011724                   DEC     CNT1            ;IF ALL CARDS ARE NOT DONE
   728   007120  001231                           BNE     P2A             ;GO PUNCH ANOTHER CARD
   729                                  ;FEED CARD #20
   730   007122  012777  020007  011750          MOV     #20007,@TCR     ;ISSUE HSO + PRI + PUN + GO
   731   007130                                   WTODR
    (1)  007130  132777  000200  011760          BITB    #200,@TIRHI
    (1)  007136  001774                           BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
   732   007140  004767  003270                   JSR     PC,P2SEND       ;SEND DATA TO PUNCH
   733   007144  162767  000002  011714          SUB     #2,PTR
   734   007152                                   WTIDR
    (1)  007152  032777  000200  011734          BIT     #200,@TIR
    (1)  007160  001774                           BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
   735   007162  004767  003054                   JSR     PC,BLKCRD       ;TEST FOR BLANK CARD
   736   007166                                   WTDONE
    (1)  007166  032777  000200  011704          BIT     #200,@TCR
    (1)  007174  001774                           BEQ     .-6             ;WAIT FOR DONE FLAG
   737   007176                                   CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
    (1)          007176                          LP=.
    (1)  007176  012767  004230  011642          MOV     #4230,GOOD
    (1)  007204  017767  011674  011624          MOV     @TSR,BAD        ;READ STATUS REGISTER
    (1)  007212  026767  011630  011616          CMP     GOOD,BAD
    (1)  007220  001401                           BEQ     .+4
    (1)  007222  104003                           ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
    (1)  007224                                   LOOP    LP
    (2)  007244  000137  007176                   JMP     @#LP
   738   007250                                   CKTCR   #120206,LP      ;TEST FOR ERR + HSO + DONE + PRI + PUN
    (1)          007250                          LP=.
    (1)  007250  012767  120206  011570          MOV     #120206,GOOD
    (1)  007256  017767  011616  011552          MOV     @TCR,BAD        ;READ CMD REGISTER
    (1)  007264  026767  011556  011544          CMP     GOOD,BAD
    (1)  007272  001401                           BEQ     .+4
    (1)  007274  104003                           ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
    (1)  007276                                   LOOP    LP
    (2)  007316  000137  007250                   JMP     @#LP
   739                                  ;CARD #20 PUNCH AND PRINT
   740   007322  012777  000407  011550          MOV     #407,@TCR       ;ISSUE CLEAR ERR + PRI + PUN + GO
   741   007330                                   WTODR
    (1)  007330  132777  000200  011560          BITB    #200,@TIRHI
    (1)  007336  001774                           BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
   742   007340                                   CKTIR   #100000,LP      ;TEST FOR ODR
    (1)          007340                          LP=.
    (1)  007340  012767  100000  011500          MOV     #100000,GOOD
    (1)  007346  017767  011542  011462          MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
    (1)  007354  026767  011466  011454          CMP     GOOD,BAD
    (1)  007362  001401                           BEQ     .+4
    (1)  007364  104003                           ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
    (1)  007366                                   LOOP    LP
    (2)  007406  000137  007340                   JMP     @#LP
   743   007412                                   CKTCR   #6,LP           ;TEST FOR PRI + PUN
```

```
(1)             007412
(1)  007412  012767  000006  011426          LP=.
(1)  007420  017767  011454  011410          MOV     #6,GOOD
(1)  007426  026767  011414  011402          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  007434  001401                          CMP     GOOD,BAD
(1)  007436  104003                          BEQ     .+4
(1)  007440                                  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
(2)  007460  000137  007412                  LOOP    LP
                                             JMP     @#LP
744                                          ;SEE IF OUTPUT DATA REQUEST WILL CAUSE AN INTERRUPT
745  007464  152777  000100  011406          BISB    #100,@TCR       ;ENABLE INTERRUPT
746  007472  012777  007556  011420          MOV     #P2INT,@PCV     ;SETUP PC VECTOR
747  007500  012777  000340  011414          MOV     #340,@PSV       ;SETUP PS VECTOR
748  007506  012767  000040  170262          MOV     #40,PS          ;SET BPU TO LEVEL #1
749  007514  000240                          NOP
750  007516  000240                          NOP
751  007520  012767  000340  170250          MOV     #340,PS         ;SET BPU TO LEVEL #7
752  007526  104000                  P2B:    ERR                     ;OUTPUT DATA REQUEST DIDN'T CAUSE AN INT
753  007530                                  LOOP    P2B
(1)  007550  000137  007526                  JMP     @#P2B
754  007554  000402                          BR      P2C
755  007556  062706  000004          P2INT:  ADD     #4,SP           ;HOUSE KEEPING ON STACK
756  007562  004767  002646          P2C:    JSR     PC,P2SEND       ;SEND DATA TO PUNCH
757  007566                                  WTDONE
(1)  007566  032777  000200  011304          BIT     #200,@TCR
(1)  007574  001774                          BEQ     .-6             ;WAIT FOR DONE FLAG
758  007576                                  CKTSR   #0,LP           ;TEST FOR STATUS =0
(1)             007576                       LP=.
(1)  007576  012767  000000  011242          MOV     #0,GOOD
(1)  007604  017767  011274  011224          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  007612  026767  011230  011216          CMP     GOOD,BAD
(1)  007620  001401                          BEQ     .+4
(1)  007622  104003                          ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
(1)  007624                                  LOOP    LP
(2)  007644  000137  007576                  JMP     @#LP
759  007650                                  CKTCR   #306,LP         ;TEST FOR DONE + INT EN + PRI + PUN
(1)             007650                       LP=.
(1)  007650  012767  000306  011170          MOV     #306,GOOD
(1)  007656  017767  011216  011152          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  007664  026767  011156  011144          CMP     GOOD,BAD
(1)  007672  001401                          BEQ     .+4
(1)  007674  104003                          ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
(1)  007676                                  LOOP    LP
(2)  007716  000137  007650                  JMP     @#LP
760
761  007722  000137  007726                  JMP     @#PASS3
762                                          ;
763                                  ;THIS TEST PERFORMS PUNCHING WITHOUT PRINTING. ONE ALL "A"S LETTER IS
764                                  ;       PUNCHED IN COLUMN 1 OF CARD 1 AND SO ON UNTIL 20 ALL "A"S LETTER
765                                  ;       ARE PUNCHED IN THE 1ST 20 COLUMNS OF CARD 20. THIS TEST
766                                  ;       VERIFIES THE DATA READ FROM PUNCHING DONE IN PASS 2.
767                                  ;
768  007726  104400                  PASS3:  TYPE
769  007730  017334                          P3MSG1                  ;"VERIFY THAT ALL CHARS WERE PUNCHED ON
770  007732  104400                          TYPE
771  007734  017422                          P3MSG2                  ;PRECESS FASHION BEGINNING WITH COLUMN 1
772  007736  104400                          TYPE
```

```
                                P2MSG2                  ;''PLACE ALL 20 CARDS IN THE PRIMARY HOPP
                                TYPE
                                P2MSG3                  ;''PRESS STOP-RESET  AND PRESS START ON T
                                JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
                            ;FEED CARD #1
                                MOV     #BF1,INPTR
                                MOV     #20401,@TCR     ;ISSUE CLEAR ERR + HSO + GO
                                WTIDR
                                BIT     #200,@TIR
                                BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
                                JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASS 2
                                SUB     #2,INPTR
                                WTDONE
                                BIT     #200,@TCR
                                BEQ     .-6             ;WAIT FOR DONE FLAG
                                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 EX
                                LOOP    LP
                                JMP     @#LP
                                CKTCR   #20200,LP       ;TEST FOR HSO + DONE
                                LP=.
                                MOV     #20200,GOOD
                                MOV     @TCR,BAD        ;READ CMD REGISTER
                                CMP     GOOD,BAD
                                BEQ     .+4
                                ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                                LOOP    LP
                                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
                                MOV     #22,CNT1        ;CNT1=18
                                MOV     #1,CARD         ;CARD=1
                            P3A:MOV     #20003,@TCR     ;ISSUE HSO + PUN + GO
                                WTODR
                                BITB    #200,@TIRHI
                                BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
                                JSR     PC,P3SEND       ;SEND DATA TO PUNCH
                                ADD     #1,CARD
                                WTIDR
                                BIT     #200,@TIR
                                BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
                                JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASS 2
                                SUB     #2,INPTR
                                WTDONE
                                BIT     #200,@TCR
                                BEQ     .-6             ;WAIT FOR DONE FLAG
                                CKTCR   #20202,LP       ;TEST FOR HSO + DONE + PUN
                                LP=.
                                MOV     #20202,GOOD
                                MOV     @TCR,BAD        ;READ CMD REGISTER
                                CMP     GOOD,BAD
```

```
                                    773  007740  017136
                                    774  007742  104400
                                    775  007744  017210
                                    776  007746  004767  002240
                                    777
                                    778  007752  012767  020104  011072
                                    779  007760  012777  020401  011112
                                    780  007766
                                    (1)  007766  032777  000200  011120
                                    (1)  007774  001774
                                    781  007776  004767  002776
                                    782  010002  162767  000002  011042
                                    783  010010
                                    (1)  010010  032777  000200  011062
                                    (1)  010016  001774
                                    784  010020
                                    (1)          010020
                                    (1)  010020  012767  004010  011020
                                    (1)  010026  017767  011052  011002
                                    (1)  010034  026767  011006  010774
                                    (1)  010042  001401
                                    (1)  010044  104003
                                    (1)  010046
                                    (2)  010066  000137  010020
                                    785  010072
                                    (1)          010072
                                    (1)  010072  012767  020200  010746
                                    (1)  010100  017767  010774  010730
                                    (1)  010106  026767  010734  010722
                                    (1)  010114  001401
                                    (1)  010116  104003
                                    (1)  010120
                                    (2)  010140  000137  010072
                                    786
                                    787
                                    788  010144  012767  000022  010672
                                    789  010152  012767  000001  010660
                                    790  010160  012777  020003  010712
                                    791  010166
                                    (1)  010166  132777  000200  010722
                                    (1)  010174  001774
                                    792  010176  004767  002306
                                    793  010202  062767  000001  010630
                                    794  010210
                                    (1)  010210  032777  000200  010676
                                    (1)  010216  001774
                                    795  010220  004767  002554
                                    796  010224  162767  000002  010620
                                    797  010232
                                    (1)  010232  032777  000200  010640
                                    (1)  010240  001774
                                    798  010242
                                    (1)          010242
                                    (1)  010242  012767  020202  010576
                                    (1)  010250  017767  010624  010560
                                    (1)  010256  026767  010564  010552
```

```
    (1)  010264  001401
    (1)  010266  104003
    (1)  010270
    (2)  010310  000137  010242
    799  010314
    (1)            010314
    (1)  010314  012767  004010  010524
    (1)  010322  017767  010556  010506
    (1)  010330  026767  010512  010500
    (1)  010336  001401
    (1)  010340  104003
    (1)  010342
    (2)  010362  000137  010314
    800  010366
    (1)            010366
    (1)  010366  012767  000000  010452
    (1)  010374  017767  010514  010434
    (1)  010402  026767  010440  010426
    (1)  010410  001401
    (1)  010412  104003
    (1)  010414
    (2)  010434  000137  010366
    801  010440  005367  010400
    802  010444  001245
    803
    804  010446  012777  020003  010424
    805  010454
    (1)  010454  132777  000200  010434
    (1)  010462  001774
    806  010464  004767  002020
    807  010470  062767  000001  010342
    808  010476
    (1)  010476  032777  000200  010410
    (1)  010504  001774
    809  010506  004767  002266
    810  010512
    (1)  010512  032777  000200  010360
    (1)  010520  001774
    811  010522
    (1)            010522
    (1)  010522  012767  004230  010316
    (1)  010530  017767  010350  010300
    (1)  010536  026767  010304  010272
    (1)  010544  001401
    (1)  010546  104003
    (1)  010550
    (2)  010570  000137  010522
    812  010574
    (1)            010574
    (1)  010574  012767  120202  010244
    (1)  010602  017767  010272  010226
    (1)  010610  026767  010232  010220
    (1)  010616  001401
    (1)  010620  104003
    (1)  010622
    (2)  010642  000137  010574
```

```
                BEQ     .+4
                ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                LOOP    LP
                JMP     @#LP
                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 EX
                LOOP    LP
                JMP     @#LP
                CKTIR   #0,LP           ;TEST INTERRUPT REG FOR 0
                LP=.
                MOV     #0,GOOD
                MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
                CMP     GOOD,BAD
                BEQ     .+4
                ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
                LOOP    LP
                JMP     @#LP
                DEC     CNT1
                BNE     P3A             ;GO PUNCH AND READ MORE CARDS
;FEED CARD #20
                MOV     #20003,@TCR     ;ISSUE HSO + PUN + GO
                WTODR
                BITB    #200,@TIRHI
                BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
                JSR     PC,P3SEND       ;SEND DATA TO PUNCH
                ADD     #1,CARD
                WTIDR
                BIT     #200,@TIR
                BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
                JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASS 2
                WTDONE
                BIT     #200,@TCR
                BEQ     .-6             ;WAIT FOR DONE FLAG
                CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
                LP=.
                MOV     #4230,GOOD
                MOV     @TSR,BAD        ;READ STATUS REGISTER
                CMP     GOOD,BAD
                BEQ     .+4
                ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                LOOP    LP
                JMP     @#LP
                CKTCR   #120202,LP      ;TEST FOR ERR + HSO  + PUN + DONE
                LP=.
                MOV     #120202,GOOD
                MOV     @TCR,BAD        ;READ CMD REGISTER
                CMP     GOOD,BAD
                BEQ     .+4
                ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                LOOP    LP
                JMP     @#LP
```

```
                                                ;PUNCH CARD #20
                                                        MOV     #3,@TCR         ;ISSUE PUN + GO
                                                        WTODR
                                                        BITB    #200,@TIRHI
                                                        BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
                                                        JSR     PC,P3SEND       ;SEND DATA TO PUNCH
                                                        WTDONE
                                                        BIT     #200,@TCR
                                                        BEQ     .-6             ;WAIT FOR DONE FLAG
                                                        CKTSR   #0,LP           ;TEST FOR STATUS=0
                                                        LP=.
                                                        MOV     #0,GOOD
                                                        MOV     @TSR,BAD        ;READ STATUS REGISTER
                                                        CMP     GOOD,BAD
                                                        BEQ     .+4
                                                        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                                                        LOOP    LP
                                                        JMP     @#LP
                                                        CKTCR   #000202,LP      ;TEST FOR ERR + DONE
                                                        LP=.
                                                        MOV     #000202,GOOD
                                                        MOV     @TCR,BAD        ;READ CMD REGISTER
                                                        CMP     GOOD,BAD
                                                        BEQ     .+4
                                                        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                                                        LOOP    LP
                                                        JMP     @#LP
                                                        JMP     @#PASS4
                                                        ;
                                                ;THIS TEST PERFORMS A SEPARATE PRINT OPERATION ON ALL CARDS BY
                                                ;       PRINTING ''CARD 01'' IN COLUMNS 1-7 OF CARD 1 UP TO ''CARD 20'' IN
                                                ;       COLUMNS 1-7 OF CARD 20. VERIFY CORRECT DATA READ FROM PUNCHING
                                                ;       DONE IN PASSES 2 + 3
                                                ;
                                                PASS4:  TYPE
                                                        P4MSG1                  ;''VERIFY THAT THERE IS THE LETTER ''A''
                                                                                ; PUNCHED IN CARD #1
                                                        TYPE
                                                        P4MSG2                  ;THAT IS HOLE PUNCHED IN ROW 12 AND 1
                                                        TYPE
                                                        P4MSG3                  ;UP TO 20 ALL A'S CHARS PUNCHED IN CARD
                                                        TYPE
                                                        P2MSG2                  ;''PLACE ALL 20 CARDS IN THE PRIMARY HOPP
                                                        TYPE
                                                        P2MSG3                  ;''PRESS STOP-RESET  AND PRESS START ON T
                                                        JSR     PC,CONTIN       ;''PRESS CR TO CONTINUE''
                                                        JSR     PC,SETBFS       ;PREPARE BUFFER FOR DATA COMPARE
                                                ;FEED CARD #1
                                                        .EVEN
                                                        MOV     #BF1,INPTR
                                                        MOV     #20401,@TCR     ;ISSUE CLEAR ERR + HSO + GO
                                                        WTIDR
                                                        BIT     #200,@TIR
                                                        BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
                                                        JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASSES 2 + 3
                                                        SUB     #2,INPTR
```

```
847  011122
(1)  011122  032777  000200  007750          WTDONE
(1)  011130  001774          BIT     #200,@TCR
848  011132                  BEQ     .-6             ;WAIT FOR DONE FLAG
(1)       011132             CKTSR   #4010,LP        ;TEST FOR CIW
(1)  011132  012767  004010  007706          LP=.
(1)  011140  017767  007740  007670          MOV     #4010,GOOD
(1)  011146  026767  007674  007662          MOV     @TSR,BAD        ;READ STATUS REGISTER
(1)  011154  001401          CMP     GOOD,BAD
(1)  011156  104003          BEQ     .+4
(1)  011160                  ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
(2)  011200  000137  011132  LOOP    LP
849  011204                  JMP     @#LP
(1)       011204             CKTCR   #20200,LP       ;TEST FOR HSO + DONE
(1)  011204  012767  020200  007634          LP=.
(1)  011212  017767  007662  007616          MOV     #20200,GOOD
(1)  011220  026767  007622  007610          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  011226  001401          CMP     GOOD,BAD
(1)  011230  104003          BEQ     .+4
(1)  011232                  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
(2)  011252  000137  011204  LOOP    LP
850                          JMP     @#LP
851  011256  012767  000022  007560  ;FEED CARDS #2-19, CHECKING INPUT DATA AND PRINTING IN COLUMNS 1-7
852  011264  012767  000001  007546          MOV     #22,CNT1        ;CNT1=18
853  011272  012767  020404  007562          MOV     #1,CARD         ;CARD=1
854  011300  012777  020017  007572          MOV     #CARD01,OUTPTR  ;OUTPTR=CARD01
855  011306  P4A:    MOV     #20017,@TCR     ;ISSUE HSO + SPD + PRI + PUN + GO
(1)  011306  132777  000200  007602          WTODR
(1)  011314  001774          BITB    #200,@TIRHI
856  011316  004767  001262  BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
857  011322  004767  001346          JSR     PC,P5SEND       ;SEND DATA TO PUNCH
858  011326  062767  000001  007504  JSR     PC,P4SEND       ;SEND DATA TO PRINT
859  011334          ADD     #1,CARD
(1)  011334  032777  000200  007552  WTIDR
(1)  011342  001774          BIT     #200,@TIR
860  011344  032777  000200  007542  BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
861  011352  001774          BIT     #200,@TIR
862  011354  004767  001420  BEQ     .-6             ;WAIT FOR DONE FLAG
863  011360  162767  000002  007464  JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASSES 2 + 3
864  011366          SUB     #2,INPTR
(1)  011366  032777  000200  007504  WTDONE
(1)  011374  001774          BIT     #200,@TCR
865  011376                  BEQ     .-6             ;WAIT FOR DONE FLAG
(1)       011376             CKTCR   #20216,LP       ;TEST FOR HSO + DONE + SPD + PRI + PUN
(1)  011376  012767  020216  007442          LP=.
(1)  011404  017767  007470  007424          MOV     #20216,GOOD
(1)  011412  026767  007430  007416          MOV     @TCR,BAD        ;READ CMD REGISTER
(1)  011420  001401          CMP     GOOD,BAD
(1)  011422  104003          BEQ     .+4
(1)  011424                  ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
(2)  011444  000137  011376  LOOP    LP
866  011450                  JMP     @#LP
(1)       011450             CKTSR   #4010,LP        ;TEST FOR CIW
(1)  011450  012767  004010  007370          LP=.
(1)  011456  017767  007422  007352          MOV     #4010,GOOD
(1)  011464  026767  007356  007344          MOV     @TSR,BAD        ;READ STATUS REGISTER
                             CMP     GOOD,BAD
```

```
    (1)  011472  001401
    (1)  011474  104003
    (1)  011476
    (2)  011516  000137  011450
    867  011522
    (1)               011522
    (1)  011522  012767  000000  007316
    (1)  011530  017767  007360  007300
    (1)  011536  026767  007304  007272
    (1)  011544  001401
    (1)  011546  104003
    (1)  011550
    (2)  011570  000137  011522
    868  011574  005367  007244
    869  011600  001237
    870
    871  011602  012777  020017  007270
    872  011610
    (1)  011610  132777  000200  007300
    (1)  011616  001774
    873  011620  132777  000200  007270
    874  011626  001774
    875  011630  062767  000001  007202
    876  011636  004767  000742
    877  011642  004767  001026
    878  011646
    (1)  011646  032777  000200  007240
    (1)  011654  001774
    879  011656  004767  001116
    880  011662  162767  000002  007162
    881  011670
    (1)  011670  032777  000200  007202
    (1)  011676  001774
    882  011700
    (1)               011700
    (1)  011700  012767  004230  007140
    (1)  011706  017767  007172  007122
    (1)  011714  026767  007126  007114
    (1)  011722  001401
    (1)  011724  104003
    (1)  011726
    (2)  011746  000137  011700
    883  011752
    (1)               011752
    (1)  011752  012767  120216  007066
    (1)  011760  017767  007114  007050
    (1)  011766  026767  007054  007042
    (1)  011774  001401
    (1)  011776  104003
    (1)  012000
    (2)  012020  000137  011752
    884
    885  012024  012777  000417  007046
    886  012032
    (1)  012032  132777  000200  007056
    (1)  012040  001774
```

```
                BEQ     .+4
                ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                LOOP    LP
                JMP     @#LP
                CKTIR   #0,LP           ;TEST INTERRUPT REG FOR 0
                LP=.
                MOV     #0,GOOD
                MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
                CMP     GOOD,BAD
                BEQ     .+4
                ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
                LOOP    LP
                JMP     @#LP
                DEC     CNT1
                BNE     P4A             ;GO PRINT AND READ MORE CARDS
;FEED CARD #20
                MOV     #20017,@TCR     ;ISSUE HSO + SPD + PRI + PUN + GO
                WTODR
                BITB    #200,@TIRHI
                BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
                BITB    #200,@TIRHI
                BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
                ADD     #1,CARD
                JSR     PC,P5SEND       ;SEND DATA TO PUNCH
                JSR     PC,P4SEND       ;SEND DATA TO PRINTER
                WTIDR
                BIT     #200,@TIR
                BEQ     .-6             ;WAIT FOR INPUT DATA REQUEST
                JSR     PC,CKDATA       ;CHECK INPUT DATA FROM PASSES 2 + 3
                SUB     #2,INPTR
                WTDONE
                BIT     #200,@TCR
                BEQ     .-6             ;WAIT FOR DONE FLAG
                CKTSR   #4230,LP        ;TEST FOR CIW + IE + HE
                LP=.
                MOV     #4230,GOOD
                MOV     @TSR,BAD        ;READ STATUS REGISTER
                CMP     GOOD,BAD
                BEQ     .+4
                ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
                LOOP    LP
                JMP     @#LP
                CKTCR   #120216,LP      ;TEST FOR ERR + HSO  + SPD + PRI + PUN +
                LP=.
                MOV     #120216,GOOD
                MOV     @TCR,BAD        ;READ CMD REGISTER
                CMP     GOOD,BAD
                BEQ     .+4
                ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
                LOOP    LP
                JMP     @#LP
;PRINT CARD #20
                MOV     #000417,@TCR    ;ISSUE CLEAR ERR + SPD + PRI + PUN + GO
                WTODR
                BITB    #200,@TIRHI
                BEQ     .-6             ;WAIT FOR OUTPUT DATA REQUEST
```

```
   887  012042  004767  000536
   888  012046  004767  000622
   889  012052
   (1)  012052  032777  000200  007020
   (1)  012060  001774
   890  012062
   (1)          012062
   (1)  012062  012767  000000  006756
   (1)  012070  017767  007010  006740
   (1)  012076  026767  006744  006732
   (1)  012104  001401
   (1)  012106  104003
   (1)  012110
   (2)  012130  000137  012062
   891  012134
   (1)          012134
   (1)  012134  012767  000216  006704
   (1)  012142  017767  006732  006666
   (1)  012150  026767  006672  006660
   (1)  012156  001401
   (1)  012160  104003
   (1)  012162
   (2)  012202  000137  012134
   892
   893  012206  000005
   894  012210  000207
   895
   896
   897
   898  012212  104400
   899  012214  015550
   900  012216  004767  001734
   901  012222  004767  002642
   902  012226  022704  000015
   903  012232  001373
   904  012234  004767  002554
   905  012240  000207
   906
   907
   908
   909  012242  012700  000001
   910  012246  004767  001704
   911  012252  032777  000200  006634
   912  012260  001774
   913  012262  012767  000000  006556
   914  012270  017767  006614  006540
   915  012276  026767  006544  006532
   916  012304  001401
   917  012306  104007
   918  012310  005200
   919  012312  022700  000121
   920  012316  001355
   921  012320
   (1)          012320
   (1)  012320  012767  000000  006520
   (1)  012326  017767  006562  006502
```

```
        JSR     PC,P5SEND       ;SEND DATA TO PUNCH
        JSR     PC,P4SEND       ;SEND DATA TO PRINTER
WTDONE
        BIT     #200,@TCR
        BEQ     .-6             ;WAIT FOR DONE FLAG
        CKTSR   #0,LP           ;TEST FOR STATUS=0
        LP=.
        MOV     #0,GOOD
        MOV     @TSR,BAD        ;READ STATUS REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;STATUS REG - VALUE READ NOT EQUAL TO EX
        LOOP    LP
        JMP     @#LP
        CKTCR   #216,LP         ;TEST FOR DONE + SPD + PRI + PUN
        LP=.
        MOV     #216,GOOD
        MOV     @TCR,BAD        ;READ CMD REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB                  ;CMD REG - VALUE READ NOT EQUAL TO EXPEC
        LOOP    LP
        JMP     @#LP

        RESET
        RTS     PC              ;EXIT CARD TEST
        .
;SUBROUTINE TO WAIT FOR CR
;
CONTIN: TYPE
        CONMSG                  ;"PRESS CR TO CONTINUE"
                JSR     PC,CKSWR
CONT1:  JSR     PC,TTYIN        ;GET A CHAR
        CMP     #15,R4          ;WAS IT CR ?
        BNE     CONT1           ;NO - KEEP WAITING
        JSR     PC,TTYOUT       ;YES - TYPE CR + LF
        RTS     PC
;
;SUBROUTINE TO TEST FOR BLANK CARD (ALL SPACES)
;
BLKCRD: MOV     #1,R0
                JSR     PC,CKSWR
BC1:    BIT     #200,@TIR
        BEQ     BC1             ;WAIT FOR INPUT DATA REQUEST
        MOV     #0,GOOD
        MOV     @TDR,BAD        ;READ DATA REGISTER
        CMP     GOOD,BAD
        BEQ     .+4
        ERR+GB+COL              ;DATA REG - COMPARE ERROR AT INDICATED C
        INC     R0
        CMP     #121,R0
        BNE     BC1             ;IF MORE COLUMNS TO READ GO DO IT
        CKTIR   #0,LP           ;TEST THAT IDR IS 0
        LP=.
        MOV     #0,GOOD
        MOV     @TIR,BAD        ;READ INTERRUPT REGISTER
```

```
(1)  012334  026767  006506  006474          CMP     GOOD,BAD
(1)  012342  001401                          BEQ     .+4
(1)  012344  104003                          ERR+GB                  ;INTERRUPT REG - VALUE READ NOT EQUAL TO
(1)  012346                                   LOOP    LP
(2)  012366  000137  012320                   JMP     @#LP
 922 012372  000207                          RTS     PC              ;EXIT
 923                                          ;
 924                                          ;PUT SPACE CHAR IN ALL LOCS BF20 - BF1+10
 925                                          ;
 926 012374  012702  020036          CLRBFS: MOV     #BF20,R2
 927 012400  012722  000000          CB1:    MOV     #0,(R2)+
 928 012404  022702  020130                  CMP     #BF1+24,R2
 929 012410  001373                          BNE     CB1
 930 012412  000207                          RTS     PC
 931                                          ;
 932                                          ;PUT 1 CHAR (4400) IN LOCS BF20 - BF1
 933                                          ;
 934 012414  012702  020036          SETBFS: MOV     #BF20,R2
 935 012420  012722  004400          SB1:    MOV     #4400,(R2)+
 936 012424  022702  020106                  CMP     #BF1+2,R2
 937 012430  001373                          BNE     SB1
 938 012432  000207                          RTS     PC
 939                                          ;
 940                                          ;SEND 80 CHARS BEGINNING AT LOC POINTED TO BY PTR
 941                                          ;
 942 012434  016700  006426          P2SEND: MOV     PTR,R0
 943 012440  004767  001512                  JSR     PC,CKSWR
 944 012444  012701  000120                  MOV     #120,R1
 945 012450  132777  000200  006440  P2SD1:  BITB    #200,@TIRHI
 946 012456  001774                          BEQ     P2SD1           ;WAIT FOR OUTPUT DATA REQUEST
 947 012460  011077  006424                  MOV     (R0),@TDR       ;SEND A CHAR
 948 012464  062700  000002                  ADD     #2,R0
 949 012470  005301                          DEC     R1              ;ALL 80 CHARS SENT ?
 950 012472  001366                          BNE     P2SD1           ;NO - GO SEND ANOTHER ONE
 951 012474  132777  000200  006414          BITB    #200,@TIRHI
 952 012502  001401                          BEQ     .+4
 953 012504  000000                          XX                      ;ERROR - GOT MORE THAN 80 ODR'S
 954 012506  000207                          RTS     PC
 955                                          ;
 956                                          ;SEND 1 CHAR TO CARD #1 UP TO 20 CHARS TO CARD #20
 957                                          ;
 958 012510  016767  006324  006360  P3SEND: MOV     CARD,TOG1
 959 012516  012700  000001                  MOV     #1,R0
 960 012522  004767  001430                  JSR     PC,CKSWR
 961 012526  132777  000200  006362  P3SD1:  BITB    #200,@TIRHI
 962 012534  001774                          BEQ     P3SD1           ;WAIT FOR OUTPUT DATA REQUEST
 963 012536  012777  004400  006344          MOV     #4400,@TDR      ;SEND "4400" CHARACTER
 964 012544  005200                          INC     R0
 965 012546  005367  006324                  DEC     TOG1
 966 012552  001365                          BNE     P3SD1           ;NO - GO SEND ANOTHER ONE
 967 012554  132777  000200  006334  P3SD2:  BITB    #200,@TIRHI
 968 012562  001774                          BEQ     P3SD2           ;WAIT FOR OUTPUT DATA REQUEST
 969 012564  112777  000000  006320          MOVB    #0,@TDRHI       ;SEND SPACE CHAR
 970 012572  005200                          INC     R0
 971 012574  022700  000121                  CMP     #121,R0         ;HAVE ALL 80 CHARS BEEN SENT
 972 012600  001365                          BNE     P3SD2           ;NO - GO SEND MORE
```

```
 973  012602  000207                                    RTS     PC
 974                                                    ;
 975                                                    ;
 976                                                    ;SEND CHAR #7777 TO CARD #1 UPTO 20 CHARS TO CARD #20 TO
 977                                                    ;PUNCH ALL HOLES IN RESPECTIVE COLUMN NUMBER
 978  012604  016767  006230  006264              P5SEND: MOV     CARD,TOG1
 979  012612  012700  000001                            MOV     #1,R0
 980  012616  132777  000200  006272              P5SD1:  BITB    #200,@TIRHI
 981  012624  001774                                     BEQ     P5SD1           ;WAIT FOR OUTPUT DATA REQUEST
 982  012626  012777  007777  006254                     MOV     #7777,@TDR      ;SEND "7777" CHARACTER
 983  012634  005200                                     INC     R0
 984  012636  005367  006234                            DEC     TOG1            ;HAVE ALL CHARS BEEN SENT
 985  012642  001365                                     BNE     P5SD1           ;NO - GO SEND ANOTHER ONE
 986  012644  132777  000200  006244              P5SD2:  BITB    #200,@TIRHI
 987  012652  001774                                     BEQ     P5SD2           ;WAIT FOR OUTPUT DATA REQUEST
 988  012654  112777  000000  006230                     MOVB    #0,@TDRHI
 989  012662  005200                                     INC     R0
 990  012664  022700  000121                            CMP     #121,R0         ;HAVE ALL 80 CHARS BEEN SENT
 991  012670  001365                                     BNE     P5SD2           ;NO - GO SEND MORE CHARS
 992  012672  000207                                     RTS     PC
 993
 994
 995
 996
 997                                                    ;
 998                                                    ;SEND 7 CHARS TO BE PRINTED "CARD XX" AND THEN ALL SPACES
 999  012674  005067  006176                            ;
1000  012700  004767  001252              P4SEND: CLR     TOG1
1001  012704  132777  000200  006204                    JSR     PC,CKSWR
1002  012712  001774                              P4SD1:  BITB    #200,@TIRHI
1003  012714  017777  006142  006166                     BEQ     P4SD1           ;WAIT FOR OUTPUT DATA REQUEST
                                                         MOV     @OUTPTR,@TDR    ;SEND A CHAR
```

```
1005  012722  062767  000002  006132          ADD     #2,OUTPTR
1006  012730  005267  006142                  INC     TOG1
1007  012734  022767  000007  006134          CMP     #7,TOG1         ;HAVE ALL PRINTING CHARS BEEN SENT
1008  012742  001360                          BNE     P4SD1           ;NO - GO PRINT MORE
1009  012744  132777  000200  006144  P4SD2:  BITB    #200,@TIRHI
1010  012752  001774                          BEQ     P4SD2           ;WAIT FOR OUTPUT DATA REQUEST
1011  012754  012777  000000  006126          MOV     #0,@TDR         ;SEND SPACE CHAR
1012  012762  005267  006110                  INC     TOG1
1013  012766  022767  000120  006102          CMP     #120,TOG1       ;HAVE ALL 80 CHARS BEEN SENT
1014  012774  001363                          BNE     P4SD2           ;NO - GO SEND MORE CHARS
1015  012776  000207                          RTS     PC
1016                                          ;
1017                                  ;COMPARE INPUT DATA PATTERN BEGINNING AT INPTR
1018                                  ;
1019          000002                  MP=2
1020
1021  013000  012700  000001  CKDATA: MOV     #1,R0
1022  013004  016767  006042  006054          MOV     INPTR,PTR
1023  013012  032777  000200  006074  CD1:    BIT     #200,@TIR
1024  013020  001774                          BEQ     CD1             ;WAIT FOR INPUT DATA REQUEST
1025  013022  032777  000002  006054          BIT     #MP,@TSR        ;TEST FOR MULTI-PUNCH
1026  013030  001402                          BEQ     1$
1027  013032  005267  006036                  INC     MPSAV           ;YES MP SET
1028  013036  017767  006024  006002  1$:     MOV     @PTR,GOOD
1029  013044  017767  006040  005764          MOV     @TDR,BAD        ;READ DATA REG
1030  013052  026767  005770  005756          CMP     GOOD,BAD
1031  013060  001401                          BEQ     2$
1032  013062  104007                          ERR+GB+COL              ;DATA REG - COMPARE ERROR AT INDICATED C
1033  013064  005767  006004  2$:     TST     MPSAV   ;LOOK FOR MULT-P BIT
1034  013070  001406                          BEQ     3$              ;NO
1035  013072  022767  000377  005736          CMP     #377,BAD        ;IS DATA MULTI-DATA
1036  013100  001407                          BEQ     5$              ;YES
1037  013102  104007                          ERR+GB+COL              ;MULTI-P SET,DATA NOT 377
1038  013104  000405                          BR      5$              ;EXIT
1039  013106  022767  000377  005722  3$:     CMP     #377,BAD        ;TEST FOR MULTI-DATA
1040  013114  001001                          BNE     5$              ;NO MULTI-DATA
1041  013116  104007                          ERR+GB+COL              ;MULTI-P DATA,NO MP SEEN
1042
1043  013120  005067  005750  5$:     CLR     MPSAV
1044  013124  005200                          INC     R0
1045  013126  062767  000002  005732          ADD     #2,PTR
1046  013134  022700  000121                  CMP     #121,R0
1047  013140  001324                          BNE     CD1             ;IF MORE COLUMNS TO READ GO DO IT
1048  013142  000207                          RTS     PC              ;EXIT
1049                                  ;
1050                                  ;CONVERT OCTAL # IN R0 TO DECIMAL # AND TYPE OUT DECIMAL #
1051                                  ;
1052  013144  010067  000110  OCTDEC: MOV     R0,TOG
1053  013150  005067  000100          CLR     ONES
1054  013154  005067  000076          CLR     TENS
1055  013160  005267  000070  OD1:    INC     ONES
1056  013164  022767  000012  000062          CMP     #12,ONES
1057  013172  001404                          BEQ     OD2
1058  013174  005367  000060                  DEC     TOG
1059  013200  001367                          BNE     OD1
1060  013202  000407                          BR      OD3
```

```
1061  013204  005067  000044          OD2:    CLR     ONES
1062  013210  005267  000042                  INC     TENS
1063  013214  005367  000040                  DEC     TOG
1064  013220  001357                          BNE     OD1
1065  013222  016704  000030          OD3:    MOV     TENS,R4
1066  013226  062704  000060                  ADD     #60,R4
1067  013232  004767  001556                  JSR     PC,TTYOUT       ;TYPEOUT TENS
1068  013236  016704  000012                  MOV     ONES,R4
1069  013242  062704  000060                  ADD     #60,R4
1070  013246  004767  001542                  JSR     PC,TTYOUT       ;TYPEOUT ONES
1071  013252  000207                          RTS     PC
1072  013254  000000          ONES:   XX
1073  013256  000000          TENS:   XX
1074  013260  000000          TOG:    XX
1075                                  ;INPUT ROUTINE FOR DEVICE ADDRESS, INT VECTOR ADDRESS AND BR LEV
1076  013262  104400          DEVCOD: TYPE
1077  013264  015116                  DCMSG1                  ;''THE FOLLOWING VALUES ARE BEING USED''
1078  013266  104400                  TYPE
1079  013270  015164                  DCMSG2                  ;''    DEVICE ADDRESS = ''
1080  013272  104406                  TYPE+6
1081  013274  021100                  TCR                     ;TYPE TCR ADDRESS
1082  013276  104400                  TYPE
1083  013300  015213                  DCMSG3                  ;''    PC INTERRUPT VECTOR ADDRESS = ''
1084  013302  104403                  TYPE+3
1085  013304  021120                  PCV                     ;TYPE PCV ADDRESS
1086  013306  104400                  TYPE
1087  013310  015257                  DCMSG4                  ;''    BUS REQUEST LEVEL = ''
1088  013312  104401                  TYPE+1
1089  013314  021124                  BRLV                    ;TYPE BRLV
1090                                  ;TEST TO SEE IF VALUES ARE TO BE CHANGED
1091  013316  104400                  TYPE
1092  013320  015311                  DCMSG5                  ;''DO YOU WANT TO CHANGE ANY OF THESE VAL
1093  013322  004767  001542  DC1:    JSR     PC,TTYIN        ;INPUT A CHAR
1094  013326  022704  000131          CMP     #131,R4         ;IS IT Y ?
1095  013332  001004                  BNE     DC2             ;NO - GO CHECK FOR N
1096  013334  004767  001454          JSR     PC,TTYOUT       ;YES - ECHO Y
1097  013340  000167  000014          JMP     DC3             ;GO GET NEW VALUES
1098  013344  022704  000116  DC2:    CMP     #116,R4         ;IS IT N ?
1099  013350  001364                  BNE     DC1             ;NO - GO BACK TO INPUT
1100  013352  004767  001436          JSR     PC,TTYOUT       ;YES - ECHO N
1101  013356  000207                  RTS     PC              ;EXIT
1102                                  ;VALUES ARE TO BE CHANGED
1103  013360  104400          DC3:    TYPE
1104  013362  015377                  DCMSG6                  ;''  ENTER VALUE AFTER (=) IS TYPED''
1105  013364  104400                  TYPE
1106  013366  015164                  DCMSG2                  ;''    DEVICE ADDRESS = ''
1107  013370  012705  000006          MOV     #6,R5
1108  013374  004767  000470          JSR     PC,RDOCT        ;GET NEW DEVICE ADDRESS
1109  013400  042704  000007          BIC     #7,R4
1110  013404  012705  021100          MOV     #TCR,R5
1111  013410  042715  177770  DC4:    BIC     #177770,(R5)
1112  013414  050425                  BIS     R4,(R5)+        ;STORE 4 NEW DEVICE ADDRESSES
1113  013416  020527  021120          CMP     R5,#PCV
1114  013422  001372                  BNE     DC4
1115  013424  104400                  TYPE
1116  013426  015213                  DCMSG3                  ;''    PC INTERRUPT VECTOR ADDRESS = ''
```

```
1117  013430  012705  000003                    MOV     #3,R5
1118  013434  004767  000430                    JSR     PC,RDOCT        ;GET NEW INT VECTOR ADDRESS
1119  013440  010467  005454                    MOV     R4,PCV          ;SET NEW PC VECTOR ADDRESS
1120  013444  062704  000002                    ADD     #2,R4
1121  013450  010467  005446                    MOV     R4,PSV          ;SET NEW PS VECTOR ADDRESS
1122  013454  104400                            TYPE
1123  013456  015257                            DCMSG4                  ;''    BUS REQUEST LEVEL = ''
1124  013460  012705  000001                    MOV     #1,R5
1125  013464  004767  000400                    JSR     PC,RDOCT        ;GET NEW BR LEVEL
1126  013470  010467  005430                    MOV     R4,BRLV         ;SET NEW BR LEVEL
1127  013474  000207                            RTS     PC              ;EXIT
1128                                            ;
1129                                    ;ERROR SUBROUTINE
1130                                    ;
1131  013476  004767  000454            ERROR:  JSR     PC,CKSWR
1132  013502  032777  020000  165170            BIT     #20000,@SWR
1133  013510  001403                            BEQ     .+10            ;IS LOOP ON ERROR SELECTED ?
1134  013512  012767  000001  005350            MOV     #1,SCOPE        ;YES - SET SCOPE TO A 1
1135  013520  011667  005330                    MOV     (SP),LASTPC     ;PUT ADDRESS OF ERROR IN LASTPC
1136  013524  032777  040000  165146            BIT     #40000,@SWR     ;IF SR14=1 DELETE TYPEOUT
1137  013532  001041                            BNE     E3              ;EXIT NOW
1138  013534  104400                            TYPE
1139  013536  015576                            PCMSG                   ;''ERROR AT ADDRESS''
1140  013540  011667  005270                    MOV     (SP),ADR
1141  013544  162767  000002  005262            SUB     #2,ADR
1142  013552  104406                            TYPE+6
1143  013554  021034                            ADR                     ;TYPE CONTENTS OF ADR
1144  013556  032777  000002  005250            BIT     #2,@ADR         ;TYPE GOOD ?
1145  013564  001404                            BEQ     E1              ;NO
1146  013566  104400                            TYPE
1147  013570  015622                            GDMSG                   ;''GOOD=''
1148  013572  104406                            TYPE+6
1149  013574  021046                            GOOD                    ;TYPE CONTENTS OF GOOD
1150  013576  032777  000001  005230    E1:     BIT     #1,@ADR         ;TYPE BAD ?
1151  013604  001404                            BEQ     E2              ;NO
1152  013606  104400                            TYPE
1153  013610  015631                            BDMSG                   ;''BAD =''
1154  013612  104406                            TYPE+6
1155  013614  021036                            BAD                     ;TYPE CONTENTS OF BAD
1156  013616  032777  000004  005210    E2:     BIT     #4,@ADR
1157  013624  001404                            BEQ     E3
1158  013626  104400                            TYPE
1159  013630  015640                            COLMSG                  ;''COLUMN #=''
1160  013632  004767  177306                    JSR     PC,OCTDEC       ;TYPEOUT DECIMAL COLUMN # IN ERROR
1161  013636  012704  000015            E3:     MOV     #15,R4
1162  013642  004767  001146                    JSR     PC,TTYOUT       ;TYPE CR + LF
1163  013646  032777  100000  165024            BIT     #100000,@SWR    ;IF SR15=1
1164  013654  001001                            BNE     E4              ;BYPASS THE ERROR HALT
1165  013656  000000            ERRHLT: XX                              ;ERROR HALT
1166  013660  000002            E4:     RTI                             ;EXIT
1167                                            ;
1168                                    ;DETERMINE IF THIS TYPEOUT IS FOR MESSAGE OR OCTAL VALUE AND BRANCH ACCO
1169                                    ;
1170  013662  011603            TYPOUT: MOV     (SP),R3                 ;GET PC FROM STACK
1171  013664  162703  000002                    SUB     #2,R3           ;SUBTRACT 2
1172  013670  032713  000007                    BIT     #7,(R3)         ;TEST LOW ORDER 3 BITS OF TRAP INSTRUCTI
```

```
1173  013674  001401                          BEQ     TYPMSG          ;IF 0'S GO TO TYPMSG
1174  013676  000421                          BR      TYPOCT          ;IF NOT 0'S GO TO TYPOCT
1175                                  ;
1176                                  ;MESSAGE TYPEOUT - R5=ADDRESS OF BEGINNING OF MESSAGE
1177  013700  004767  000252          TYPMSG: JSR     PC,CKSWR
1178  013704  011604                          MOV     (SP),R4
1179  013706  011405                          MOV     (R4),R5         ;R5=ADDRESS OF MESSAGE
1180  013710  112504                  TM1:    MOVB    (R5)+,R4        ;PUT CHAR IN R4
1181  013712  001410                          BEQ     TM3             ;IF IT EQUALS 0 TERMINATE
1182  013714  022704  000046                  CMP     #46,R4          ;DOES CHAR = CRLF FLAG ?
1183  013720  001002                          BNE     TM2             ;NO - GO TYPE CHAR
1184  013722  012704  000015                  MOV     #15,R4          ;YES - TYPE CR+LF
1185  013726  004767  001062          TM2:    JSR     PC,TTYOUT       ;GO TYPE CONTENTS OF R4
1186  013732  000766                          BR      TM1             ;GO GET NEXT CHAR
1187  013734  062716  000002          TM3:    ADD     #2,(SP)         ;ADD 2 TO SP FOR RETURN
1188  013740  000002                          RTI                     ;EXIT
1189                                  ;
1190                                  ;OCTAL TYPEOUT - R4=VALUE  R5=# OF DIGITS TO TYPE
1191                                  ;
1192  013742  011305                  TYPOCT: MOV     (R3),R5
1193  013744  042705  177770                  BIC     #177770,R5      ;R5 = # OF DIGITS TO BE TYPED
1194  013750  062703  000002                  ADD     #2,R3
1195  013754  011367  005104                  MOV     (R3),PT
1196  013760  017704  005100                  MOV     @PT,R4          ;R4 = VALUE TO BE TYPED
1197  013764  010567  005052                  MOV     R5,CNT          ;SET CNT FOR STACKING
1198  013770  010467  005076          TO.1:   MOV     R4,TEMP         ;PUT VALUE IN TEMP
1199  013774  042767  177770  005070        BIC     #177770,TEMP    ;MASK OUT ALL BUT BITS 2,1+0
1200  014002  062767  000060  005062        ADD     #60,TEMP        ;MAKE ASCII NUMBER
1201  014010  016746  005056                  MOV     TEMP,-(SP)      ;STORE NUMBER ON STACK
1202  014014  006004                          ROR     R4
1203  014016  006004                          ROR     R4
1204  014020  006004                          ROR     R4
1205  014022  005367  005014                  DEC     CNT             ;IF CNT IS NOT ZERO
1206  014026  001360                          BNE     TO.1            ;GO STACK ANOTHER NUMBER
1207  014030  022705  000006                  CMP     #6,R5
1208  014034  001002                          BNE     TO.2            ;IF 6 NUMBERS WERE SELECTED
1209  014036  042716  000006                  BIC     #6,(SP)         ;CLEAR BITS 1+0 OF LAST NUMBER STACKED
1210  014042  010567  004774          TO.2:   MOV     R5,CNT          ;SET CNT FOR TYPING
1211  014046  012604                  TO.3:   MOV     (SP)+,R4        ;GET A NUMBER FROM THE STACK
1212  014050  004767  000740                  JSR     PC,TTYOUT       ;TYPE THE NUMBER
1213  014054  005367  004762                  DEC     CNT             ;IF COUNT IS NOT ZERO
1214  014060  001372                          BNE     TO.3            ;GO TYPE ANOTHER NUMBER
1215  014062  062716  000002                  ADD     #2,(SP)         ;ADD 2 TO SP FOR RETURN
1216  014066  000002                          RTI                     ;EXIT
1217  014070                                  S7$:;
1218                                  ;OCTAL TYPEIN - R5=#OF DIGITS TO BE INPUT INTO R4
1219                                  ;
1220  014070  010567  004746          RDOCT:  MOV     R5,CNT          ;CNT=# OF DIGITS TO BE INPUT
1221  014074  005067  004772                  CLR     TEMP
1222  014100  004767  000764          RD.1:   JSR     PC,TTYIN        ;GET CHAR FROM TTY
1223  014104  004767  000704                  JSR     PC,TTYOUT       ;ECHO IT
1224  014110  042704  177770                  BIC     #177770,R4      ;MAKE CHAR OCTAL
1225  014114  006167  004752                  ROL     TEMP
1226  014120  006167  004746                  ROL     TEMP
1227  014124  006167  004742                  ROL     TEMP
1228  014130  042767  000007  004734          BIC     #7,TEMP         ;CLEAR TEMP BITS 2,1+0
```

```
1229   014136  060467  004730                          ADD     R4,TEMP         ;ADD INPUT CHAR TO OTHERS
1230   014142  005367  004674                          DEC     CNT             ;IF ALL CHARS NOT READ IN
1231   014146  001354                                  BNE     RD.1            ;GO READ ANOTHER ONE
1232   014150  016704  004716                          MOV     TEMP,R4         ;PUT FINAL OCTAL VALUE IN R4
1233   014154  000207                                  RTS     PC              ;EXIT
1234                                                  ;
1235                                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
1236                                                  ;*ROUTINE IS ENTERED FROM TRAP HANDLERR,AND WILL
1237                                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
1238                                                  ;*WHEN OPERATING IN TTY FLAG MODE.
1239   014156  022767  000176  164514                  CKSWR:  CMP     #SWREG,SWR      ;IS THE SOFT-SWR SELECTED?
1240   014164  001075                                  BNE     K15             ;BRANCH IF NO
1241   014166  105767  163366                          TSTB    TKS             ;CHAR THERE?
1242   014172  100072                                  BPL     K15             ;IF NO,DON'T WAIT AROUND
1243   014174  116746  163362                          MOVB    TKB,-(SP)       ;SAVE THE CHAR
1244   014200  042716  177600                          BIC     #^C177,(SP)     ;STRIP OFF THE ASCII
1245   014204  022726  000007                          CMP     #7,(SP)+        ;IS IT A CONTROL G?
1246   014210  001063                                  BNE     K15             ;NO RETURN TO USER
1247   014212  126727  164474  000001                  CMPB    AUTOB,#1        ;ARE WE RUNNING IN AUTO-MODE?
1248   014220  001457                                  BEQ     K15             ;BRANCH IF YES
1249   014222  104400  014463                          TYPE    ,CNTLG          ;ECHO THE CONTROL-G (^G)
1250   014226  104400  014470                  GTSWR:  TYPE    ,MSWR           ;TYPE CURRENT CONTENTS
1251   014232  016746  163740                          MOV     SWREG,-(SP)     ;SAVE SWREG FOR TYPEOUT
1252   014236  004767  000300                          JSR     PC,TYPOC        ;GO TYPE-OCTAL ASCII(ALL DIGITS)
1253   014242  104400  014500                          TYPE    ,MNEW           ;PROMPT FOR NEW SWR
1254   014246  005046                          K19:    CLR     -(SP)           ;CLEAR COUNTER
1255   014250  005046                                  CLR     -(SP)           ;NEW SWR
1256   014252  105767  163302                  K7:     TSTB    TKS             ;CHAR THERE?
1257   014256  100375                                  BPL     K7              ;IF NOT TRY AGAIN
1258   014260  116746  163276                          MOVB    TKB,-(SP)       ;PICK UP CHAR
1259   014264  042716  177600                          BIC     #^C177,(SP)     ;MAKE IT 7-BIT ASCII
1260   014270  021627  000025                  K9:     CMP     (SP),#25        ;IT IS A CONTROL-U?
1261   014274  001005                                  BNE     K10             ;BRANCH IF NOT
1262   014276  104400  014456                          TYPE    ,CNTLU          ;YES ECHO CONTROL - U(^U)
1263   014302  062706  000006                  K20:    ADD     #6,SP           ;IGNORE PREVIOUS INPUT
1264   014306  000757                                  BR      K19             ;LET'S TRY IT AGAIN
1265   014310  021627  000015                  K10:    CMP     (SP),#15        ;IS IT A <CR>?
1266   014314  001022                                  BNE     K16             ;BRANCH IF NO
1267   014316  005766  000004                          TST     4(SP)           ;YES,IS IT THE FIRST CHAR?
1268   014322  001403                                  BEQ     K11             ;BRANCH IF YES
1269   014324  016677  000002  164346                  MOV     2(SP),@SWR      ;SAVE NEW SWR
1270   014332  062706  000010                  K11:    ADD     #10,SP          ;CLEAR UP STACK
1271                                                  ;TTY INPUT ROUTINE
1272                                                  ;
1273   014336  104400  014512                  K14:    TYPE    ,CRLF           ;ECHO <CR> AND <LF>
1274   014342  126727  164345  000001                  CMPB    INTAG,#1        ;RE-ENABLE TTY KBD INTERRUPTS
1275   014350  001003                                  BNE     K15             ;BRANCH IF NOT
1276   014352  012767  000100  163200                  MOV     #100,TKS        ;RE-ENABLE TTY KBD INTERRUPTS
1277   014360  000207                          K15:    RTS     PC              ;RETURN
1278   014362  004767  000356                  K16:    JSR     PC,TYPEC        ;ECHO CHAR
1279   014366  021627  000060                          CMP     (SP),#60        ;CHAR < 0?
1280   014372  002420                                  BLT     K18             ;BRANCH IF YES
1281   014374  021627  000067                          CMP     (SP),#67        ;CHAR > 7?
1282   014400  003015                                  BGT     K18             ;BRANCH IF YES
1283   014402  042726  000060                          BIC     #60,(SP)+       ;STRIP-OFF ASCII
1284   014406  005766  000002                          TST     2(SP)           ;IS THIS THE FIRST CHAR
```

```
1285  014412  001403                          BEQ     K17             ;BRANCH IF YES
1286  014414  006316                          ASL     (SP)            ;NO SHIFT PRESENT
1287  014416  006316                          ASL     (SP)            ;CHAR OVER TO MAKE
1288  014420  006316                          ASL     (SP)            ;ROOM FOR NEW ONE
1289  014422  005266  000002          K17:    INC     2(SP)           ;KEEP COUNT OF CHARACTER
1290  014426  056616  177776                  BIS     -2(SP),(SP)     ;SET IN NEW CHAR
1291  014432  000707                          BR      K7              ;GET THE NEXT ONE
1292  014434  104400  014511          K18:    TYPE    ,QUES           ;TYPE ?<CR><LF>
1293  014440  000720                          BR      K20             ;SIMULATE CONTROL-U
1294  014442  000014                  TTYN:   .BLKB   12.              ;RESERVE 12 BYTE FOR TTYT
1295                                                                   ;Y INPUT
1296  014456  052536  005015     000  CNTLU:  .ASCIZ  /^U/<15><12>    ;CONTROL "U"
1297  014463     136  006507  000012  CNTLG:  .ASCIZ  /^G/<15><12>    ;CONTROL "G"
1298  014470  005015  053523  036522  MSWR:   .ASCIZ  <15><12>/SWR= /
      014476  000040
1299
1300  014500  020040  042516  020127  MNEW:   .ASCIZ  /  NEW = /
      014506  020075     000
1301
1302  014511     077                  QUES:   .ASCII  /?/             ;QUESTION MARK
1303  014512     015                  CRLF:   .ASCII  <15>            ;CARRIAGE RETURN
1304  014513     012     000          LF:     .ASCIZ  <12>            ;LINE FEED
1305                                  ;*******************************************************
1306
1307                                  ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO
1308                                  ;*A 6-DIGIT OCTAL (ASCII) NUMBER AND TYPE IT
1309                                  ;*TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF
1310                                  ;*DIGITS TO TYPE
1311                                  ;*CALL:
1312                                  ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
1313                                  ;*      TYPOS                   ;CALL FOR TYPE OUT
1314                                  ;*      .BYTE   N               ;N=1 TO 6 FOR NUMBER OF DIGITS
1315                                   ;*TO TYPE
1316                                   ;*       .BYTE   M                ;M=1  OR 0
1317                                                                     ;1=TYPE LEADING ZEROS
1318                                                                     ;0=SUPPRESS LEADING ZEROS
1319                                   ;*
1320                                   ;*TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS
1321                                   ;*THE LAST
1322                                   ;*TYPOS OR TYPOC
1323                                   ;*CALL:
1324                                   ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
1325                                   ;*      TYPON                   ;CALL FOR TYPEOUT
1326                                   ;*
1327                                   ;*TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1328                                   ;*CALL:
1329                                   ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
1330                                   ;*      TYPOC                   ;CALL FOR TYPEOUT
1331          014516                  .EVEN
1332  014516  017646  000000          TYPOS:  MOV     @(SP),-(SP)     ;PICKUP THE MODE
1333  014522  116667  000001  000211          MOVB    1(SP),FILL0     ;LOAD ZERO FILL SWITCH
1334  014530  112667  000207                  MOVB    (SP)+,MODE0+1   ;NUMBER OF DIGIT TO TYPE
1335  014534  062716  000002                  ADD     #2,(SP)          ;ADJUST RETURN ADDRESS
1336  014540  000406                          BR      TYPON
1337  014542  112767  000001  000171  TYPOC:  MOVB    #1,FILL0         ;SET THE ZERO FILL SWITCH
1338  014550  112767  000006  000165          MOVB    #6,MODE0+1       ;SET FOR SIX(6) DIGITS
```

```
1339  014556  112767  000005  000154          TYPON:  MOVB    #5,CNTO         ;SET THE ITERATION COUNT
1340  014564  010346                                  MOV     R3,-(SP)        ;SAVR R3
1341  014566  010446                                  MOV     R4,-(SP)        ;SAVE R4
1342  014570  010546                                  MOV     R5,-(SP)        ;SAVE R5
1343  014572  116704  000145                         MOVB    MODE0+1,R4      ;GET THE NUMBER OF DIGIT TO TYPE
1344  014576  005404                                  NEG     R4
1345  014600  062704  000006                         ADD     #6,R4           ;SUBSTRACT IT FOR MAX.ALLOWED
1346  014604  110467  000132                         MOVB    R4,MODE0        ;SAVE IT FOR USE
1347  014610  116704  000125                         MOVB    FILL0,R4        ;GET THE ZERO FILL SWITCH
1348  014614  016605  000010                         MOV     10(SP),R5       ;PICK UP THE INPUT NUMBER
1349  014620  005003                                  CLR     R3              ;CLEAR THE OUTPUT WORD
1350  014622  006105                          A1:     ROL     R5              ;ROTATE MSB INTO ''C''
1351  014624  000404                                  BR      A3              ;GO TO MSB
1352  014626  006105                          A2:     ROL     R5              ;FORM THIS DIGIT
1353  014630  006105                                  ROL     R5
1354  014632  006105                                  ROL     R5
1355  014634  010503                                  MOV     R5,R3
1356  014636  006103                          A3:     ROL     R3              ;GET LSB OF THIS DIGIT
1357  014640  105367  000076                         DECB    MODE0           ;TYPE THIS DIGIT?
1358  014644  100022                                  BPL     A7              ;BR IF NO
1359  014646  042703  177770                         BIC     #177770,R3      ;GET RID OF JUNK
1360  014652  001002                                  BNE     A4              ;TEST FOR 0
1361  014654  005704                                  TST     R4                      ;SUPPRESS THIS 0?
1362  014656  001403                                  BEQ     A5              ;BR IF YES
1363  014660  005204                          A4:     INC     R4              ;DON'T SUPPRESS ANYMORE 0'S
1364  014662  052703  000060                         BIS     #'0,R3          ;MAKE THIS DIGIT ASCII
1365  014666  052703  000040                  A5:    BIS     #' ,R3          ;MAKE ASCII IF NOT ALREADY
1366  014672  110367  000040                         MOVB    R3,A8           ;SAVE FOR TYPING
1367  014676  010446                                  MOV     R4,-(SP)
1368  014700  010546                                  MOV     R5,-(SP)
1369  014702  104400  014736                          TYPE    ,A8             ;GO TYPE THIS DIGIT
1370  014706  012605                                  MOV     (SP)+,R5
1371  014710  012604                                  MOV     (SP)+,R4
1372  014712  105367  000022                  A7:    DECB    CNTO            ;COUNT BY 1
1373  014716  003343                                  BGT     A2              ;BR IF MORE TO DO
1374  014720  002402                                  BLT     A6              ;BR IF DONE
1375  014722  005204                                  INC     R4              ;INSURE LAST DIGIT IS NOT A BLANK
1376  014724  000740                                  BR      A2              ;GO DO THE LAST DIGIT
1377  014726  012605                          A6:     MOV     (SP)+,R5        ;RESTORE R5
1378  014730  012604                                  MOV     (SP)+,R4        ;RESTORE R4
1379  014732  012603                                  MOV     (SP)+,R3        ;RESTORE R3
1380  014734  000207                                  RTS     PC
1381  014736     000                   A8:     .BYTE   0               ;STORAGE FOR ASCII DIGIT
1382  014737     000                           .BYTE   0               ;TERMINATOR FOR TYPE ROUTINE
1383  014740     000                   CNTO:    .BYTE   0               ;OCTAL DIGIT COUNTER
1384  014741     000                   FILL0:  .BYTE   0               ;ZERO FILL SWITCH
1385  014742  000000                   MODE0:  .WORD   0               ;NUMBER OF DIGIT TO TYPE
1386                                          ;
1387                                          ;
1388                                          ;
1389  014744  105767  162614          TYPEC:  TSTB    TPS             ;WAIT UNTIL THE PRINTER IS READY
1390  014750  100375                          BPL     TYPEC
1391  014752  116667  000002  162606          MOVB    2(SP),TPB       ;LOAD CHARACTER TO BE TYPED INTO DATA REG
1392  014760  122766  000015  000002          CMPB    #15,2(SP)       ;IS CHAR CARRIAGE RETURN?
1393  014766  001003                          BNE     B1
1394  014770  105067  000014                  CLRB    CHARCNT         ;YES CLEAR CHAR COUNT
```

```
1395  014774  000406                        BR      TYPEX           ;EXIT
1396  014776  122766  014513  000002  B1:   CMPB    #LF,2(SP)       ;IS CHAR LINE FEED?
1397  015004  001402                        BEQ     TYPEX           ;BRANCH IF YES
1398  015006  105227                        INCB    (PC)+           ;COUNT THE CHARS
1399  015010  000000              CHARCNT:  .WORD   0               ;CHARACTER COUNT SPACE
1400  015012  000207              TYPEX:    RTS     PC
1401
1402
1403                                        ;
1404                                        ;TYPE THE CHARACTER IN R4
1405                                        ;
1406  015014  032777  040000  163656  TTYOUT: BIT     #40000,@SWR     ;IF SR14=1 DELETE TYPEOUT
1407  015022  001021                        BNE     TTO.3
1408  015024  010467  162536                MOV     R4,TPB          ;TYPE CHAR IN R4
1409  015030  105767  162530        TTO.1:  TSTB    TPS
1410  015034  100375                        BPL     TTO.1           ;WAIT FOR DONE
1411  015036  022704  000015                CMP     #15,R4          ;WAS IT A CR ?
1412  015042  001003                        BNE     TTO.2           ;IF NO - GO EXIT
1413  015044  012704  000012                MOV     #12,R4          ;IF YES - PUT LF IN R4
1414  015050  000761                        BR      TTYOUT          ;GO TYPE THE LF
1415  015052  012767  040000  004000  TTO.2:  MOV     #40000,LOC
1416  015060  005367  003774                DEC     LOC
1417  015064  001375                        BNE     .-4
1418  015066  000207              TTO.3:    RTS     PC              ;EXIT
1419                                        ;
1420                                        ;READ CHAR FROM TTY INTO R4
1421                                        ;
1422  015070  005267  162464      TTYIN:    INC     TKS             ;FETCH A CHAR
1423  015074  032767  000200  162456  TTI.1:  BIT     #200,TKS
1424  015102  001774                        BEQ     TTI.1           ;WAIT FOR DONE FLAG
1425  015104  016704  162452                MOV     TKB,R4          ;READ CHAR INTO R4
1426  015110  042704  177600                BIC     #177600,R4      ;MASK R4 WITH ALL BUT 177
1427  015114  000207                        RTS     PC              ;EXIT
1428  015116  023046  044124  020105        DCMSG1: .ASCIZ  /&&THE FOLLOWING VALUES ARE BEING USED/
      015124  047506  046114  053517
      015132  047111  020107  040526
      015140  052514  051505  040440
      015146  042522  041040  044505
      015154  043516  052440  042523
      015162  000104
1429  015164  020046  020040  042040  DCMSG2: .ASCIZ  /&    DEVICE ADDRESS = /
      015172  053105  041511  020105
      015200  042101  051104  051505
      015206  020123  020075     000
1430  015213     046  020040  020040  DCMSG3: .ASCIZ  /&    PC INTERRUPT VECTOR ADDRESS = /
      015220  041520  044440  052116
      015226  051105  052522  052120
      015234  053040  041505  047524
      015242  020122  042101  051104
      015250  051505  020123  020075
      015256     000
1431  015257     046  020040  020040  DCMSG4: .ASCIZ  /&    BUS REQUEST LEVEL = /
      015264  052502  020123  042522
      015272  052521  051505  020124
      015300  042514  042526  020114
      015306  020075     000
```

SEQ 0043

```
 1432  015311     046  042046  020117
       015316  047531  020125  040527
       015324  052116  052040  020117
       015332  044103  047101  042507
       015340  040440  054516  047440
       015346  020106  044124  051505
       015354  020105  040526  052514
       015362  051505  024040  020131
       015370  051117  047040  037451
       015376     000
 1433  015377     046  020040  047105
       015404  042524  020122  040526
       015412  052514  020105  043101
       015420  042524  020122  036450
       015426  020051  051511  052040
       015434  050131  042105     000
 1434  015441     046  025046  025052
       015446  025052  046040  043517
       015454  041511  052040  051505
       015462  051524  025040  025052
       015470  025052  023046     000
 1435  015475     046  050046  052125
       015502  034040  031460  026465
       015510  030070  032464  047440
       015516  043106  046055  047111
       015524  026105  051040  046505
       015532  053117  020105  046101
       015540  020114  040503  042122
       015546  000123
 1436  015550  050046  042522  051523
       015556  041440  020122  047524
       015564  041440  047117  044524
       015572  052516  000105
 1437  015576  023046  051105  047522
       015604  020122  052101  040440
       015612  042104  042522  051523
       015620  000040
 1438  015622  043446  047517  036504
       015630     000
 1439  015631     046  040502  020104
       015636  000075
 1440  015640  041446  046117  046525
       015646  020116  036443     000
 1441  015653     046  050046  052125
       015660  034040  031460  026465
       015666  030070  032464  047440
       015674  026516  044514  042516
       015702  040440  042116  050040
       015710  042522  051523  051440
       015716  040524  052122  047440
       015724  020116  044124  020105
       015732  030070  032463  034055
       015740  032060  000065
 1442  015744  023046  052520  020124
       015752  030070  032463  034055
       015760  032060  020065  043117
```

```
DCMSG5: .ASCIZ  /&&DO YOU WANT TO CHANGE ANY OF THESE VALUES (Y OR N)?/

DCMSG6: .ASCIZ  /&  ENTER VALUE AFTER (=) IS TYPED/

LOGTST: .ASCIZ  /&&***** LOGIC TESTS *****&&/

INITM1: .ASCIZ  /&&PUT 8035-8045 OFF-LINE, REMOVE ALL CARDS/

CONMSG: .ASCIZ  /&PRESS CR TO CONTINUE/

PCMSG:  .ASCIZ  /&&ERROR AT ADDRESS /

GDMSG:  .ASCIZ  /&GOOD=/

BDMSG:  .ASCIZ  /&BAD =/

COLMSG: .ASCIZ  /&COLUMN #=/

T2MSG1: .ASCIZ  /&&PUT 8035-8045 ON-LINE AND PRESS START ON THE 8035-804

T5M1:   .ASCIZ  /&&PUT 8035-8045 OFF-LINE AND PRESS START ON THE 8035_80
```

```
                015766  026506  044514  042516
                015774  040440  042116  050040
                016002  042522  051523  051440
                016010  040524  052122  047440
                016016  020116  044124  020105
                016024  030070  032463  034137
                016032  032060  000065
1443            016036  023046  051120  051505          T5M2:    .ASCIZ  /&&PRESS START ON THE 8035-8045/
                016044  020123  052123  051101
                016052  020124  047117  052040
                016060  042510  034040  031460
                016066  026465  030070  032464
                016074     000
1444            016075     046  050046  046125          T5MSG1: .ASCIZ  /&&PULL CARD PLATE OF STACKER #1 BACK TO POSITION WHICH/
                016102  020114  040503  042122
                016110  050040  040514  042524
                016116  047440  020106  052123
                016124  041501  042513  020122
                016132  030443  041040  041501
                016140  020113  047524  050040
                016146  051517  052111  047511
                016154  020116  044127  041511
                016162  000110
1445            016164  053446  046111  020114          T5MSG2: .ASCIZ  /&WILL SIMULATE STACKER FULL AND RELEASE/
                016172  044523  052515  040514
                016200  042524  051440  040524
                016206  045503  051105  043040
                016214  046125  020114  047101
                016222  020104  042522  042514
                016230  051501  000105
1446            016234  050046  047522  051107          T5MSG3: .ASCIZ  /&PROGRAM WILL WAIT FOR STACKER FULL FLAG/
                016242  046501  053440  046111
                016250  020114  040527  052111
                016256  043040  051117  051440
                016264  040524  045503  051105
                016272  043040  046125  020114
                016300  046106  043501     000
1447            016305     046  050046  042522          T5MSG4: .ASCIZ  /&&PRESS START ON THE 8035-8045/
                016312  051523  051440  040524
                016320  052122  047440  020116
                016326  044124  020105  030070
                016334  032463  034055  032060
                016342  000065
1448            016344  023046  052520  046114          T5MSG5: .ASCIZ  /&&PULL CARD PLATE OF STACKER #2 BACK TO POSITION WHICH/
                016352  041440  051101  020104
                016360  046120  052101  020105
                016366  043117  051440  040524
                016374  045503  051105  021440
                016402  020062  040502  045503
                016410  052040  020117  047520
                016416  044523  044524  047117
                016424  053440  044510  044103
                016432     000
1449            016433     046  050046  042522          T5MSG6: .ASCIZ  /&&PRESS START ON THE 8035-8045/
                016440  051523  051440  040524
                016446  052122  047440  020116
```

```
                016454  044124  020105  030070
                016462  032463  034055  032060
                016470  000065
1450            016472  023046  051120  051505          T5MSG7: .ASCIZ  /&&PRESS START ON THE 8035-8045/
                016500  020123  052123  051101
                016506  020124  047117  052040
                016514  042510  034040  031460
                016522  026465  030070  032464
                016530     000
1451            016531     046  051120  051505          T6MSG1: .ASCIZ  /&PRESS  START ON THE 8035-8045/
                016536  020123  051440  040524
                016544  052122  047440  020116
                016552  044124  020105  030070
                016560  032463  034055  032060
                016566  000065
1452            016570  023046  025052  025052          CRDTST: .ASCIZ  /&&***** CARD TESTS *****&&/
                016576  020052  040503  042122
                016604  052040  051505  051524
                016612  025040  025052  025052
                016620  023046     000
1453            016623     046  051046  046505          P1MSG1: .ASCIZ  /&&REMOVE ALL CARDS AND PUT THE 8035-8045 OFF-LINE/
                016630  053117  020105  046101
                016636  020114  040503  042122
                016644  020123  047101  020104
                016652  052520  020124  044124
                016660  020105  030070  032463
                016666  034055  032060  020065
                016674  043117  026506  044514
                016702  042516     000
1454            016705     046  050046  052125          P1MSG2: .ASCIZ  /&&PUT 10 BLANK CARDS IN EACH INPUT HOPPER/
                016712  030440  020060  046102
                016720  047101  020113  040503
                016726  042122  020123  047111
                016734  042440  041501  020110
                016742  047111  052520  020124
                016750  047510  050120  051105
                016756     000
1455            016757     046  052520  020124          P1MSG3: .ASCIZ  /&PUT 8035-8045 ON-LINE AND PRESS  START ON THE 8035-804
                016764  030070  032463  034055
                016772  032060  020065  047117
                017000  046055  047111  020105
                017006  047101  020104  051120
                017014  051505  020123  051440
                017022  040524  052122  047440
                017030  020116  044124  020105
                017036  030070  032463  034055
                017044  032060  000065
1456            017050  053046  051105  043111          P2MSG1: .ASCIZ  /&VERIFY THAT THERE ARE 10 BLANK CARDS IN EACH STACKER/
                017056  020131  044124  052101
                017064  052040  042510  042522
                017072  040440  042522  030440
                017100  020060  046102  047101
                017106  020113  040503  042122
                017114  020123  047111  042440
                017122  041501  020110  052123
                017130  041501  042513  000122
```

H 4

SEQ 0046

```
1457  017136  050046  040514  042503          P2MSG2: .ASCIZ  /&PLACE ALL 20 CARDS IN THE PRIMARY HOPPER/
      017144  040440  046114  031040
      017152  020060  040503  042122
      017160  020123  047111  052040
      017166  042510  050040  044522
      017174  040515  054522  044040
      017202  050117  042520  000122
1458  017210  050046  042522  051523          P2MSG3: .ASCIZ  /&PRESS STOP-RESET  AND PRESS START ON THE 8035-8045/
      017216  051440  047524  026520
      017224  042522  042523  020124
      017232  040440  042116  050040
      017240  042522  051523  051440
      017246  040524  052122  047440
      017254  020116  044124  020105
      017262  030070  032463  034055
      017270  032060  000065
1459  017274  050046  042522  051523                  P2MSG4: .ASCIZ  /&PRESS STOP-RESET ON 8035-8045 /
      017302  051440  047524  026520
      017310  042522  042523  020124
      017316  047117  034040  031460
      017324  026465  030070  032464
      017332  000040
1460  017334  053046  051105  043111          P3MSG1: .ASCIZ  /&VERIFY THAT ALL CHARS WERE PUNCHED ON EACH CARD IN A/
      017342  020131  044124  052101
      017350  040440  046114  041440
      017356  040510  051522  053440
      017364  051105  020105  052520
      017372  041516  042510  020104
      017400  047117  042440  041501
      017406  020110  040503  042122
      017414  044440  020116  000101
1461  017422  050046  042522  042503          P3MSG2: .ASCIZ  /&PRECESS FASHION BEGINNING W.TH COLUMN 11 OF 1ST CARD/
      017430  051523  043040  051501
      017436  044510  047117  041040
      017444  043505  047111  044516
      017452  043516  053440  052111
      017460  020110  047503  052514
      017466  047115  030440  020061
      017474  043117  030440  052123
      017502  041440  051101  000104
1462  017510  053046  051105  043111          P4MSG1: .ASCIZ  /&VERIFY THAT THERE IS A ALL A'S CHAR PUNCHED IN CARD 1/
      017516  020131  044124  052101
      017524  052040  042510  042522
      017532  044440  020123  020101
      017540  046101  020114  023501
      017546  020123  044103  051101
      017554  050040  047125  044103
      017562  042105  044440  020116
      017570  040503  042122  030440
      017576     000
1463  017577     046  044124  052101          P4MSG2: .ASCIZ  /&THAT IS HOLE PUNCHED IN ROW 12 AND 1/
      017604  044440  020123  047510
      017612  042514  050040  047125
      017620  044103  042105  044440
      017626  020116  047522  020127
      017634  031061  040440  042116
```

```
                017642  030440      000
1464            017645      046  050125  052040  P4MSG3: .ASCIZ  /&UP TO 20 ''ALL A'S'' CHARS PUNCHED IN CARD 20/
                017652  020117  030062  021040
                017660  046101  020114  023501
                017666  021123  041440  040510
                017674  051522  050040  047125
                017702  044103  042105  044440
                017710  020116  040503  042122
                017716  031040  000060
1465            017722  023046  047105  020104  ENDMSG: .ASCIZ  /&&END OF TESTING&&/
                017730  043117  052040  051505
                017736  044524  043516  023046
                017744      000
1466            017745      046  020046  042504  PROGNM: .ASCIZ /&& DECSPEC-11-CZCTBA-CTS11-JC-HOLLERITH 12-BITS-TEST &&/
                017752  051503  042520  026503
                017760  030461  041455  041532
                017766  041124  026501  052103
                017774  030523  026461  041512
                020002  044055  046117  042514
                020010  044522  044124  030440
                020016  026462  044502  051524
                020024  052055  051505  020124
                020032  023046      000
1467
1468                    020036                  .EVEN
1469            020036  000000  BF20:   0
1470            020040  000000  BF19:   0
1471            020042  000000  BF18:   0
1472            020044  000000  BF17:   0
1473            020046  000000  BF16:   0
1474            020050  000000  BF15:   0
1475            020052  000000  BF14:   0
1476            020054  000000  BF13:   0
1477            020056  000000  BF12:   0
1478            020060  000000  BF11:   0
1479            020062  000000  BF10:   0
1480            020064  000000  BF9:    0
1481            020066  000000  BF8:    0
1482            020070  000000  BF7:    0
1483            020072  000000  BF6:    0
1484            020074  000000  BF5:    0
1485            020076  000000  BF4:    0
1486            020100  000000  BF3:    0
1487            020102  000000  BF2:    0
1488            020104  000000  BF1:    0
1489            020106  000000          0
1490            020110  000000          0
1491            020112  000000          0
1492            020114  000000          0
1493            020116  000000          0
1494            020120  000000          0
1495            020122  000000          0
1496            020124  000000          0
1497            020126  000000          0
1498            020130  004006      4006
1499            020132  000006      6
```

J 4

SEQ 0048

```
1500  020134  000102                                  102     ;#
1501  020136  001042                                  1042    ;%
1502  020140  004000                                  4000    ;&
1503  020142  000022                                  22      ;'
1504  020144  004022                                  4022    ;(
1505  020146  002022                                  2022    ;)
1506  020150  002042                                  2042    ;*
1507  020152  004012                                  4012    ;+
1508  020154  001102                                  1102    ;,
1509  020156  002000                                  2000    ;-
1510  020160  001400                                  1400    ;/
1511  020162  001000                                  1000    ;0
1512  020164  000400                                  400     ;1
1513  020166  000200                                  200     ;2
1514  020170  000100                                  100     ;3
1515  020172  000040                                  40      ;4
1516  020174  000020                                  20      ;5
1517  020176  000010                                  10      ;6
1518  020200  000004                                  4       ;7
1519  020202  000002                                  2       ;8
1520  020204  000001                                  1       ;9
1521  020206  000202                                  202     ;:
1522  020210  002012                                  2012    ;;
1523  020212  004042                                  4042    ;<
1524  020214  000012                                  12      ;=
1525  020216  001012                                  1012    ;>
1526  020220  001006                                  1006    ;?
1527  020222  000042                                  42      ;@
1528  020224  004400                                  4400    ;A
1529  020226  004200                                  4200    ;B
1530  020230  004100                                  4100    ;C
1531  020232  004040                                  4040    ;D
1532  020234  004020                                  4020    ;E
1533  020236  004010                                  4010    ;F
1534  020240  004004                                  4004    ;G
1535  020242  004002                                  4002    ;H
1536  020244  004001                                  4001    ;I
1537  020246  002400                                  2400    ;J
1538  020250  002200                                  2200    ;K
1539  020252  002100                                  2100    ;L
1540  020254  002040                                  2040    ;M
1541  020256  002020                                  2020    ;N
1542  020260  002010                                  2010    ;O
1543  020262  002004                                  2004    ;P
1544  020264  002002                                  2002    ;Q
1545  020266  002001                                  2001    ;R
1546  020270  001200                                  1200    ;S
1547  020272  001100                                  1100    ;T
1548  020274  001040                                  1040    ;U
1549  020276  001020                                  1020    ;V
1550  020300  001010                                  1010    ;W
1551  020302  001004                                  1004    ;X
1552  020304  001002                                  1002    ;Y
1553  020306  001001                                  1001    ;Z
1554  020310  004202                                  4202    ;[
1555  020312  002202                                  2202    ;]
```

K 4

SEQ 0049

```
1556  020314  002006          2006    ;^
1557  020316  001022          1022    ;
1558
1559  020320  000000          0
1560  020322  000000          0
1561  020324  000000          0
1562  020326  000000          0
1563  020330  000000          0
1564  020332  000000          0
1565  020334  000000          0
1566  020336  000000          0
1567  020340  000000          0
1568  020342  000000          0
1569  020344  000000          0
1570  020346  000000          0
1571  020350  000000          0
1572  020352  000000          0
1573  020354  000000          0
1574  020356  000000          0
1575  020360  000000          0
1576  020362  000000          0
1577  020364  000000          0
1578  020366  000000          0
1579  020370  000000          0
1580  020372  000000          0
1581  020374  000000          0
1582  020376  000000          0
1583  020400  000000          0
1584  020402  000000          0
1585  020404  004100          CARD01: 4100        ;C
1586  020406  004400                  4400        ;A
1587  020410  002001                  2001        ;R
1588  020412  004040                  4040        ;D
1589  020414  000000                  0           ;SP
1590  020416  001000                  1000        ;0
1591  020420  000400                  400         ;1
1592  020422  004100          CARD02: 4100        ;C
1593  020424  004400                  4400        ;A
1594  020426  002001                  2001        ;R
1595  020430  004040                  4040        ;D
1596  020432  000000                  0           ;SP
1597  020434  001000                  1000        ;0
1598  020436  000200                  200         ;2
1599  020440  004100          CARD03: 4100        ;C
1600  020442  004400                  4400        ;A
1601  020444  002001                  2001        ;R
1602  020446  004040                  4040        ;D
1603  020450  000000                  0           ;SP
1604  020452  001000                  1000        ;0
1605  020454  000100                  100         ;3
1606  020456  004100          CARD04: 4100        ;C
1607  020460  004400                  4400        ;A
1608  020462  002001                  2001        ;R
1609  020464  004040                  4040        ;D
1610  020466  000000                  0           ;SP
1611  020470  001000                  1000        ;0
```

```
1612  020472  000040                                    40          ;4
1613  020474  004100                          CARD05:   4100        ;C
1614  020476  004400                                    4400        ;A
1615  020500  002001                                    2001        ;R
1616  020502  004040                                    4040        ;D
1617  020504  000000                                    0           ;SP
1618  020506  001000                                    1000        ;0
1619  020510  000020                                    20          ;5
1620  020512  004100                          CARD06:   4100        ;C
1621  020514  004400                                    4400        ;A
1622  020516  002001                                    2001        ;R
1623  020520  004040                                    4040        ;D
1624  020522  000000                                    0           ;SP
1625  020524  001000                                    1000        ;0
1626  020526  000010                                    10          ;6
1627  020530  004100                          CARD07:   4100        ;C
1628  020532  004400                                    4400        ;A
1629  020534  002001                                    2001        ;R
1630  020536  004040                                    4040        ;D
1631  020540  000000                                    0           ;SP
1632  020542  001000                                    1000        ;0
1633  020544  000004                                    4           ;7
1634  020546  004100                          CARD08:   4100        ;C
1635  020550  004400                                    4400        ;A
1636  020552  002001                                    2001        ;R
1637  020554  004040                                    4040        ;D
1638  020556  000000                                    0           ;SP
1639  020560  001000                                    1000        ;0
1640  020562  000002                                    2           ;8
1641  020564  004100                          CARD09:   4100        ;C
1642  020566  004400                                    4400        ;A
1643  020570  002001                                    2001        ;R
1644  020572  004040                                    4040        ;D
1645  020574  000000                                    0           ;SP
1646  020576  001000                                    1000        ;0
1647  020600  000001                                    1           ;9
1648  020602  004100                          CARD10:   4100        ;C
1649  020604  004400                                    4400        ;A
1650  020606  002001                                    2001        ;R
1651  020610  004040                                    4040        ;D
1652  020612  000000                                    0           ;SP
1653  020614  000400                                    400         ;1
1654  020616  001000                                    1000        ;0
1655  020620  004100                          CARD11:   4100        ;C
1656  020622  004400                                    4400        ;A
1657  020624  002001                                    2001        ;R
1658  020626  004040                                    4040        ;D
1659  020630  000000                                    0           ;SP
1660  020632  000400                                    400         ;1
1661  020634  000400                                    400         ;1
1662  020636  004100                          CARD12:   4100        ;C
1663  020640  004400                                    4400        ;A
1664  020642  002001                                    2001        ;R
1665  020644  004040                                    4040        ;D
1666  020646  000000                                    0           ;SP
1667  020650  000400                                    400         ;1
```

```
  1668  020652  000200                                    200         ;2
  1669  020654  004100                            CARD13: 4100        ;C
  1670  020656  004400                                    4400        ;A
  1671  020660  002001                                    2001        ;R
  1672  020662  004040                                    4040        ;D
  1673  020664  000000                                    0           ;SP
  1674  020666  000400                                    400         ;1
  1675  020670  000100                                    100         ;3
  1676  020672  004100                            CARD14: 4100        ;C
  1677  020674  004400                                    4400        ;A
  1678  020676  002001                                    2001        ;R
  1679  020700  004040                                    4040        ;D
  1680  020702  000000                                    0           ;SP
  1681  020704  000400                                    400         ;1
  1682  020706  000040                                    40          ;4
  1683  020710  004100                            CARD15: 4100        ;C
  1684  020712  004400                                    4400        ;A
  1685  020714  002001                                    2001        ;R
  1686  020716  004040                                    4040        ;D
  1687  020720  000000                                    0           ;SP
  1688  020722  000400                                    400         ;1
  1689  020724  000020                                    20          ;5
  1690  020726  004100                            CARD16: 4100        ;C
  1691  020730  004400                                    4400        ;A
  1692  020732  002001                                    2001        ;R
  1693  020734  004040                                    4040        ;D
  1694  020736  000000                                    0           ;SP
  1695  020740  000400                                    400         ;1
  1696  020742  000010                                    10          ;6
  1697  020744  004100                            CARD17: 4100        ;C
  1698  020746  004400                                    4400        ;A
  1699  020750  002001                                    2001        ;R
  1700  020752  004040                                    4040        ;D
  1701  020754  000000                                    0           ;SP
  1702  020756  000400                                    400         ;1
  1703  020760  000004                                    4           ;7
  1704  020762  004100                            CARD18: 4100        ;C
  1705  020764  004400                                    4400        ;A
  1706  020766  002001                                    2001        ;R
  1707  020770  004040                                    4040        ;D
  1708  020772  000000                                    0           ;SP
  1709  020774  000400                                    400         ;1
  1710  020776  000002                                    2           ;8
  1711  021000  004100                            CARD19: 4100        ;C
  1712  021002  004400                                    4400        ;A
  1713  021004  002001                                    2001        ;R
  1714  021006  004040                                    4040        ;D
  1715  021010  000000                                    0           ;SP
  1716  021012  000400                                    400         ;1
  1717  021014  000001                                    1           ;9
  1718  021016  004100                            CARD20: 4100        ;C
  1719  021020  004400                                    4400        ;A
  1720  021022  002001                                    2001        ;R
  1721  021024  004040                                    4040        ;D
  1722  021026  000000                                    0           ;SP
  1723  021030  000200                                    200         ;2
```

```
1724  021032  001000                                    1000      ;0
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735                                                    .EVEN
1736  021034  000000                                ADR:    XX
1737  021036  000000                                BAD:    XX
1738  021040  000000                                CARD:   XX
1739  021042  000000                                CNT:    XX
1740  021044  000000                                CNT1:   XX
1741  021046  000000                                GOOD:   XX
1742  021050  000000                                HOLD:   XX
1743  021052  000000                                INPTR:  XX
1744  021054  000000                                LASTPC: XX
1745  021056  000000                                LEVEL:  XX
1746  021060  000000                                LOC:    XX
1747  021062  000000                                OUTPTR: XX
1748  021064  000000                                PT:     XX
1749  021066  000000                                PTR:    XX
1750  021070  000000                                SCOPE:  XX
1751  021072  000000                                TEMP:   XX
1752  021074  000000                                MPSAV:  XX
1753  021076  000000                                TOG1:   XX
1754
1755                                                ;
1756                                                ;******** DEVICE ADDRESSES ********
1757                                                ;
1758  021100  177160                                TCR:    177160
1759  021102  177161                                TCRHI:  177161
1760  021104  177162                                TSR:    177162
1761  021106  177163                                TSRHI:  177163
1762  021110  177164                                TDR:    177164
1763  021112  177165                                TDRHI:  177165
1764  021114  177166                                TIR:    177166
1765  021116  177167                                TIRHI:  177167
1766                                                ;
1767                                                ;******** INTRERRUPT VECTOR ADDRESSES ********
1768                                                ;
1769  021120  000230                                PCV:    230
1770  021122  000232                                PSV:    232
1771                                                ;
1772                                                ;******** BUS REQUEST LEVEL ********
1773                                                ;
1774  021124  000004                                BRLV:   4
1775
1776          000001                                        .END
```

```
ADR     021034          1140* 1141* 1143  1144  1150  1156  1736#
AENVM   000706           331# 378
APASS   000704           330# 377*
ASWREG  000710           332# 380
AUTOB   000712           333# 1247
A1      014622          1350#
A2      014626          1352# 1373  1376
A3      014636          1351  1356#
A4      014660          1360  1363#
A5      014666          1362  1365#
A6      014726          1374  1377#
A7      014712          1358  1372#
A8      014736          1366* 1369  1381#
B     = 000001           245#
BAD     021036           394* 395* 396* 405* 411* 435* 436* 437* 460* 461  466* 476* 477*
                         480* 483* 484* 497* 500* 501* 525* 526* 531* 532* 543* 548* 573*
                         574* 598* 624* 627* 633* 634* 648* 649* 654* 656* 663* 664* 668*
                         670* 677* 678* 685* 686* 690* 691* 713* 714* 720* 724* 725* 726*
                         737* 738* 742* 743* 758* 759* 784* 785* 798* 799* 800* 811* 812*
                         818* 819* 848* 849* 865* 866* 867* 882* 883* 890* 891* 914* 915
                         921* 1029* 1030 1035  1039  1155  1737#
BC1     012252           911# 912  920
BDMSG   015631          1153  1439#
BF1     020104           716  778  842  928  936  1488#
BF10    020062          1479#
BF11    020060          1478#
BF12    020056          1477#
BF13    020054          1476#
BF14    020052          1475#
BF15    020050          1474#
BF16    020046          1473#
BF17    020044          1472#
BF18    020042          1471#
BF19    020040          1470#
BF2     020102          1487#
BF20    020036           926  934  1469#
BF3     020100          1486#
BF4     020076          1485#
BF5     020074          1484#
BF6     020072          1483#
BF7     020070          1482#
BF8     020066          1481#
BF9     020064          1480#
BLKCRD  012242           647  655  662  669  675  684  711  735  909#
BRLV    021124           459  1089 1126* 1774#
B1      014776          1393  1396#
CARD    021040           789* 793* 807* 852* 858* 875* 958  978  1738#
CARD01  020404           853  1585#
CARD02  020422          1592#
CARD03  020440          1599#
CARD04  020456          1606#
CARD05  020474          1613#
CARD06  020512          1620#
CARD07  020530          1627#
CARD08  020546          1634#
CARD09  020564          1641#
```

```
CARD10  020602          1648#
CARD11  020620          1655#
CARD12  020636          1662#
CARD13  020654          1669#
CARD14  020672          1676#
CARD15  020710          1683#
CARD16  020726          1690#
CARD17  020744          1697#
CARD18  020762          1704#
CARD19  021000          1711#
CARD20  021016          1718#
CB1     012400           927#   929
CD1     013012          1023#  1024   1047
CHARCN  015010          1394*  1399#
CKDATA  013000           781    795    809    845    862    879   1021#
CKSWR   014156           614    900    910    943    960   1000   1131   1177   1239#
CLRBFS  012374           707    926#
CNT     021042           589*   590*   592*   594*   596*   628*   629*   631*  1197*  1205*  1210*  1213*  1220*
                        1230*  1739#
CNTLG   014463          1249   1297#
CNTLU   014456          1262   1296#
CNTO    014740          1339*  1372*  1383#
CNT1    021044           651*   657*   672*   679*   717*   727*   788*   801*   851*   868*  1740#
COL   = 000004           242#   917   1032   1037   1041
COLMSG  015640          1159   1440#
CONMSG  015550           899   1436#
CONTIN  012212           392    433    511    516    552    559    577    587    613    622    706    776    838
                         898#
CONT1   012222           901#   903
CRDTST  016570           610   1452#
CRLF    014512          1273   1303#
DCMSG1  015116          1077   1428#
DCMSG2  015164          1079   1106   1429#
DCMSG3  015213          1083   1116   1430#
DCMSG4  015257          1087   1123   1431#
DCMSG5  015311          1092   1432#
DCMSG6  015377          1104   1433#
DC1     013322          1093#  1099
DC2     013344          1095   1098#
DC3     013360          1097   1103#
DC4     013410          1111#  1114
DDISP = 177570           326#   329    367
DEVCOD  013262           348   1076#
DISPLA  000702           329#   367*   375*
DISPRE  000174           319#   375
DSWR  = 177570           325#   328    366
ENDMSG  017722           359   1465#
ERR   = 104000           240#   394    395    396    405    411    435    436    437    454    463    466    476
                         477    480    483    484    493    497    500    501    525    526    531    532    543
                         548    562    573    574    598    624    627    633    634    643    648    649    654
                         656    663    664    668    670    677    678    685    686    690    691    713    714
                         720    724    725    726    737    738    742    743    752    758    759    784    785
                         798    799    800    811    812    818    819    848    849    865    866    867    882
                         883    890    891    917    921   1032   1037   1041
ERRHLT  013656          1165#
ERROR   013476           316   1131#
```

```
ERRVEC= 000004            324#    364     365*    376*
E1      013576           1145    1150#
E2      013616           1151    1156#
E3      013636           1137    1157    1161#
E4      013660           1164    1166#
FILLO   014741           1333*   1337*   1347    1384#
G     = 000002            244#
GB    = 000003            243#    394     395     396     405     411     435     436     437     463     466     476     477
                          480     483     484     497     500     501     525     526     531     532     543     548     573
                          574     598     624     627     633     634     648     649     654     656     663     664     668
                          670     677     678     685     686     690     691     713     714     720     724     725     726
                          737     738     742     743     758     759     784     785     798     799     800     811     812
                          818     819     848     849     865     866     867     882     883     890     891     917     921
                         1032    1037    1041
GDMSG   015622           1147    1438#
GOOD    021046            394*    395*    396*    405*    411*    435*    436*    437*    459*    461     466*    476*    477*
                          480*    483*    484*    497*    500*    501*    525*    526*    531*    532*    543*    548*    573*
                          574*    598*    624*    627*    633*    634*    648*    649*    654*    656*    663*    664*    668*
                          670*    677*    678*    685*    686*    690*    691*    713*    714*    720*    724*    725*    726*
                          737*    738*    742*    743*    758*    759*    784*    785*    798*    799*    800*    811*    812*
                          818*    819*    848*    849*    865*    866*    867*    882*    883*    890*    891*    913*    915
                          921*   1028*   1030    1149    1741#
GTSWR   014226            345    1250#
HOLD    021050            403*    404     405     1742#
INIT    001222            352     386#
INITM1  015475            389    1435#
INIT1   001242            393#    394     395     396
INPTR   021052            778*    782*    796*    842*    846*    863*    880*   1022    1743#
INTAG   000713            334#   1274
K10     014310           1261    1265#
K11     014332           1268    1270#
K14     014336           1273#
K15     014360           1240    1242    1246    1248    1275    1277#
K16     014362           1266    1278#
K17     014422           1285    1289#
K18     014434           1280    1282    1292#
K19     014246           1254#   1264
K20     014302           1263#   1293
K7      014252           1256#   1257    1291
K9      014270           1260#
LASTPC  021054            394     395     396     405     411     435     436     437     455     464     466     476     477
                          480     483     484     494     497     500     501     525     526     531     532     543     548
                          564     573     574     598     624     627     633     634     644     648     649     654     656
                          663     664     668     670     677     678     685     686     690     691     713     714     720
                          724     725     726     737     738     742     743     753     758     759     784     785     798
                          799     800     811     812     818     819     848     849     865     866     867     882     883
                          890     891     921    1135*   1744#
LEVEL   021056            444*    452*    460    1745#
LF      014513           1304#   1396
LOC     021060           1415*   1416*   1746#
LOGTST  015441            387    1434#
LP    = 012320            394#    395#    396#    405#    411#    435#    436#    437#    466#    476#    477#    480#    483#
                          484#    497#    500#    501#    525#    526#    531#    532#    543#    548#    573#    574#    598#
                          624#    627#    633#    634#    648#    649#    654#    656#    663#    664#    668#    670#    677#
                          678#    685#    686#    690#    691#    713#    714#    720#    724#    725#    726#    737#    738#
                          742#    743#    758#    759#    784#    785#    798#    799#    800#    811#    812#    818#    819#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0056

```
                    848#    849#    865#    866#    867#    882#    883#    890#    891#    921#
MNEW    014500     1253    1300#
MODE0   014742     1334*   1338*   1343    1346*   1357*   1385#
MP    = 000002     1019#   1025
MPSAV   021074     1027*   1033    1043*   1752#
MSWR    014470     1250    1298#
OCTDEC  013144     1052#   1160
OD1     013160     1055#   1059    1064
OD2     013204     1057    1061#
OD3     013222     1060    1065#
ONES    013254     1053*   1055*   1056    1061*   1068    1072#
OUTPTR  021062      853*   1003    1005*   1747#
PASS1   004364      356     608#
PASS2   006366      693     700#
PASS3   007726      761     768#
PASS4   011030      820     827#
PCMSG   015576     1139    1437#
PCV     021120      445*    487*    637*    746*    1085    1113    1119*   1769#
PROGNM  017745      344    1466#
PS    = 177776      225#    339*    443*    449*    489*    492*    639*    642*    748*    751*
PSV     021122      446*    488*    638*    747*    1121*   1770#
PT      021064     1195*   1196    1748#
PTR     021066      716*    722*    733*    942     1022*   1028    1045*   1749#
P1A     004756      639#    644
P1B     005032      645     647#
P1D     005170      652#    658
P1E     005666      673#    680
P1INT   005026      637     646#
P1MSG1  016623      612     1453#
P1MSG2  016705      618     1454#
P1MSG3  016757      620     1455#
P2A     006604      718#    728
P2B     007526      752#    753
P2C     007562      754     756#
P2INT   007556      746     755#
P2MSG1  017050      701     1456#
P2MSG2  017136      703     773     835     1457#
P2MSG3  017210      528     545     705     775     837     1458#
P2MSG4  017274      391     1459#
P2SD1   012450      945#    946     950
P2SEND  012434      721     732     756     942#
P3A     010160      790#    802
P3MSG1  017334      769     1460#
P3MSG2  017422      771     1461#
P3SD1   012526      961#    962     966
P3SD2   012554      967#    968     972
P3SEND  012510      792     806     816     958#
P4A     011300      854#    869
P4MSG1  017510      828     1462#
P4MSG2  017577      831     1463#
P4MSG3  017645      833     1464#
P4SD1   012704     1001#   1002    1008
P4SD2   012744     1009#   1010    1014
P4SEND  012674      857     877     888     999#
P5SD1   012616      980#    981     985
P5SD2   012644      986#    987     991
```

```
P5SEND  012604            856     876     887     978#
QUES    014511           1292    1302#
RDOCT   014070           1108    1118    1125    1220#
RD.1    014100           1222#   1231
RT1     001150            365     372#
RT2     001156            371     372     374#
RT3     001172            369     376#
RT4     001220            379     381#
SB1     012420            935#    937
SCOPE   021070            341*    394     395     396     405     411     435     436     437     455     464     466     476
                          477     480     483     484     494     497     500     501     525     526     531     532     543
                          548     564     573     574     598     624     627     633     634     644     648     649     654
                          656     663     664     668     670     677     678     685     686     690     691     713     714
                          720     724     725     726     737     738     742     743     753     758     759     784     785
                          798     799     800     811     812     818     819     848     849     865     866     867     882
                          883     890     891     921    1134*   1750#
SETBFS  012414            839     934#
SR    = 177570            226#
START   001000            337     339#
ST1     001046            347     350#    361
ST2     001062            351     354#
ST3     001076            355     358#
SWR     000700            328#    346     350     354     366*    368     374*    380*   1132    1136    1163    1239    1269*
                         1406
SWRCK   001110            342     364#
SWREG   000176            320#    374    1239    1251
S7$     014070           1217#
TCR     021100            394     404*    405     409*    411     435     448*    466     477     480     484     486*    497
                          499*    500     526     532     554*    574     588*    624     626*    627     636*    649     652*
                          656     660*    664     666*    670     673*    676     678     682*    686     688*    689     691
                          709*    712     714     718*    723     725     730*    736     738     740*    743     745*    757
                          759     779*    783     785     790*    797     798     804*    810     812     814*    817     819
                          843*    847     849     854*    864     865     871*    881     883     885*    889     891    1081
                         1110    1758#
TCRHI   021102            475*    482*    513*    534*    616*   1759#
TDR     021110            914     947*    963*    982*   1003*   1011*   1029   1762#
TDRHI   021112            969*    988*   1763#
TEMP    021072           1198*   1199*   1200*   1201    1221*   1225*   1226*   1227*   1228*   1229*   1232    1751#
TENS    013256           1054*   1062*   1065    1073#
TEST1   001446            398     402#
TEST2   001664            413     431#
TEST3   002076            439     443#    455     464
TEST4   002342            468     475#    476     477
TEST5   003210            503     509#
TEST6   004222            580     584#
TIR     021114            396     437     634     653     661     667     674     683     710     726     734     742     780
                          794     800     808     844     859     860     867     878     911     921    1023   1764#
TIRHI   021116            719     731     741     791     805     815     855     872     873     886     945     951     961
                          967     980     986    1001    1009   1765#
TKB   = 177562            228#    1243    1258    1425
TKS   = 177560            227#    1241    1256    1276*   1422*   1423
TM1     013710           1180#   1186
TM2     013726           1183    1185#
TM3     013734           1181    1187#
TOG     013260           1052*   1058*   1063*   1074#
TOG1    021076            958*    965*    978*    984*    999*   1006*   1007    1012*   1013    1753#
```

```
TO.1     013770          1198#   1206
TO.2     014042          1208    1210#
TO.3     014046          1211#   1214
TPB    = 177566           230#   1391*   1408*
TPS    = 177564           229#   1389    1409
TSR      021104           395     436     476     483     501     525     531     543     548     573     598     633     648
                          654     663     668     677     685     690     713     720     724     737     758     784     799
                          811     818     848     866     882     890    1025    1760#
TSRH1    021106           523     529     541     546     560     571    1761#
TTI.1    015074          1423#   1424
TTO.1    015030          1409#   1410
TTO.2    015052          1412    1415#
TTO.3    015066          1407    1418#
TTYIN    015070           901    1093    1222    1422#
TTYN     014442          1294#
TTYOUT   015014           904    1067    1070    1096    1100    1162    1185    1212    1223    1406#   1414
TYPE   = 104400           241#    343     358     386     388     390     431     509     514     517     519     521     527
                          535     537     539     544     550     555     557     565     567     569     575     585     609
                          611     617     619     700     702     704     768     770     772     774     827     830     832
                          834     836     898    1076    1078    1080    1082    1084    1086    1088    1091    1103    1105
                         1115    1122    1138    1142    1146    1148    1152    1154    1158    1249    1250    1253    1262
                         1273    1292    1369
TYPEC    014744          1278    1389#   1390
TYPEX    015012          1395    1397    1400#
TYPMSG   013700          1173    1177#
TYPOC    014542          1252    1337#
TYPOCT   013742          1174    1192#
TYPON    014556          1336    1339#
TYPOS    014516          1332#
TYPOUT   013662           322    1170#
T1A      001452           403#    407
T1B      001456           404#    405
T1C      001544           409#    411
T1END    001662           406     427#
T1LST    001632           402     415#
T2MSG1   015653           432     515    1441#
T3A      002134           449#    453
T3B      002262           456     465#    466
T3INT    002204           445     458#
T4A      002474           479#    480
T4B      002550           482#    483     484
T4C      002724           489#    494
T4D      003000           495     497#
T4E      003052           499#    500     501
T4INT    002774           487     496#
T5A      003254           523#    524     525
T5B      003414           529#    530     531
T5C      003572           541#    542     543
T5D      003660           546#    547     548
T5E      003776           560#    564
T5F      004050           571#    572     573
T5MSG1   016075           518     556    1444#
T5MSG2   016164           520     538     558    1508    1445#
T5MSG3   016234           522     540     570    1446#
T5MSG4   016305          1447#
T5MSG5   016344           536     566    1448#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0059

```
T5MSG6  016433            1449#
T5MSG7  016472             576    1450#
T5M1    015744             510    1442#
T5M2    016036             551    1443#
T6MSG1  016531             586    1451#
XX    = 000000             239#    360     953    1072    1073    1074    1165    1736    1737    1738    1739    1740    1741
                          1742    1743    1744    1745    1746    1747    1748    1749    1750    1751    1752    1753
.     = 021126             310#    314     315#    318#    321#    327#    336#    338#    394     395     396     405     411
                           435     436     437     455     462     464     466     476     477     480     483     484     494
                           497     500     501     525     526     531     532     543     548     561     564     573     574
                           591     593     595     597     598     624     627     630     632     633     634     644     648
                           649     653     654     656     661     663     664     667     668     670     674     676     677
                           678     683     685     686     689     690     691     710     712     713     714     719     720
                           723     724     725     726     731     734     736     737     738     741     742     743     753
                           757     758     759     780     783     784     785     791     794     797     798     799     800
                           805     808     810     811     812     815     817     818     819     844     847     848     849
                           855     859     861     864     865     866     867     872     874     878     881     882     883
                           886     889     890     891     916     921     952    1133    1294#   1331#   1417    1468#
```

```
.MAIN.  MACY11 30A(1052)  20-MAY-80  09:53  PAGE 6
CZCTBA.P11    20-MAY-80 08:54          CROSS REFERENCE TABLE -- MACRO NAMES                                    SEQ 0060

CKTCR     271#    394     405     411     435     466     477     480     484     497     500     526     532     574     624
          627     649     656     664     670     678     686     691     714     725     738     743     759     785     798
          812     819     849     865     883     891
CKTDR     291#
CKTIR     301#    396     437     634     726     742     800     867     921
CKTSR     281#    395     436     476     483     501     525     531     543     548     573     598     633     648     654
          663     668     677     685     690     713     720     724     737     758     784     799     811     818     848
          866     882     890
LOOP      261#    394     395     396     405     411     435     436     437     455     464     466     476     477     480
          483     484     494     497     500     501     525     526     531     532     543     548     564     573     574
          598     624     627     633     634     644     648     649     654     656     663     664     668     670     677
          678     685     686     690     691     713     714     720     724     725     726     737     738     742     743
          753     758     759     784     785     798     799     800     811     812     818     819     848     849     865
          866     867     882     883     890     891     921
WTDONE    247#    676     689     712     723     736     757     783     797     810     817     847     864     881     889
WTIDR     257#    653     661     667     674     683     710     734     780     794     808     844     859     878
WTODR     252#    719     731     741     791     805     815     855     872     886

. ABS.  021126     000


 ERRORS DETECTED:  0

 CZCTBA,CZCTBA/CRF=CZCTBA.P11
 RUN-TIME: 10 21 2 SECONDS
 RUN-TIME RATIO: 65/34=1.8
 CORE USED:  8K  (15 PAGES)
```