

# LSI-11

DECIMAL INSTRUCTION TEST  
MD-11-DVKAJ-A

EP DVKAJ A DL A  
COPYRIGHT 1977  
FICHE 1 OF 1

MAR 1977  
**digital**  
MADE IN USA

This section contains a grid of 150 small tables, arranged in 10 columns and 15 rows. Each table is a test case for the LSI-11 processor, likely for the DECIMAL INSTRUCTION TEST. The tables contain various data points, including hexadecimal values, binary representations, and test results. The text is very small and difficult to read, but the structure is consistent across all tables.

A small table located in the bottom right corner of the page. It contains several rows of data, possibly a summary or a specific test result. The text is small and difficult to read.



45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95

1.0 GENERAL INFORMATION  
-----

1.1 ABSTRACT  
-----

THIS DIAGNOSTIC VERIFIES THE OPERATION OF THE DIBOL DECIMAL INSTRUCTIONS OF THE LSI-11 (ADDN, SUBN, CMPN, CVTNL). THE PROGRAM CHECKS THAT EACH INSTRUCTION IS INTERRUPTABLE USING THE CONSOLE SLU INTERFACE (SEE PARA 2.3.4) AND RUNS ALTERNATE PASSES WITH THE TRACE TRAP ENABLED, UNLESS INHIBITED BY THE SWITCH REGISTER ( 2). THE PROGRAM IS DESIGNED TO RUN ON AN LSI-11 WITH 4K OF MEMORY AND THE DIS MICROMS. IT CAN BE RUN UNDER XXDP APT, AND ACT MONITORS. THE SOFTWARE SWITCH REGISTER IS AT LOCATION 176.

TO FULLY TEST THE LSI-11 DIBOL INSTRUCTION SET MICROMS, THE FOLLOWING DIAGNOSTICS MUST BE RUN:

MD-11-DVKAI\* DIS MOVE & STRING INSTRUCTION TEST  
MD-11-DVKAJ\* [THIS DIAGNOSTIC]  
MD-11-DVKAB\* LSI-11 EIS INSTRUCTION TEST

WHERE "\*" IS THE LATEST REVISION

1.2 SYSTEM REQUIREMENTS  
-----

1.2.1 EQUIPMENT  
-----

LSI-11(KD11-P) WITH A SERIAL LINE INTERFACE AND 4K OF MEMORY

1.2.2 STORAGE  
-----

THE PROGRAM USES MEMORY FROM 000000 TO 017040.

1.2.3 PRELIMINARY PROGRAMS  
-----

IT IS ASSUMED THAT THE FOLLOWING DIAGNOSTICS HAVE BEEN RUN:

LSI-11 BASIC CPU TEST MD-11-DVKAA\*  
LSI-11 TRAPS TEST MD-11-DVKAD\*

WHERE "\*" IS THE LATEST REVISION

97  
 98  
 99  
 100  
 101  
 102  
 103  
 104  
 105  
 106  
 107  
 108  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116  
 117  
 118  
 119  
 120  
 121  
 122  
 123  
 124  
 125  
 126  
 127  
 128  
 129  
 130  
 131  
 132  
 133  
 134  
 135  
 136  
 137  
 138  
 139  
 140

2.0 OPERATING INSTRUCTIONS  
 -----

2.1 LOADING PROCEDURES  
 -----

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES

2.2 STARTING PROCEDURE  
 -----

LOAD THE SWITCH REGISTER WITH THE DESIRED SETTING  
 (SOFTWARE SWITCH REGISTER LOCATION = 176)

THE PROGRAM SHOULD ALWAYS BE STARTED AT 200.  
 STARTING AT 200, THE PROGRAM CLEARS ALL PROGRAM PARAMETERS AND  
 THEN PRINTS ITS MAINDEC IDENTIFICATION. "END PASS" IS PRINTED  
 AT THE END OF EACH FULL PASS OF THE DIAGNOSTIC.

2.3 OPERATING PROCEDURES  
 -----

2.3.1 OPERATIONAL SWITCH REGISTER  
 -----

LOCATION 176 IS USED FOR THE SOFTWARE SWITCH REGISTER AND  
 THE FOLLOWING OPTIONS MAY BE SELECTED BY INSERTING A 1 IN THEIR  
 RESPECTIVE BIT POSITIONS.

- BIT15 - HALT ON ERROR
- BIT14 + SCOPE LOOP
- BIT13 - INHIBIT ERROR TYPEOUT
- BIT12 - INHIBIT TRACE TRAP
- BIT11 - UNUSED
- BIT10 - UNUSED
- BIT09 - LOOP ON ERROR
- BIT08 - LOOP ON TEST IN SWR<05:00>
- BIT07 - INHIBIT INTERRUPTABILITY TESTS

NOTE: ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BITS OF BYTE \$ENVM  
 HIGH.

141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
1742.3.2 RUNNING UNDER APT

THE APT MAILBOX-E-TABLE IS LOCATED AT LOCATION 566.

USING THE CONSOLE INTERFACE AS THE INTERRUPTING DEVICE, THE INTERRUPTABILITY TESTS WILL BE RUN ON ONLY THE FIRST PASS TO AVOID INTERFERENCE WITH THE APT INTERFACE. IF INTERRUPTABILITY TESTS ARE DESIRED ON ALL PASSES, ANOTHER SLU MUST BE SUPPLIED AND ITS RECEIVER STATUS REGISTER ADDRESS & ITS INTERRUPT VECTOR MUST BE PLACED IN THE APT E-TABLE AT LOCATIONS "\$BASE" & "\$VEC 1" RESPECTIVELY.

2.3.3 RUN WITH ALTERNATE CONSOLE ADDRESS

TO USE A CONSOLE ADDRESS OTHER THAN 177560, THE OPERATOR MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM AT THE LOCATIONS LABELED:

\$TKS: RCSR ADDRESS  
\$TKB: RBUF ADDRESS  
\$TPS: TCSR ADDRESS  
\$TPB: TBUF ADDRESS

2.3.4 RUN INTERRUPT TESTS WITH ALTERNATE SLU

TO USE A SERIAL LINE INTERFACE ADDRESS OTHER THAN THE STANDARD CONSOLE ADDRESS (177560), THE OPERATOR MUST SUPPLY THE CORRECT ADDRESS AND INTERRUPT VECTOR BY INSERTING THEM IN THE LOCATIONS LABELED:

\$BASE: \*RCSR ADDRESS\*  
\$VECT1: \*RECEIVER INTERRUPT VECTOR\*

175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205

2.4 EXECUTION TIMES

-----  
 THE GIVEN EXECUTION TIMES TAKE INTO ACCOUNT THE RANDOM CHARACTERISTIC OF THE INTERRUPT TESTS. THE EXECUTION TIME OF THE FIRST PASS IS APPROXIMATELY 10 SECONDS; BUT SUBSEQUENT PASSES WITH INTERRUPT TESTS ENABLED COULD TAKE AS LONG AS 32 SECONDS. THEREFORE THE 32 SECOND EXECUTION TIME IS USED. THE PASS TIME WITHOUT INTERRUPTS IS APPROXIMATELY 2 SECONDS.

3.0 ERROR REPORTING

-----  
 IF A ROUTINE FAILS AND THE INHIBIT ERROR TYPEOUT (BIT13) OF THE SWR IS NOT SET, THE PC OF THE ERROR IS PRINTED. THE OPERATOR CAN FIND THE ERROR REPORT IN THE COMMENT FIELD OF THAT PC LOCATION IN THE PROGRAM LISTING. IF HALT ON ERROR (BIT15) OF THE SWR IS SET THE PROGRAM WILL HALT AFTER PRINTING THE ERROR PC AND ENTER THE MACHINE ODT.

E.G.       XXXXXX           <--PC OF THE ERROR  
               XXXXXX           <--PC+2 OF THE HALT ON ERROR LOCATION  
               a               <--ODT ENTERED

WHERE "XXXXXX" IS AN OCTAL VALUE

206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246

4.0 SUBROUTINE ABSTRACTS  
-----

4.1 TRAPCATCHER  
-----

A ".+2 - HALT" SEQUENCE IS REPEATED FROM 0-776 TO CATCH ALL UNEXPECTED TRAPS. THUS ALL UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR+2, EXCEPT TRAPS TO LOCATION 0,4, & 10 WHICH GO TO THEIR RESPECTIVE REPORTING ROUTINES "TZERO", "TIMTRP" & "ILLTRP". THE OTHER EXCEPTION IS LOCATION 100 (RTC INTERRUPT VECTOR) WHICH CONTAINS A ".+2 - RTI" SEQUENCE (RETURNS FROM THE INTERRUPT).

4.2 SCOPE  
-----

THIS ROUTINE CALL IS PLACED BETWEEN EACH SUBTEST. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED & UPDATES THE TEST NUMBER. IF A SCOPE LOOP IS REQUESTED IT WILL JUMP TO THE START OF THE SUBTEST AT WHICH THE SCOPE LOOP IS REQUESTED.

4.3 ERROR  
-----

THIS ROUTINE CALL IS PLACED WHEREEVER AN ERROR REPORT IS DESIRED. THE LOWER BYTE OF THIS CALL IS USED AS THE ERROR NUMBER. THIS ROUTINE REPORTS ERRORS TO APT USING "\$APTYPE", TYPES ERRORS TO THE CONSOLE USING THE "\$STYPE" & "\$STYPOCT" ROUTINES, AND HANDLES ERROR RESPONSES VIA SWR SETTINGS.

4.4 \$POWER  
-----

THIS ROUTINE SAVES ALL GENERAL REGISTERS DURING POWER-DOWN AND RESTORES THEM AT POWER-UP. IF A POWER FAILURE OCCURS "POWER" IS PRINTED AT THE CONSOLE AFTER POWER IS RESTORED AND THE PROGRAM IS RESTARTED AT TEST# 1.

247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274

- 4.5 NPREP  
-----  
THIS ROUTINE IS USED TO STORE A COPY OF THE INSTRUCTION TEST ARGUMENTS TO BE STORED IN RC-->R5.
- 4.6 GENR  
-----  
THIS ROUTINE IS USED TO TRANSFER INSTRUCTION TEST ARGUMENTS TO THE GENERAL REGISTERS AND TO COPY THE STACK POINTER BEFORE THE TEST INSTRUCTION EXECUTION.
- 4.7 XPSW  
-----  
THIS ROUTINE IS USED TO STORED THE EXPECTED PSW OF THE INSTRUCTION TEST AND TO SET THE T-BIT IN THE EXPECTED PSW ON PASSES USING THE TRACE TRAP.
- 4.8 INTR  
-----  
THIS ROUTINE IS USED TO DETECT WHEN THE TEST INSTRUCTION HAS BEEN INTERRUPTED AND TO CONTINUE THE INTERRUPT STREAM UNTIL THE INSTRUCTION IS INTERRUPTED.

275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330

5.0 MISCELLANEOUS  
-----

5.1 STACK POINTER  
-----

STACK POINTER IS INITIALLY SET TO 600.

5.2 PASS COUNT  
-----

A 16 BIT LOCATION "\$PASS" IS USED TO KEEP THE PASS COUNT. IT IS CLEARED BY STARTING AT 200.

5.3 TEST NUMBER  
-----

A 16 BIT LOCATION "\$STNM" IS USED TO KEEP TRACK OF THE SUBTEST NUMBER. THIS NUMBER IS ALSO PLACED IN THE APT E-TABLE AT "\$TESTN" WHEN UNDER APT.

5.4 POWER FAIL  
-----

THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. AFTER POWERING DOWN AND THEN UP AGAIN, THE PROGRAM WILL RESTART FROM TEST# 1 (I.E., RESTARTS THE PASS THAT WAS INTERRUPTED) AFTER TYPING "POWER" TO THE CONSOLE. HOWEVER IF THE PROGRAM IS STORED IN MOS MEMORY THAT CAN NOT HOLD DATA WITH POWER DOWN, THEN THE PROGRAM WILL NOT RECOVER FROM A POWER FAIL.

5.5 EVENT INTERRUPTS  
-----

THIS DIAGNOSTIC CAN BE RUN WITH THE REAL TIME CLOCK ACTIVE (INTERRUPT = 100). LOCATION 100 POINTS TO LOCATION 102 WHICH CONTAINS AN "RTI" INSTRUCTION. THUS ON CLOCK INTERRUPTS, AN RTI IS EXECUTED TO HANDLE IT.

!

.ENABLE ABS  
.LIST ME  
.NLIST MC,MD,CND



```

351
352
353       .SBTTL  BASIC DEFINITIONS
354
355       ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
356       001100  STACK= 1100
357       .EQUIV  EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
358       .EQUIV  IOT,SCOPE     ;;BASIC DEFINITION OF SCOPE CALL
359
360       ;*MISCELLANEOUS DEFINITIONS
361       000011  HT= 11          ;;CODE FOR HORIZONTAL TAB
362       000012  LF= 12          ;;CODE FOR LINE FEED
363       000015  CR= 15          ;;CODE FOR CARRIAGE RETURN
364       000200  CRLF= 200       ;;CODE FOR CARRIAGE RETURN-LINE FEED
365       177776  PS= 177776     ;;PROCESSOR STATUS WORD
366       .EQUIV  PS,PSW
367       177774  STKLMT= 177774  ;;STACK LIMIT REGISTER
368       177772  PIRQ= 177772   ;;PROGRAM INTERRUPT REQUEST REGISTER
369       177570  DSWR= 177570   ;;HARDWARE SWITCH REGISTER
370       177570  DDISP= 177570  ;;HARDWARE DISPLAY REGISTER
371
372       ;*GENERAL PURPOSE REGISTER DEFINITIONS
373       000000  R0= %0          ;;GENERAL REGISTER
374       000001  R1= %1          ;;GENERAL REGISTER
375       000002  R2= %2          ;;GENERAL REGISTER
376       000003  R3= %3          ;;GENERAL REGISTER
377       000004  R4= %4          ;;GENERAL REGISTER
378       000005  R5= %5          ;;GENERAL REGISTER
379       000006  R6= %6          ;;GENERAL REGISTER
380       000007  R7= %7          ;;GENERAL REGISTER
381       000006  SP= %6          ;;STACK POINTER
382       000007  PC= %7          ;;PROGRAM COUNTER
383
384       ;*PRIORITY LEVEL DEFINITIONS
385       000000  PR0= 0          ;;PRIORITY LEVEL 0
386       000040  PR1= 40         ;;PRIORITY LEVEL 1
387       000100  PR2= 100        ;;PRIORITY LEVEL 2
388       000140  PR3= 140        ;;PRIORITY LEVEL 3
389       000200  PR4= 200        ;;PRIORITY LEVEL 4
390       000240  PR5= 240        ;;PRIORITY LEVEL 5
391       000300  PR6= 300        ;;PRIORITY LEVEL 6
392       000340  PR7= 340        ;;PRIORITY LEVEL 7
393
394       ;*"SWITCH REGISTER" SWITCH DEFINITIONS
395       100000  SW15= 100000
396       040000  SW14= 40000
397       020000  SW13= 20000
398       010000  SW12= 10000
399       004000  SW11= 4000
400       002000  SW10= 2000
401       001000  SW09= 1000
402       000400  SW08= 400
403       000200  SW07= 200
404       000100  SW06= 100
405       000040  SW05= 40
406       000020  SW04= 20

```

407 000010  
 408 000004  
 409 000002  
 410 000001  
 411  
 412  
 413  
 414  
 415  
 416  
 417  
 418  
 419  
 420  
 421  
 422  
 423  
 424  
 425  
 426  
 427  
 428  
 429  
 430  
 431  
 432  
 433  
 434  
 435  
 436  
 437  
 438  
 439  
 440  
 441  
 442  
 443  
 444  
 445  
 446  
 447  
 448  
 449  
 450  
 451  
 452  
 453  
 454  
 455  
 456  
 457  
 458  
 459  
 460  
 461  
 462

SW03= 10  
 SW02= 4  
 SW01= 2  
 SW00= 1  
 .EQUIV SW09,SW9  
 .EQUIV SW08,SW8  
 .EQUIV SW07,SW7  
 .EQUIV SW06,SW6  
 .EQUIV SW05,SW5  
 .EQUIV SW04,SW4  
 .EQUIV SW03,SW3  
 .EQUIV SW02,SW2  
 .EQUIV SW01,SW1  
 .EQUIV SW00,SW0  
  
 :\*DATA BIT DEFINITIONS (BIT00 TO BIT15)  
 BIT15= 100000  
 BIT14= 40000  
 BIT13= 20000  
 BIT12= 10000  
 BIT11= 4000  
 BIT10= 2000  
 BIT09= 1000  
 BIT08= 400  
 BIT07= 200  
 BIT06= 100  
 BIT05= 40  
 BIT04= 20  
 BIT03= 10  
 BIT02= 4  
 BIT01= 2  
 BIT00= 1  
 .EQUIV BIT09,BIT9  
 .EQUIV BIT08,BIT8  
 .EQUIV BIT07,BIT7  
 .EQUIV BIT06,BIT6  
 .EQUIV BIT05,BIT5  
 .EQUIV BIT04,BIT4  
 .EQUIV BIT03,BIT3  
 .EQUIV BIT02,BIT2  
 .EQUIV BIT01,BIT1  
 .EQUIV BIT00,BIT0  
  
 :\*BASIC "CPU" TRAP VECTOR ADDRESSES  
 ERRVEC= 4  
 RESVEC= 10  
 TBITVEC= 14  
 TRTVEC= 14  
 BPTVEC= 14  
 IOTVEC= 20  
 PWRVEC= 24  
 EMTVEC= 30  
 TRAPVEC= 34  
 TKVEC= 60  
 TPVEC= 64  
 PIRQVEC= 240

;; TIME OUT AND OTHER ERRORS  
 ;; RESERVED AND ILLEGAL INSTRUCTIONS  
 ;; "T" BIT  
 ;; TRACE TRAP  
 ;; BREAKPOINT TRAP (BPT)  
 ;; INPUT/OUTPUT TRAP (IOT) \*\*SCOPE\*\*  
 ;; POWER FAIL  
 ;; EMULATOR TRAP (EMT) \*\*ERROR\*\*  
 ;; "TRAP" TRAP  
 ;; TTY KEYBOARD VECTOR  
 ;; TTY PRINTER VECTOR  
 ;; PROGRAM INTERRUPT REQUEST VECTOR

MO1

MAIN MACY1! 27(1006) 21-DEC-76 11:53 PAGE 12  
CVKAJA.P11 20-DEC-76 15:02 BASIC DEFINITIONS

SEQ 0014

463	000200	APTSIZE=	200
464	000001	APTENV=	001
465	000100	APTSPool=	100
466	000040	APTCSUP=	040
467	171400	\$SWR=	171400
468	000300	\$SWRMK=	300
469	000020	TBIT=	20
470	000001	\$TN=	1
471	000001	N=	1
472	000000	X=	\$TN-1
473	177777	NXM=	177777
474	177560	ABASE=	177560
475	000060	AVECT1=	60
476	000000	.	=0

```

478
479
480      .SBTTL  TRAP CATCHER
481
482      000000      .=0
483      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2.HALT"
484      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
485      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
486      000174      .=174
487      000174      000300      DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
488      000176      000000      SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
489
490
491      000000      000000      .=0
492      000000      014702      TZERO      ;SET LOCATIONS 0,4,6 TO ERROR REPORTS
493      000002      000340      340
494      000004      014712      TIMTRP
495      000006      000340      340
496      000010      014722      ILLTRP
497      000012      000340      340
498
499
500
501      000100      000100      .=100
502      000100      000102      .WORD 102      ;HANDLE EVENT LINE INTERRUPTS
503      000102      000002      .WORD 2
504
505
506      000200      000200      .=200
507      000200      000167      000512      JMP START      ;STARTING LOCATION FOR PROGRAM
508
    
```

```

509
510      000400
511      .SBTTL  .=400
512      ACT11 HOOKS
513      ;:*****
514      ;HOOKS REQUIRED BY ACT11
515      $SVPC=.      :SAVE PC
516      .=46
517 000046 $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SECF
518      .=52
519 000052 .WORD 0      ;;2)SET LOC.52 TO ZERO
520      .= $SVPC      ;; RESTORE PC
521      .SBTTL  APT PARAMETER BLOCK
522
523      ;:*****
524      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
525      ;:*****
526      .SX=.      ;;SAVE CURRENT LOCATION
527      .=24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
528 000024 200      ;;JR APT START UP
529      .=44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
530 000044 $APTHDR      ;;POINT TO APT HEADER BLOCK
531      .=.SX      ;;RESET LOCATION COUNTER
532      ;:*****
533      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
534      ;INTERFACE SPEC.
535
536 000400 $APTHD:
537 000400 000000 $HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
538 000402 000566 $MBADR: .WORD $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
539 000404 000030 $TSTM: .WORD 30      ;;RUN TIM OF LONGEST TEST
540 000406 000040 $PASTM: .WORD 40      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
541 000410 000000 $UNITM: .WORD      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
542 000412 000027 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
543

```

002

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 15  
SVKASA.P11 20-DEC-76 15:02 APT PARAMETER BLOCK

SEQ 0017

544

545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600

000500  
000500 000000  
000502 000  
000503 000  
000504 000000  
000506 000000  
000510 000000  
000512 000000  
000514 000  
000515 001  
000516 000000  
000520 000000  
000522 000000  
000524 000000  
000526 000000  
000530 000000  
000532 000000  
000534 000  
000535 000  
000536 000000  
000540 177570  
000542 177570  
000544 177560  
000546 177562  
000550 177564  
000552 177566  
000554 000  
000555 002  
000556 012  
000557 000  
000560 000000  
000562 077  
000563 015  
000564 000012  
  
000566  
000566 000000  
000570 000000  
000572 000000  
000574 000000  
000576 000000  
000600 000000  
000602 000000  
000604 000000  
000606

```
.SBTTL COMMON TAGS
;*****
;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
;USED IN THE PROGRAM.
      . =500
$CMTAG:      . =500      ;; START OF COMMON TAGS
      .WORD      0
$TSTNM:     .BYTE      0      ;; CONTAINS THE TEST NUMBER
$ERFLG:     .BYTE      0      ;; CONTAINS ERROR FLAG
$ICNT:      .WORD      0      ;; CONTAINS SUBTEST ITERATION COUNT
$LPADR:     .WORD      0      ;; CONTAINS SCOPE LOOP ADDRESS
$LPERR:     .WORD      0      ;; CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL:     .WORD      0      ;; CONTAINS TOTAL ERRORS DETECTED
$ITEMB:     .BYTE      0      ;; CONTAINS ITEM CONTROL BYTE
$ERMAX:     .BYTE      1      ;; CONTAINS MAX. ERRORS PER TEST
$ERRPC:     .WORD      0      ;; CONTAINS PC OF LAST ERROR INSTRUCTION
$GDADR:     .WORD      0      ;; CONTAINS ADDRESS OF 'GOOD' DATA
$BDADR:     .WORD      0      ;; CONTAINS ADDRESS OF 'BAD' DATA
$GCDAT:     .WORD      0      ;; CONTAINS 'GOOD' DATA
$BDDAT:     .WORD      0      ;; CONTAINS 'BAD' DATA
      .WORD      0      ;; RESERVED--NOT TO BE USED
$AUTOB:     .BYTE      0      ;; AUTOMATIC MODE INDICATOR
$INTAG:     .BYTE      0      ;; INTERRUPT MODE INDICATOR
      .WORD      0
$SWR:       .WORD      DSWR      ;; ADDRESS OF SWITCH REGISTER
$DISPLAY:   .WORD      DDISP     ;; ADDRESS OF DISPLAY REGISTER
$TKS:       177560      ;; TTY KBD STATUS
$TKB:       177562      ;; TTY KBD BUFFER
$TPS:       177564      ;; TTY PRINTER STATUS REG. ADDRESS
$TPB:       177566      ;; TTY PRINTER BUFFER REG. ADDRESS
$NULL:      .BYTE      0      ;; CONTAINS NULL CHARACTER FOR FILLS
$FILLS:     .BYTE      2      ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
$FILLC:     .BYTE     12      ;; INSERT FILL CHARS. AFTER A "LINE FEED"
$TPFLG:     .BYTE      0      ;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
$ESCAPE:    0          ;; ESCAPE ON ERROR ADDRESS
$QUES:      .ASCII     /?/     ;; QUESTION MARK
$CRLF:      .ASCII     <15>    ;; CARRIAGE RETURN
$LF:        .ASCII     <12>    ;; LINE FEED
;*****
.SBTTL APT MAILBOX-ETABLE
;*****
.EVEN
$MAIL:      ;; APT MAILBOX
$MSGTY:     .WORD      AMSGTY   ;; MESSAGE TYPE CODE
$FATAL:     .WORD      AFATAL   ;; FATAL ERROR NUMBER
$TESTN:     .WORD      ATESTN   ;; TEST NUMBER
$PASS:      .WORD      APASS    ;; PASS COUNT
$DEVCT:     .WORD      ADEVCT   ;; DEVICE COUNT
$UNIT:      .WORD      AUNIT    ;; I/O UNIT NUMBER
$MSGAD:     .WORD      AMSGAD   ;; MESSAGE ADDRESS
$MSGLG:     .WORD      AMSGLG   ;; MESSAGE LENGTH
$ETABLE:    ;; APT ENVIRONMENT TABLE
```

601	000606	000	\$ENV:	.BYTE	AENV	:: ENVIRONMENT BYTE
602	000607	000	\$ENVM:	.BYTE	AENVM	:: ENVIRONMENT MODE BITS
603	000610	000000	\$SWREG:	.WORD	ASWREG	:: APT SWITCH REGISTER
604	000612	000000	\$USWR:	.WORD	AUSWR	:: USER SWITCHES
605	000614	000000	\$CPUOP:	.WORD	ACPUOP	:: CPU TYPE, OPTIONS
606			.*			BITS 15-11=CPU TYPE
607			.*			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
608			.*			11/70=06, PDQ=07, Q=10
609			.*			BIT 10=REAL TIME CLOCK
610			.*			BIT 9=FLOATING POINT PROCESSOR
611			.*			BIT 8=MEMORY MANAGEMENT
612	000616	000	\$MAMS1:	.BYTE	AMAMS1	:: HIGH ADDRESS, M.S. BYTE
613	000617	000	\$MTYP1:	.BYTE	AMTYP1	:: MEM. TYPE, BLK#1
614			.*			MEM. TYPE BYTE -- (HIGH BYTE)
615			.*			900 NSEC CORE=001
616			.*			300 NSEC BIPOLAR=002
617			.*			500 NSEC MOS=003
618	000620	000000	\$MADR1:	.WORD	AMADR1	:: HIGH ADDRESS, BLK#1
619			.*			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
620	000622	000	\$MAMS2:	.BYTE	AMAMS2	:: HIGH ADDRESS, M.S. BYTE
621	000623	000	\$MTYP2:	.BYTE	AMTYP2	:: MEM. TYPE, BLK#2
622	000624	000000	\$MADR2:	.WORD	AMADR2	:: MEM. LAST ADDRESS, BLK#2
623	000626	000	\$MAMS3:	.BYTE	AMAMS3	:: HIGH ADDRESS, M.S. BYTE
624	000627	000	\$MTYP3:	.BYTE	AMTYP3	:: MEM. TYPE, BLK#3
625	000630	000000	\$MADR3:	.WORD	AMADR3	:: MEM. LAST ADDRESS, BLK#3
626	000632	000	\$MAMS4:	.BYTE	AMAMS4	:: HIGH ADDRESS, M.S. BYTE
627	000633	000	\$MTYP4:	.BYTE	AMTYP4	:: MEM. TYPE, BLK#4
628	000634	000000	\$MADR4:	.WORD	AMADR4	:: MEM. LAST ADDRESS, BLK#4
629	000636	000060	\$VECT1:	.WORD	AVECT1	:: INTERRUPT VECTOR#1 BUS PRIORITY#1
630	000640	000000	\$VECT2:	.WORD	AVECT2	:: INTERRUPT VECTOR#2 BUS PRIORITY#2
631	000642	177560	\$BASE:	.WORD	ABASE	:: BASE ADDRESS OF EQUIPMENT UNDER TEST
632	000644		\$ETEND:			
633			.MEXIT			

634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671

.SBTTL ERROR POINTER TABLE

;\*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.  
 ;\*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
 ;\*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.  
 ;\*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).  
 ;\*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;\* EM ;:POINTS TO THE ERROR MESSAGE  
 ;\* DH ;:POINTS TO THE DATA HEADER  
 ;\* DT ;:POINTS TO THE DATA  
 ;\* DF ;:POINTS TO THE DATA FORMAT

000644

\$ERRTB:

S1LN:	.WORD	0	:SOURCE1 LENGTH
S1ADR:	.WORD		:SOURCE1 ADDRESS
S2LN:	.WORD	0	:SOURCE2 LENGTH
S2ADR:	.WORD	0	:SOURCE2 ADDRESS
DSTLN:	.WORD		:DESTINATION LENGTH
DSTAD:	.WORD	BUF	:DESTINATION ADDRESS
FILL:	.WORD		:FILL CHARACTER
TABLE:	.WORD	0	:TRANSLATION TABLE ADDRESS
TCSR:	177564		:TCSR ADDRESS OF SLU USED FOR INTERRUPTS
TBUF:	177566		:TBUF ADDRESS
TVECT:	64		:TRANSMIT INTERRUPT VECTOR
TPSW:	66		:AND PSW LOCATION
PCI:	0		:ADDRESS OF TEST INSTRUCTION TO INTERRUPT
CCODES:	0		:CONDITION CODES AFTER EXECUTION OF TEST INSTRUCTION
EXPPSW:	0		:EXPECTED PSW
SAYR6:	0		:STACK POINTER VALUE BEFORE TEST INSTRUCTION EXECUTION
BADR6:	0		:BAD STACK POINTER VALUE
OLDPC:	0		:ADDRESS WHERE UNEXPECTED TRAP OR INTERRUPT OCCURRED
TEMP:	0		
TEMP1:	0		
TEMP2:	0		



```

728 001240 062700 000004      ADD      #4,RO      ;ADJUST TO TCSR ADDRESS
729 001244 010037 000664      MOV      RO,@#TCSR  ;STORE TCSR ADDRESS
730 001250 005720              TST      (RO)+      ;ADJUST TO TBUF ADDRESS
731 001252 010037 000666      MOV      RO,@#TBUF  ;STORE TBUF ADDRESS
732 001256 013700 000636      MOV      @#$VECT1,RO ;GET SLU INTERRUPT VECTOR
733 001262 062700 000004      ADD      #4,RO      ;ADJUST TO TRANSMIT INTERRUPT VECTOR
734 001266 010037 000670      MOV      RO,@#TVECT ;STORE TRANSMIT INTERRUPT VECTOR
735 001272 005720              TST      (RO)+      ;ADJUST TO TRANSMIT INTERRUPT PSW
736 001274 010037 000672      MOV      RO,@#TPSW  ;STORE TRANSMIT INTERRUPT PSW LOCATION
737
738
739 001300 106427 000200      BEGIN:  M*PS      #200      ;SET PRIORITY TO 7
740

```

```

741
742
743
744
745
746 001304 000304
747 001306 004567 013420
748 001312 000000
749 001314 001550
750 001316 000001
751 001320 001551
752 001322 000000
753 001324 004767 013474
754 001330 013767 000010 177354
755 001336 013767 000012 177350
756 001344 012737 001402 000010
757 001352 012737 000340 000012
758 001360 004567 013460
759 001364 000217
760 001366 004767 013366
761 001372 000277
762
763 001374 076050
764
765 001376 104001
766
767
768 001400 000464
769
770 001402
771 001402 020067 177236
772 001406 001401
773 001410 104002
774
775
776 001412 020167 177230
777 001416 001401
778 001420 104003
779
780
781 001422 020267 177222
782 001426 001401
783 001430 104004
784
785
786 001432 020367 177214
787 001436 001401
788 001440 104005
789
790
791 001442 020467 177212
792 001446 001401
793 001450 104006
794
795
796 001452 020567 177200

```

```

*****
*TEST 1 TEST "ADDN" WITH SOURCE1 LENGTH =0 & SOURCE2 STRING VALID
*****
TST1: SCOPE
      JSR    R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      0      ;SOURCE1 LENGTH
      S1T1   ;SOURCE1 ADDRESS
      1      ;SOURCE2 LENGTH
      S2T1   ;SOURCE2 ADDRESS
      0      ;DESTINATION LENGTH
      JSR    PC,CLBUF     ;CLEAR BUFFER AREA
      MOV    Q#10,TEMP1  ;SAVE TIMEOUT TRAP VECTOR CONTENTS
      MOV    Q#12,TEMP2
      MOV    #T1CONT,Q#10 ;POINT TIMEOUT VECTOR TO TEST CONTINUATION
      MOV    #340,Q#12
      JSR    R5,XPSW     ;STORE EXPECTED PSW VALUE
      .WORD  217
      JSR    PC,GENR    ;SET UP GENERAL REGISTERS
      SCC
T1:   ADDN      ;EXECUTE "ADDN" INSTRUCTION
      ERROR 1      ;*****TEST 1 - ERROR 1*****
                        ;DECIMAL INSTRUCTION DID NOT TRAP ON
                        ;A ZERO SOURCE1 LENGTH
      BR ENDT1
T1CONT:
      CMP    R0,S1LN    ;CHECK R0 UNCHANGED
      BEQ    64$
      ERROR 2      ;*****TEST 1 - ERROR 2*****
                        ;R0 CHANGED
                        ;R0 SHOULD STILL EQUAL CONTENTS OF "S1LN"
      64$:  CMP    R1,S1ADR ;CHECK R1 UNCHANGED
      BEQ    65$
      ERROR 3      ;*****TEST 1 - ERROR 3*****
                        ;R1 CHANGED
                        ;R1 SHOULD STILL EQUAL CONTENTS OF "S1ADR"
      65$:  CMP    R2,S2LN    ;CHECK R2 UNCHANGED
      BEQ    66$
      ERROR 4      ;*****TEST 1 - ERROR 4*****
                        ;R2 CHANGED
                        ;R2 SHOULD STILL EQUAL THE CONTENTS OF "S2LN"
      66$:  CMP    R3,S2ADR ;CHECK R3 UNCHANGED
      BEQ    67$
      ERROR 5      ;*****TEST 1 - ERROR 5*****
                        ;R3 CHANGED
                        ;R3 SHOULD STILL EQUAL THE CONTENTS OF "S2ADR"
      67$:  CMP    R4,FILL    ;CHECK R4 UNCHANGED
      BEQ    68$
      ERROR 6      ;*****TEST 1 - ERROR 6*****
                        ;R4 CHANGED
                        ;R4 SHOULD STILL EQUAL THE CONTENTS OF "FILL"
      68$:  CMP    R5,DSTAD

```

J02

797	001456	001401				BEQ	69\$		
798	001460	104007				ERROR	7		;*****TEST 1 - ERROR 7*****
799									;R5 CHANGED
800									;R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"
801	001462			69\$:					
802	001462	016600	000002			MOV	2(SP),R0		;CHECK PSW AT TIME OF TRAP
803	001466	042700	177400			BIC	#177400,R0		
804	001472	020067	177202			CMP	R0,EXPPSW		
805	001476	001401				BEQ	1\$		
806	001500	104010				ERROR	10		;*****TEST 1 - ERROR 10*****
807									;PSW ERROR
808									;EXPECTED PSW IS STORED AT "EXPPSW"
809									;RESULTANT PSW IS IN R0
810	001502	021627	001376	1\$:		CMP	(SP),#T1+2		;CHECK ADDRESS OF TRAP
811	001506	001403				BEQ	2\$		
812	001510	011637	000522			MOV	(SP),@#\$BADDR		;STORE BAD ADDRESS
813	001514	104011				ERROR	11		;*****TEST 1 - ERROR 11*****
814									;TRAP ADDRESS ERROR
815									;EXPECTED ADDRESS IS "T1+2"
816									;RESULTANT ADDRESS IS STORED AT "\$BADDR"
817	001516	010600		2\$:		MOV	SP,R0		;CHECK SP RESTORATION
818	001520	062700	000004			ADD	#4,R0		
819	001524	020037	000702			CMP	R0,@\$SAVR6		
820	001530	001401				BEQ	3\$		
821	001532	104012				ERROR	12		;*****TEST 1 - ERROR 12*****
822									;STACK POINTER WAS NOT RESTORED
823									;EXPECTED SP VALUE IS STORED AT "SAVR6"
824									;ERRONEOUS SO VALUE IS AT "BADR6"
825	001534	012716	001552	3\$:		MOV	#ENDT1,(SP)		;RESTORE SP & PSW
826	001540	013766	000700	000002		MOV	@EXPPSW,2(SP)		
827	001546	000002				RTI			
828									
829	001550			S1T1:					;SOURCE1 STRING
830	001550	060				.BYTE	60		;MOST SIGNIFICANT DIGIT
831	001551			S2T1:					;SOURCE2 STRING
832	001551	060				.BYTE	60		;MOST SIGNIFICANT DIGIT
833						.EVEN			
834									
835	001552	016737	177134	000010	ENDT1:	MOV	TEMP1,@#10		;RESTORE TIMEOUT VECTOR CONTENTS
836	001560	016737	177130	000012		MOV	TEMP2,@#12		

```

837
838
839
840      ;*****
841      ;*TEST 2      TEST "ADDN" WITH A POSITIVE LEADING SIGN SOURCE1, & VALID SOURCE2
842      ;*****
843      †ST2:  SCOPE
844      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
845      1          ;SOURCE1 LENGTH
846      S1T2     ;SOURCE1 ADDRESS
847      1          ;SOURCE2 LENGTH
848      S2T2     ;SOURCE2 ADDRESS
849      0          ;DESTINATION LENGTH
850      JSR      PC,CLBUF     ;CLEAR BUFFER AREA
851      MOV      @#10,TEMP1   ;SAVE TIMEOUT TRAP VECTOR CONTENTS
852      MOV      @#12,TEMP2
853      MOV      #T2CONT,@#10 ;POINT TIMEOUT VECTOR TO TEST CONTINUATION
854      MOV      #340,@#12
855      JSR      R5,XPSW     ;STORE EXPECTED PSW VALUE
856      .WORD    217
857      JSR      PC,GENR     ;SET UP GENERAL REGISTERS
858      SCC
859      T2:      ADDN        ;EXECUTE "ADDN" INSTRUCTION
860
861      ERROR    1          ;*****TEST 2 - ERROR 1*****
862      ;DECIMAL INSTRUCTION DID NOT TRAP ON
863      ;A POSITIVE LEADING SIGN SOURCE1
864      BR      ENDT2
865
866      T2CONT:
867      CMP      R0,S1LN     ;CHECK R0 UNCHANGED
868      BEQ      64$
869      ERROR    2          ;*****TEST 2 - ERROR 2*****
870      ;R0 CHANGED
871      ;R0 SHOULD STILL EQUAL CONTENTS OF "S1LN"
872      64$:    CMP      R1,S1ADR ;CHECK R1 UNCHANGED
873      BEQ      65$
874      ERROR    3          ;*****TEST 2 - ERROR 3*****
875      ;R1 CHANGED
876      ;R1 SHOULD STILL EQUAL CONTENTS OF "S1ADR"
877      65$:    CMP      R2,S2LN ;CHECK R2 UNCHANGED
878      BEQ      66$
879      ERROR    4          ;*****TEST 2 - ERROR 4*****
880      ;R2 CHANGED
881      ;R2 SHOULD STILL EQUAL THE CONTENTS OF "S2LN"
882      66$:    CMP      R3,S2ADR ;CHECK R3 UNCHANGED
883      BEQ      67$
884      ERROR    5          ;*****TEST 2 - ERROR 5*****
885      ;R3 CHANGED
886      ;R3 SHOULD STILL EQUAL THE CONTENTS OF "S2ADR"
887      67$:    CMP      R4,FILL ;CHECK R4 UNCHANGED
888      BEQ      68$
889      ERROR    6          ;*****TEST 2 - ERROR 6*****
890      ;R4 CHANGED
891      ;R4 SHOULD STILL EQUAL THE CONTENTS OF "FILL"
892      68$:    CMP      R5,DSTAD ;CHECK R5 UNCHANGED

```

L02

```

893 001740 001401      BEQ      69$
894 001742 104007      ERROR    7      ;*****TEST 2 - ERROR 7*****
895                                     ;R5 CHANGED
896                                     ;R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"
897 001744                                     69$:
898 001744 016600 000002      MOV      2(SP),R0      ;CHECK PSW AT TIME OF TRAP
899 001750 042700 177400      BIC     #177400,R0
900 001754 020067 176720      CMP     R0,EXPPSW
901 001760 001401      BEQ     1$
902 001762 104010      ERROR   10      ;*****TEST 2 - ERROR 10*****
903                                     ;PSW ERROR
904                                     ;EXPECTED PSW IS STORED AT "EXPPSW"
905                                     ;RESULTANT PSW IS IN R0
906 001764 021627 001650      1$:      CMP     (SP),#T2+2      ;CHECK ADDRESS OF TRAP
907 001770 001403      BEQ     2$
908 001772 011637 000522      MOV     (SP),@#$BDADR      ;STORE BAD ADDRESS
909 001776 104011      ERROR   11      ;*****TEST 2 - ERROR 11*****
910                                     ;TRAP ADDRESS ERROR
911                                     ;EXPECTED ADDRESS IS "T2+2"
912                                     ;RESULTANT ADDRESS IS STORED AT "$BDADR"
913 002000 010600                                     2$:      MOV     SP,R0
914 002002 062700 000004      ADD     #4,R0
915 002006 020037 000702      CMP     R0,@$SAVR6
916 002012 001401      BEQ     3$
917 002014 104012      ERROR   12      ;*****TEST 2 - ERROR 12*****
918                                     ;STACK POINTER WAS NOT RESTORED
919                                     ;EXPECTED SP VALUE IS STORED AT "SAVR6"
920                                     ;ERRONEOUS SO VALUE IS AT "BADR6"
921 002016 012716 002034 000002 3$:      MOV     #ENDT2,(SP)      ;RESTORE SP & PSW
922 002022 013766 000700      MOV     @EXPPSW,2(SP)
923 002030 000012      RTI
924
925 002032                                     S1T2:
926 002032      055      .BYTE   55      ;SOURCE1 STRING
927 002033                                     S2T2:
928 002033      060      .BYTE   60      ;MOST SIGNIFICANT DIGIT
929                                     .EVEN
930
931 002034 016737 176652 000010      ENDT2:  MOV     TEMP1,@#10      ;RESTORE TIMEOUT VECTOR CONTENTS
932 002042 016737 176646 000012      MOV     TEMP2,@#12

```

# M02

```

933
934
935
936
937
938 002050 000004
939 002052 004567 012654
940 002056 000001
941 002060 002314
942 002062 000001
943 002064 002315
944 002066 000000
945 002070 004767 012730
946 002074 013767 000010 176610
947 002102 013767 000012 176604
948 002110 012737 002146 000010
949 002116 012737 000340 000012
950 002124 004567 012714
951 002130 000217
952 002132 004767 012622
953 002136 000277
954
955 002140 076050
956
957 002142 104001
958
959
960 002144 000464
961
962 002146
963 002146 020067 176472
964 002152 001401
965 002154 104002
966
967
968 002156 020107 176464
969 002162 001401
970 002164 104003
971
972
973 002166 020267 176456
974 002172 001401
975 002174 104004
976
977
978 002176 020367 176450
979 002202 001401
980 002204 104005
981
982
983 002206 020467 176446
984 002212 001401
985 002214 104006
986
987
988 002216 020567 176434

```

```

*****
*TEST 3 TEST "ADDN" WITH A NEGATIVE LEADING SIGN SOURCE1, & VALID SOURCE2
*****
T3: SCOPE
      JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
          1 ;SOURCE1 LENGTH
          S1T3 ;SOURCE1 ADDRESS
          1 ;SOURCE2 LENGTH
          S2T3 ;SOURCE2 ADDRESS
          0 ;DESTINATION LENGTH
      JSR PC,CLBUF ;CLEAR BUFFER AREA
      MOV Q#10,TEMP1 ;SAVE TIMEOUT TRAP VECTOR CONTENTS
      MOV Q#12,TEMP2
      MOV #T3CONT,Q#10 ;POINT TIMEOUT VECTOR TO TEST CONTINUATION
      MOV #340,Q#12
      JSR R5,XPSW ;STORE EXPECTED PSW VALUE
          .WORD 217
      JSR PC,GENR ;SET UP GENERAL REGISTERS
      SCC
T3: ADDN ;EXECUTE "ADDN" INSTRUCTION
      ERROR 1 ;*****TEST 3 - ERROR 1*****
          ;DECIMAL INSTRUCTION DID NOT TRAP ON
          ;A NEGATIVE LEADING SIGN SOURCE2
      BR ENDT3
T3CONT:
      CMP R0,S1LN ;CHECK R0 UNCHANGED
      BEQ 64$
      ERROR 2 ;*****TEST 3 - ERROR 2*****
          ;R0 CHANGED
          ;R0 SHOULD STILL EQUAL CONTENTS OF "S1LN"
      CMP R1,S1ADR ;CHECK R1 UNCHANGED
      BEQ 65$
      ERROR 3 ;*****TEST 3 - ERROR 3*****
          ;R1 CHANGED
          ;R1 SHOULD STILL EQUAL CONTENTS OF "S1ADR"
      CMP R2,S2LN ;CHECK R2 UNCHANGED
      BEQ 66$
      ERROR 4 ;*****TEST 3 - ERROR 4*****
          ;R2 CHANGED
          ;R2 SHOULD STILL EQUAL THE CONTENTS OF "S2LN"
      CMP R3,S2ADR ;CHECK R3 UNCHANGED
      BEQ 67$
      ERROR 5 ;*****TEST 3 - ERROR 5*****
          ;R3 CHANGED
          ;R3 SHOULD STILL EQUAL THE CONTENTS OF "S2ADR"
      CMP R4,FILL ;CHECK R4 UNCHANGED
      BEQ 68$
      ERROR 6 ;*****TEST 3 - ERROR 6*****
          ;R4 CHANGED
          ;R4 SHOULD STILL EQUAL THE CONTENTS OF "FILL"
      CMP R5,DSTAD ;CHECK R5 UNCHANGED

```

# NO2

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 26  
 DVKAJA.P11 20-DEC-76 15:02 T3

TEST "ADDN" WITH A NEGATIVE LEADING SIGN SOURCE1, & VALID SOURCE2

SEQ 0028

989	002222	001401			BEQ	69\$		
990	002224	104007			ERROR	7		*****TEST 3 - ERROR 7*****
991								;RS CHANGED
992								;RS SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"
993	002226			69\$:				
994	002226	016600	000002		MOV	2(SP),R0		;CHECK PSW AT TIME OF TRAP
995	002232	042700	177400		BIC	#177400,R0		
996	002236	020067	176436		CMP	R0,EXPPSW		
997	002242	001401			BEQ	1\$		
998	002244	104010			ERROR	10		*****TEST 3 - ERROR 10*****
999								;PSW ERROR
1000								;EXPECTED PSW IS STORED AT "EXPPSW"
1001								;RESULTANT PSW IS IN R0
1002	002246	021627	002142	1\$:	CMP	(SP),#T3+2		;CHECK ADDRESS OF TRAP
1003	002252	001403			BEQ	2\$		
1004	002254	011637	000522		MOV	(SP),@#\$BDADR		;STORE BAD ADDRESS
1005	002260	104011			ERROR	11		*****TEST 3 - ERROR 11*****
1006								;TRAP ADDRESS ERROR
1007								;EXPECTED ADDRESS IS "T3+2"
1008								;RESULTANT ADDRESS IS STORED AT "\$BDADR"
1009	002262	010600		2\$:	MOV	SP,R0		;CHECK SP RESTORATION
1010	002264	062700	000004		ADD	#4,R0		
1011	002270	020037	000702		CMP	R0,@\$SAVR6		
1012	002274	001401			BEQ	3\$		
1013	002276	104012			ERROR	12		*****TEST 3 - ERROR 12*****
1014								;STACK POINTER WAS NOT RESTORED
1015								;EXPECTED SP VALUE IS STORED AT "SAVR5"
1016								;ERRONEOUS SO VALUE IS AT "BADR6"
1017	002300	012716	002316	3\$:	MOV	#ENDT3,(SP)		;RESTORE SP & PSW
1018	002304	013766	000700	000002	MOV	@#EXPPSW,2(SP)		
1019	002312	000002			RTI			
1020								
1021	002314			S1T3:				;SOURCE1 STRING
1022	002314	053			.BYTE	53		;MOST SIGNIFICANT DIGIT
1023	002315			S2T3:				;SOURCE2 STRING
1024	002315	060			.BYTE	60		;MOST SIGNIFICANT DIGIT
1025					.EVEN			
1026								
1027	002316	016737	176370	000010	ENDT3: MOV	TEMP1,@#10		;RESTORE TIMEOUT VECTOR CONTENTS
1028	002324	016737	176364	000012	MOV	TEMP2,@#12		

1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084

002332 000004  
002334 004567 012372  
002340 000001  
002342 002576  
002344 000000  
002346 002577  
002350 000000  
002352 004767 012446  
002356 013767 000010 176326  
002364 013767 000012 176322  
002372 012737 002430 000010  
002400 012737 000340 000012  
002406 004567 012432  
002412 000217  
002414 004767 012340  
002420 000277  
  
002422 076050  
002424 104001  
002426 000464  
  
002430  
002430 020067 176210  
002434 001401  
002436 104002  
  
002440 020167 176202  
002444 001401  
002446 104003  
  
002450 020267 176174  
002454 001401  
002456 104004  
  
002460 020367 176166  
002464 001401  
002466 104005  
  
002470 020467 176164  
002474 001401  
002476 104006  
  
002500 020567 176152

```
*****  
:TEST 4 TEST "ADDN" WITH SOURCE2 LENGTH = 0, & VALID SOURCE1  
*****  
TST4: SCOPE  
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST  
1 ;SOURCE1 LENGTH  
S1T4 ;SOURCE1 ADDRESS  
0 ;SOURCE2 LENGTH  
S2T4 ;SOURCE2 ADDRESS  
0 ;DESTINATION LENGTH  
JSR PC,CLBUF ;CLEAR BUFFER AREA  
MOV @#10,TEMP1 ;SAVE TIMEOUT TRAP VECTOR CONTENTS  
MOV @#12,TEMP2  
MOV #T4CONT,@#10 ;POINT TIMEOUT VECTOR TO TEST CONTINUATION  
MOV #340,@#12  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 217  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC  
  
T4: ADDN ;EXECUTE "ADDN" INSTRUCTION  
  
ERROR 1 ;*****TEST 4 - ERROR 1*****  
;DECIMAL INSTRUCTION DID NOT TRAP ON  
;A ZERO SOURCE2 LENGTH  
  
BR ENDT4  
  
T4CONT:  
CMP R0,S1LN ;CHECK R0 UNCHANGED  
BEQ 64$  
ERROR 2 ;*****TEST 4 - ERROR 2*****  
;R0 CHANGED  
;R0 SHOULD STILL EQUAL CONTENTS OF "S1LN"  
;CHECK R1 UNCHANGED  
64$: CMP R1,S1ADR  
BEQ 65$  
ERROR 3 ;*****TEST 4 - ERROR 3*****  
;R1 CHANGED  
;R1 SHOULD STILL EQUAL CONTENTS OF "S1ADR"  
;CHECK R2 UNCHANGED  
65$: CMP R2,S2LN  
BEQ 66$  
ERROR 4 ;*****TEST 4 - ERROR 4*****  
;R2 CHANGED  
;R2 SHOULD STILL EQUAL THE CONTENTS OF "S2LN"  
;CHECK R3 UNCHANGED  
66$: CMP R3,S2ADR  
BEQ 67$  
ERROR 5 ;*****TEST 4 - ERROR 5*****  
;R3 CHANGED  
;R3 SHOULD STILL EQUAL THE CONTENTS OF "S2ADR"  
;CHECK R4 UNCHANGED  
67$: CMP R4,FILL  
BEQ 68$  
ERROR 6 ;*****TEST 4 - ERROR 6*****  
;R4 CHANGED  
;R4 SHOULD STILL EQUAL THE CONTENTS OF "FILL"  
;CHECK R5 UNCHANGED  
68$: CMP R5,DSTAD
```

1085	002504	001401			BEQ	69\$		
1086	002506	134007			ERROR	7		;*****TEST 4 - ERROR 7*****
1087								;R5 CHANGED
1088								;R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"
1089	002510			69\$:				
1090	002510	016600	000002		MOV	2(SP),R0		;CHECK PSW AT TIME OF TRAP
1091	002514	042700	177400		BIC	#177400,R0		
1092	002520	020067	176154		CMP	R0,EXPPSW		
1093	002524	001401			BEQ	1\$		
1094	002526	104010			ERROR	10		;*****TEST 4 - ERROR 10*****
1095								;PSW ERROR
1096								;EXPECTED PSW IS STORED AT "EXPPSW"
1097								;RESULTANT PSW IS IN R0
1098	002530	021627	002424	1\$:	CMP	(SP),#T4+2		;CHECK ADDRESS OF TRAP
1099	002534	001403			BEQ	2\$		
1100	002536	011637	000522		MOV	(SP),2#\$BDADR		;STORE BAD ADDRESS
1101	002542	104011			ERROR	11		;*****TEST 4 - ERROR 11*****
1102								;TRAP ADDRESS ERROR
1103								;EXPECTED ADDRESS IS "T4+2"
1104								;RESULTANT ADDRESS IS STORED AT "\$BDADR"
1105	002544	010600		2\$:	MOV	SP,R0		;CHECK SP RESTORATION
1106	002546	062700	000004		ADD	#4,R0		
1107	002552	020037	000702		CMP	R0,2\$SAVR6		
1108	002556	001401			BEQ	3\$		
1109	002560	104012			ERROR	12		;*****TEST 4 - ERROR 12*****
1110								;STACK POINTER WAS NOT RESTORED
1111								;EXPECTED SP VALUE IS STORED AT "SAVR6"
1112								;ERRONEOUS SO VALUE IS AT "BADR6"
1113	002562	012716	002600	3\$:	MOV	#ENDT4,(SP)		;RESTORE SP & PSW
1114	002566	013766	000700	000002	MOV	2\$EXPPSW,2(SP)		
1115	002574	000002			RTI			
1116								
1117	002576			S1T4:				;SOURCE1 STRING
1118	002576	060			.BYTE	60		;MOST SIGNIFICANT DIGIT
1119	002577			S2T4:				;SOURCE2 STRING
1120	002577	060			.BYTE	60		;MOST SIGNIFICANT DIGIT
1121					.EVEN			
1122								
1123	002600	016737	176106	000010	ENDT4: MOV	TEMP1,2\$10		;RESTORE TIMEOUT VECTOR CONTENTS
1124	002606	016737	176102	000012	MOV	TEMP2,2\$12		

```

1125
1126
1127
1128      ;*****
1129      ;*TEST 5      TEST "ADDN" BY ADDING TWO ZEROES WITH ZERO DESTINATION LENGTH
1130      ;*****
1130 002614 000004      TSTS:  SCOPE
1131 002616 004567 012110      JSR      RS,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
1132 002622 000001      JSR      1      ;SOURCE1 LENGTH
1133 002624 002772      JSR      S1T5      ;SOURCE1 ADDRESS
1134 002626 000001      JSR      1      ;SOURCE2 LENGTH
1135 002630 002773      JSR      S2T5      ;SOURCE2 ADDRESS
1136 002632 000000      JSR      0      ;DESTINATION LENGTH
1137 002634 004767 012164      JSR      PC,CLBUF      ;CLEAR BUFFER AREA
1138 002640 004567 012200      JSR      RS,XPSW
1139 002644 000204      .WORD 204
1140 002646 004767 012106      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
1141 002652 000277      SCC      ;SET UP THE COMPLEMENT OF EXPECTED CC'S
1142 002654 000244
1143 002656 076050      CLZ
1144      ADDN
1145 002660 106767 176012      MFPS     CCODES      ;STORE RESULTANT PSW
1146 002664 042767 177400 176004      BIC      #177400,CCODES ;CLEAR UNUSED BITS
1147 002672 023767 000700 175776      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
1148 002700 001401      BEQ      64$
1149 002702 104001      ERROR   1      ;*****TEST 5 - ERROR 1*****
1150      PSW ERROR
1151      EXPECTED PSW IS STORED AT "SAVR6"
1152      ERRONEOUS SP VALUE IS AT "BADR6"
1153 002704      64$:
1154 002704 005700      TST      R0      ;CHECK R0=0
1155 002706 001401      BEQ      65$
1156 002710 104002      ERROR   2      ;*****TEST 5 - ERROR 2*****
1157      ;R0 SHOULD BE ZERO
1158 002712 005701      65$:  TST      R1      ;CHECK R1=0
1159 002714 001401      BEQ      66$
1160 002716 104003      ERROR   3      ;*****TEST 5 - ERROR 3*****
1161      ;R1 SHOULD BE ZERO
1162 002720 005702      66$:  TST      R2      ;CHECK R2=0
1163 002722 001401      BEQ      67$
1164 002724 104004      ERROR   4      ;*****TEST 5 - ERROR 4*****
1165      ;R2 SHOULD BE ZERO
1166 002726 005703      67$:  TST      R3      ;CHECK R3=0
1167 002730 001401      BEQ      68$
1168 002732 104005      ERROR   5      ;*****TEST 5 - ERROR 5*****
1169      ;R3 SHOULD BE ZERO
1170 002734 020467 175714      68$:  CMP      R4,DSTLN      ;CHECK R4= DESTINATION LENGTH
1171 002740 001401      BEQ      69$
1172 002742 104006      ERROR   6      ;*****TEST 5 - ERROR 6*****
1173      ;R4 SHOULD STILL BE DESTINATION LENGTH
1174 002744 020567 175706      69$:  CMP      R5,DSTAD      ;CHECK R5 = DESTINATION ADDRESS
1175 002750 001401      BEQ      70$
1176 002752 104007      ERROR   7      ;*****TEST 5 - ERROR 7*****
1177      ;R5 SHOULD STILL BE DESTINATION ADDRESS
1178 002754 023706 000702      70$:  CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
1179 002760 001403      BEQ      71$
1180 002762 010637 000704      MOV      SP,@#BADR6      ;COPY BAD SP VALUE

```

E03

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 30  
DVKAJA.P11 20-DEC-76 15:02 TS

TEST "ADDN" BY ADDING TWO ZEROES WITH ZERO DESTINATION LENGTH

SEQ 0032

1181 002766 104010  
1182  
1183  
1184  
1195 002770  
1186 002770 000401  
1187 002772  
1188 002772 060  
1189 002773  
1190 002773 060  
1191  
1192

ERROR 10  
  
715: BR TST6  
S1T5: .BYTE 60  
S2T5: .BYTE 60  
.EVEN

:\*\*\*\*\*TEST 5 - ERROR 10\*\*\*\*\*  
:STACK POINTER NOT RESTORED BY INSTRUCTION  
:EXPECTED SP IS STORED AT "SAVR6"  
:ERRONEOUS SP VALUE IS AT "BADR6"  
  
:BR TO NEXT TEST  
:SOURCE1 STRING  
:MOST SIGNIFICANT DIGIT  
:SOURCE2 STRING  
:MOST SIGNIFICANT DIGIT

1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248

002774 000004  
002776 004567 011730  
003002 000004  
003004 003176  
003006 000002  
003010 003202  
003012 000002  
003014 004767 012004  
003020 004567 012020  
003024 000200  
003026 004767 011726  
003032 000277  
003034 076050  
  
003036 106767 175634  
003042 042767 177400 175626  
003050 023767 500700 175620  
003056 001401  
003060 104001  
  
003062  
003062 005700  
003064 001401  
003066 104002  
  
003070 005701  
003072 001401  
003074 104003  
  
003076 005702  
003100 001401  
003102 104004  
  
003104 005703  
003106 001401  
003110 104005  
  
003112 020467 175536  
003116 001401  
003120 104006  
  
003122 020567 175530  
003126 001401  
003130 104007  
  
003132 023706 000702  
003136 001403  
003140 010637 000704

```
::*****  
:*TEST 6 TEST "ADDN" WITH POSITIVE OPERANDS, SRC1 LENGTH .GT. SRC2 LENGTH, DL = NO  
:*****  
↑ST6: SCOPE  
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST  
4 ;SOURCE1 LENGTH  
S1T6 ;SOURCE1 ADDRESS  
2 ;SOURCE2 LENGTH  
S2T6 ;SOURCE2 ADDRESS  
2 ;DESTINATION LENGTH  
PC,CLBUF ;CLEAR BUFFER AREA  
JSR R5,XPSW  
.WORD 200  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S  
ADDN  
  
MFPS CCODES ;STORE RESULTANT PSW  
BIC #177400,CCODES ;CLEAR UNUSED BITS  
CMP @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALJE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 6 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "SAVR6"  
;ERRONEOUS SP VALJE IS AT "BADR6"  
  
64$: TST R0 ;CHECK R0=0  
BEQ 65$  
ERROR 2 ;*****TEST 6 - ERROR 2*****  
;R0 SHOULD BE ZERO  
65$: TST R1 ;CHECK R1=0  
BEQ 66$  
ERROR 3 ;*****TEST 6 - ERROR 3*****  
;R1 SHOULD BE ZERO  
66$: TST R2 ;CHECK R2=0  
BEQ 67$  
ERROR 4 ;*****TEST 6 - ERROR 4*****  
;R2 SHOULD BE ZERO  
67$: TST R3 ;CHECK R3=0  
BEQ 68$  
ERROR 5 ;*****TEST 6 - ERROR 5*****  
;R3 SHOULD BE ZERO  
68$: CMP R4,DSTLN ;CHECK R4= DESTINATION LENGTH  
BEQ 69$  
ERROR 6 ;*****TEST 6 - ERROR 6*****  
;R4 SHOULD STILL BE DESTINATION LENGTH  
69$: CMP R5,DSTAD ;CHECK R5 = DESTINATION ADDRESS  
BEQ 70$  
ERROR 7 ;*****TEST 6 - ERROR 7*****  
;R5 SHOULD STILL BE DESTINATION ADDRESS  
70$: CMP @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 71$ ;BR IF OK  
MOV SP,@BADR6 ;COPY BAD SP VALJE
```

G03

1249	003144	104310		ERROR	10					*****TEST 6 - ERROR 10*****
1250										;STACK POINTER NOT RESTORED BY INSTRUCTION
1251										;EXPECTED SP IS STORED AT "SAVR6"
1252										;ERRONEOUS SP VALUE IS AT "BADRE"
1253	003146			71\$:						
1254										;CHECK ANSWER
1255	003146	012700	003204	MOV	#ANS6,R0					;POINT R0 TO EXPECTED ANSWER
1256	003152	016701	175500	MOV	DSTAD,R1					;POINT R1 TO RESULTANT ANSWER
1257	003156	016702	175472	MOV	DSTLN,R2					;STORE ANSWER LENGTH IN R1
1258	003162	122021		72\$:	CMPB	(R0)+,(R1)+				;COMPARE EACH DIGIT
1259	003164	001401		BEQ	73\$					;BR IF EQUAL
1260	003166	104311		ERROR	11					*****TEST 6 - ERROR 11*****
1261										;ERRONEOUS ANSWER
1262										;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1263										;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1264	003170	005302		73\$:	DEC	R2				;DECREMENT ANSWER LENGTH
1265	003172	001373		BNE	72\$					;BR IF NOT FINISHED
1266	003174	000404		BR	TST7					;BR TO NEXT TEST
1267	003176			S1T6:						;SOURCE1 STRING
1268	003176	060			.BYTE	60				;MOST SIGNIFICANT DIGIT
1269	003177	060			.BYTE	60				
1270	003200	062			.BYTE	62				
1271	003201	061			.BYTE	61				
1272	003202			S2T6:						;SOURCE2 STRING
1273	003202	067			.BYTE	67				;MOST SIGNIFICANT DIGIT
1274	003203	070			.BYTE	70				
1275	003204			ANS6:						;EXPECTED ANSWER
1276	003204	071			.BYTE	71				
1277	003205	071			.BYTE	71				
1278										
1279					.EVEN					

H03

```

1280
1281
1282
1283
1284
1285 003206 000004
1286 003210 004567 011516
1287 003214 000002
1288 003216 003412
1289 003220 000004
1290 003222 003414
1291 003224 000004
1292 003226 004767 011572
1293 003232 004567 011606
1294 003236 000210
1295 003240 004767 011514
1296 003244 000277
1297 003246 000250
1298 003250 076050
1299
1300 003252 106767 175420
1301 003256 042767 177400 175412
1302 003264 023767 000700 175404
1303 003272 001401
1304 003274 104001
1305
1306
1307
1308 003276
1309 003276 005700
1310 003300 001401
1311 003302 104002
1312
1313 003304 005701
1314 003306 001401
1315 003310 104003
1316
1317 003312 005702
1318 003314 001401
1319 003316 104004
1320
1321 003320 005703
1322 003322 001401
1323 003324 104005
1324
1325 003326 020467 175322
1326 003332 001401
1327 003334 104006
1328
1329 003336 020567 175314
1330 003342 001401
1331 003344 104007
1332
1333 003346 023706 000702
1334 003352 001403
1335 003354 010637 000704

```

```

*****
*TEST 7 TEST "ADDN" WITH NEGATIVE OPERANDS, S2L .GT. S1L, NO OVERFLOW
*****
TST7: SCOPE
      JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
          2 ;SOURCE1 LENGTH
          S1T7 ;SOURCE1 ADDRESS
          4 ;SOURCE2 LENGTH
          S2T7 ;SOURCE2 ADDRESS
          4 ;DESTINATION LENGTH
      JSR PC,CLBUF ;CLEAR BUFFER AREA
      JSR R5,XPSW
      .WORD 210
      JSR PC,GENR ;SET UP GENERAL REGISTERS
      SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CLN
      ADDN
MFP5 CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 7 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALJE IS AT "BADR6"
64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 7 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$:
TST R1
BEQ 66$
ERROR 3 ;*****TEST 7 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$:
TST R2
BEQ 67$
ERROR 4 ;*****TEST 7 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$:
TST R3
BEQ 68$
ERROR 5 ;*****TEST 7 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4= DESTINATION LENGTH
68$:
CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 7 - ERROR 6*****
;R4 SHOULD STILL BE DESTINATION LENGTH
;CHECK R5 = DESTINATION ADDRESS
69$:
CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 7 - ERROR 7*****
;R5 SHOULD STILL BE DESTINATION ADDRESS
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
MOV SP,#BADR6 ;COPY BAD SP VALJE

```

I03

1336	003360	104010		ERROR	10		*****TEST 7 - ERROR 10*****
1337							;STACK POINTER NOT RESTORED BY INSTRUCTION
1338							;EXPECTED SP IS STORED AT "SAVR6"
1339							;ERRONEOUS SP VALUE IS AT "BADR6"
1340	003362			71\$:			
1341							;CHECK ANSWER
1342	003362	012700	003420	MOV	#ANS7,R0		;POINT R0 TO EXPECTED ANSWER
1343	003366	016701	175264	MOV	DSTAD,R1		;POINT R1 TO RESULTANT ANSWER
1344	003372	016702	175256	MOV	DSTLN,R2		;STORE ANSWER LENGTH IN R1
1345	003376	122021		72\$:	CMPB	(R0)+,(R1)+	;COMPARE EACH DIGIT
1346	003400	001401			SEQ	73\$	;BR IF EQUAL
1347	003402	104011		ERROR	11		*****TEST 7 - ERROR 11*****
1348							;ERRONEOUS ANSWER
1349							;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1350							;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1351	003404	005302		73\$:	DEC	R2	;DECREMENT ANSWER LENGTH
1352	003406	001373			BNE	72\$	;BR IF NOT FINISHED
1353	003410	000405			BR	TST10	;BR TO NEXT TEST
1354	003412			S1T7:			;SOURCE1 STRING
1355	003412	063			.BYTE	63	;MOST SIGNIFICANT DIGIT
1356	003413	165			.BYTE	165	
1357	003414			S2T7:			;SOURCE2 STRING
1358	003414	360			.BYTE	60	;MOST SIGNIFICANT DIGIT
1359	003415	071			.BYTE	71	
1360	003416	066			.BYTE	66	
1361	003417	165			.BYTE	165	
1362	003420			ANS7:			;EXPECTED ANSWER
1363	003420	061			.BYTE	61	;MOST SIGNIFICANT DIGIT
1364	003421	060			.BYTE	60	
1365	003422	060			.BYTE	60	
1366	003423	160			.BYTE	160	
1367							
1368					.EVEN		

```

1369
1370
1371
1372
1373
1374 003424 000004
1375 003426 004567 011300
1376 003432 000004
1377 003434 00363C
1378 003436 000002
1379 003440 003634
1380 003442 000002
1381 003444 004767 011354
1382 003450 004567 01137C
1383 003454 000212
1384 003456 004767 011276
1385 003462 000265
1386 003464 000252
1387 003466 076050
1388
1389 003470 106767 175202
1390 003474 042767 17740C 175174
1391 003502 023767 000700 175166
1392 003510 001401
1393 003512 104001
1394
1395
1396
1397 003514
1398 003514 005700
1399 003516 001401
1400 003520 104002
1401
1402 003522 005701
1403 003524 001401
1404 003526 104003
1405
1406 003530 005702
1407 003532 001401
1408 003534 104004
1409
1410 003536 005703
1411 003540 001401
1412 003542 104005
1413
1414 003544 020467 175104
1415 003550 001401
1416 003552 104006
1417
1418 003554 020567 175076
1419 003560 001401
1420 003562 104007
1421
1422 003564 023706 000702
1423 003570 001403
1424 003572 010637 000704

```

```

*****
*TEST 10 TEST "ADDN" WITH NEGATIVE OPERANDS, OVERFLOW
*****
TST10: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      4        ;SOURCE1 LENGTH
      S1T10    ;SOURCE1 ADDRESS
      2        ;SOURCE2 LENGTH
      S2T10    ;SOURCE2 ADDRESS
      2        ;DESTINATION LENGTH
      JSR      PC,CLBUF     ;CLEAR BUFFER AREA
      JSR      R5,XPSW
      .WORD    212
      JSR      PC,GENR     ;SET UP GENERAL REGISTERS
      +SEZ!SEC             ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      +CLN!CLV
      ADDN

MFPS   CCODES      ;STORE RESULTANT PSW
BIC    #177400,CCODES ;CLEAR UNUSED BITS
CMP    @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ    64$
ERROR  1
*****TEST 10 - ERROR 1*****
PSW ERROR
EXPECTED PSW IS STORED AT "SAVR6"
ERRONEOUS SP VALUE IS AT "BADR6"

64$:   TST      R0          ;CHECK R0=0
      BEQ      65$
      ERROR    2
*****TEST 10 - ERROR 2*****
R0 SHOULD BE ZERO
CHECK R1=0

65$:   TST      R1
      BEQ      66$
      ERROR    3
*****TEST 10 - ERROR 3*****
R1 SHOULD BE ZERO
CHECK R2=0

66$:   TST      R2
      BEQ      67$
      ERROR    4
*****TEST 10 - ERROR 4*****
R2 SHOULD BE ZERO
CHECK R3=0

67$:   TST      R3
      BEQ      68$
      ERROR    5
*****TEST 10 - ERROR 5*****
R3 SHOULD BE ZERO
CHECK R4= DESTINATION LENGTH

68$:   CMP      R4,DSTLN
      BEQ      69$
      ERROR    6
*****TEST 10 - ERROR 6*****
R4 SHOULD STILL BE DESTINATION LENGTH
CHECK R5 = DESTINATION ADDRESS

69$:   CMP      R5,DSTAD
      BEQ      70$
      ERROR    7
*****TEST 10 - ERROR 7*****
R5 SHOULD STILL BE DESTINATION ADDRESS
VERIFY STACK POINTER IS RESTORED
BR IF OK
COPY BAD SP VALUE

```

# K03

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 36  
 DVKAJA.P11 20-DEC-76 15:02 T10

TEST "ADDN" WITH NEGATIVE OPERANDS, OVERFLOW

SEQ 0039

```

1425 003576 104010          ERROR 10          ;*****TEST 10 - EPOR 10*****
1426                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
1427                                     ;EXPECTED SP IS STORED AT "SAVR6"
1428                                     ;ERRONEOUS SP VALUE IS AT "BADRE"
1429 003600          71$:
1430                                     ;CHECK ANSWER
1431 003600 012700 003636      MOV      #ANS10,R0      ;POINT R0 TO EXPECTED ANSWER
1432 003604 016701 175046      MOV      DSTAD,R1      ;POINT R1 TO RESULTANT ANSWER
1433 003610 016702 175040      MOV      DSTLN,R2      ;STORE ANSWER LENGTH IN R1
1434 003614 122021          72$:      CMPB     (R0)+,(R1)+    ;COMPARE EACH DIGIT
1435 003616 001401          BEQ      73$          ;BR IF EQUAL
1436 003620 104011          ERROR 11          ;*****TEST 10 - ERROR 1:*****
1437                                     ;ERRONEOUS ANSWER
1438                                     ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1439                                     ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1440 003622 005302          73$:      DEC      R2          ;DECREMENT ANSWER LENGTH
1441 003624 001373          BNE      72$          ;BR IF NOT FINISHED
1442 003626 000404          BR       TST11        ;BR TO NEXT TEST
1443 003630          S1T10:          ;SOURCE1 STRING
1444 003630          .BYTE 60          ;MOST SIGNIFICANT DIGIT
1445 003631          .BYTE 61
1446 003632          .BYTE 62
1447 003633          .BYTE 163
1448 003634          S2T10:          ;SOURCE2 STRING
1449 003634          .BYTE 65          ;MOST SIGNIFICANT DIGIT
1450 003635          .BYTE 164
1451 003636          ANS10:          ;EXPECTED ANSWER
1452 003636          .BYTE 67          ;MOST SIGNIFICANT DIGIT
1453 003637          .BYTE 167
1454
1455          .EVEN
  
```

L03

```

1456
1457
1458
1459
1460
1461 003640 000004
1462 003642 004567 011064
1463 003646 000002
1464 003650 004044
1465 003652 000002
1466 003654 004046
1467 003656 000002
1468 003660 004767 011140
1469 003664 004567 011154
1470 003670 000206
1471 003672 004767 011062
1472 003676 000271
1473 003700 000246
1474 003702 076050
1475
1476 003704 106767 174766
1477 003710 042767 177400 174760
1478 003716 023767 000700 174752
1479 003724 001401
1480 003726 104001
1481
1482
1483
1484 003730
1485 003730 005700
1486 003732 001401
1487 003734 104002
1488
1489 003736 005701
1490 003740 001401
1491 003742 104003
1492
1493 003744 005702
1494 003746 001401
1495 003750 104004
1496
1497 003752 005703
1498 003754 001401
1499 003756 104005
1500
1501 003760 020467 174670
1502 003764 001401
1503 003766 104006
1504
1505 003770 020567 174662
1506 003774 001401
1507 003776 104007
1508
1509 004000 023706 000702
1510 004004 001403
1511 004006 010637 000704

```

```

;*****
; *TEST 11 TEST "ADDN" WITH POSITIVE OPERANDS, S2L=S1L, OVERFLOW
;*****
†ST11: SCOPE
      JSR      RS,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      2
      S1T11      ;SOURCE1 LENGTH
      2
      S2T11      ;SOURCE1 ADDRESS
      2
      PC,CLBUF   ;SOURCE2 LENGTH
      JSR      RS,XPSW      ;SOURCE2 ADDRESS
      2
      .WORD 206 ;DESTINATION LENGTH
      JSR      PC,GENR      ;CLEAR BUFFER AREA
      +SEN!SEC
      +CLZ!CLV
      ADDN
      MFPS      CCODES      ;STORE RESULTANT PSW
      BIC      #177400,CCODES ;CLEAR UNUSED BITS
      CMP      @#EXPPSW.CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$
      ERROR    1
      ;*****TEST 11 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED AT "SAVR6"
      ;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
      TST      R0
      BEQ      65$
      ERROR    2
      ;*****TEST 11 - ERROR 2*****
      ;R0 SHOULD BE ZERO
      ;CHECK R1=0
65$:
      TST      R1
      BEQ      66$
      ERROR    3
      ;*****TEST 11 - ERROR 3*****
      ;R1 SHOULD BE ZERO
      ;CHECK R2=0
66$:
      TST      R2
      BEQ      67$
      ERROR    4
      ;*****TEST 11 - ERROR 4*****
      ;R2 SHOULD BE ZERO
      ;CHECK R3=0
67$:
      TST      R3
      BEQ      68$
      ERROR    5
      ;*****TEST 11 - ERROR 5*****
      ;R3 SHOULD BE ZERO
      ;CHECK R4= DESTINATION LENGTH
68$:
      CMP      R4,DSTLN
      BEQ      69$
      ERROR    6
      ;*****TEST 11 - ERROR 6*****
      ;R4 SHOULD STILL BE DESTINATION LENGTH
      ;CHECK R5 = DESTINATION ADDRESS
69$:
      CMP      R5,DSTAD
      BEQ      70$
      ERROR    7
      ;*****TEST 11 - ERROR 7*****
      ;R5 SHOULD STILL BE DESTINATION ADDRESS
      ;VERIFY STACK POINTER IS RESTORED
      ;BR IF OK
      MOV      SP,@#BADR6 ;COPY BAD SP VALUE

```

# M03

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 38  
 CVKAJA.P11 20-DEC-76 15:02 T11

TEST "ADDN" WITH POSITIVE OPERANDS, S2L=S1L, OVERFLOW

SEQ 0040

1512	004012	104010		ERROR	10		*****TEST 11 - ERROR 10*****
1513							:STACK POINTER NOT RESTORED BY INSTRUCTION
1514							:EXPECTED SP IS STORED AT "SAVR6"
1515							:ERRONEOUS SP VALUE IS AT "BADR6"
1516	004014		71\$:				:CHECK ANSWER
1517							:POINT R0 TO EXPECTED ANSWER
1518	004014	012700		MOV	*ANS11,R0		:POINT R1 TO RESULTANT ANSWER
1519	004020	016701	004050	MOV	DSTAD,R1		:STORE ANSWER LENGTH IN R1
1520	004024	016702	174624	MOV	DSTLN,R2		:COMPARE EACH DIGIT
1521	004030	122021		72\$:	CMPB	(R0)+,(R1)+	:BR IF EQUAL
1522	004032	001401			SEQ	73\$	*****TEST 11 - ERROR 11*****
1523	004034	104011			ERROR	11	:ERRONEOUS ANSWER
1524							:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1525							:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1526							:DECREMENT ANSWER LENGTH
1527	004036	005302	73\$:	DEC	R2		:BR IF NOT FINISHED
1528	004040	001373		BNE	72\$		:BR TO NEXT TEST
1529	004042	000403		BR	TST12		:SOURCE1 STRING
1530	004044		S1T11:				:MOST SIGNIFICANT DIGIT
1531	004044	063		.BYTE	63		
1532	004045	062		.BYTE	62		
1533	004046		S2T11:				:SOURCE2 STRING
1534	004046	056		.BYTE	66		:MOST SIGNIFICANT DIGIT
1535	004047	07C		.BYTE	70		
1536	004050		ANS11:				:EXPECTED ANSWER
1537	004050	06C		.BYTE	60		
1538	004051	06C		.BYTE	6C		
1539							
1540				.EVEN			

# NO3

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 39  
 DVKAJA.P11 20-DEC-76 15:02 T11

TEST "ADDN" WITH POSITIVE OPERANDS, S2L=S1L,OVERFLOW

SEQ 0041

```

1541
1542
1543
1544
1545
1546 004052 000004
1547 004054 004567 010652
1548 004060 000002
1549 004062 004256
1550 004064 000002
1551 004066 004260
1552 004070 000001
1553 004072 004767 010726
1554 004076 004567 010742
1555 004102 000212
1556 004104 004767 010650
1557 004110 000265
1558 004112 000252
1559 004114 076050
1560
1561 004116 106767 174554 MFPS CCODES ;STORE RESULTANT PSW
1562 004122 042767 177400 174546 BIC #177400,CCODES ;CLEAR UNUSED BITS
1563 004130 023767 000700 174540 CMP @#EXPPSW.CCODES ;CHECK PSW AGAINST EXPECTED VALUE
1564 004136 001401 BEQ 64$ ;BR, IF EQUAL
1565 004140 104001 ERROR 1 ;*****TEST 12 - ERROR 1*****
1566 ;PSW ERROR
1567 ;EXPECTED PSW IS STORED AT "SAVR6"
1568 ;ERRONEOUS SP VALUE IS AT "BADR6"
1569
1570 004142 005700 64$: TST R0 ;CHECK R0=0
1571 004144 001401 BEQ 65$
1572 004146 104002 ERROR 2 ;*****TEST 12 - ERROR 2*****
1573 ;R0 SHOULD BE ZERO
1574 004150 005701 65$: TST R1 ;CHECK R1=0
1575 004152 001401 BEQ 66$
1576 004154 104003 ERROR 3 ;*****TEST 12 - ERROR 3*****
1577 ;R1 SHOULD BE ZERO
1578 004156 005702 66$: TST R2 ;CHECK R2=0
1579 004160 001401 BEQ 67$
1580 004162 104004 ERROR 4 ;*****TEST 12 - ERROR 4*****
1581 ;R2 SHOULD BE ZERO
1582 004164 005703 67$: TST R3 ;CHECK R3=0
1583 004166 001401 BEQ 68$
1584 004170 104005 ERROR 5 ;*****TEST 12 - ERROR 5*****
1585 ;R3 SHOULD BE ZERO
1586 004172 020467 174456 68$: CMP R4,DSTLN ;CHECK R4= DESTINATION LENGTH
1587 004176 001401 BEQ 69$
1588 004200 104006 ERROR 6 ;*****TEST 12 - ERROR 6*****
1589 ;R4 SHOULD STILL BE DESTINATION LENGTH
1590 004202 020567 174450 69$: CMP R5,DSTAD ;CHECK R5 = DESTINATION ADDRESS
1591 004206 001401 BEQ 70$
1592 004210 104007 ERROR 7 ;*****TEST 12 - ERROR 7*****
1593 ;R5 SHOULD STILL BE DESTINATION ADDRESS
1594 004212 023706 000702 70$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
1595 004216 001403 BEQ 71$ ;BR IF OK
1596 004220 010637 000704 MOV SP,@#BADR6 ;COPY BAD SP VALUE
  
```

```

1597 004224 104010          ERROR 10          ;*****TEST 12 - ERROR 10*****
1598                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
1599                                     ;EXPECTED SP IS STORED AT "SAVR6"
1600                                     ;ERRONEOUS SP VALUE IS AT "BADR6"
1601 004226          71$:
1602                                     ;CHECK ANSWER
1603 004226 012700 004262      MOV #ANS12,R0      ;POINT R0 TO EXPECTED ANSWER
1604 004232 016701 174420      MOV DSTAD,R1      ;POINT R1 TO RESULTANT ANSWER
1605 004236 016702 174412      MOV DSTLN,R2      ;STORE ANSWER LENGTH IN R1
1606 004242 122021          72$: CMPB (R0)+,(R1)+  ;COMPARE EACH DIGIT
1607 004244 001401          BEQ 73$          ;BR IF EQUAL
1608 004246 104011          ERROR 11          ;*****TEST 12 - ERROR 11*****
1609                                     ;ERRONEOUS ANSWER
1610                                     ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1611                                     ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1612 004250 005302          73$: DEC R2          ;DECREMENT ANSWER LENGTH
1613 004252 001373          BNE 72$          ;BR IF NOT FINISHED
1614 004254 000403          BR TST13         ;BR TO NEXT TEST
1615 004256          S1T12: ;SOURCE1 STRING
1616 004256          .BYTE 66      ;MOST SIGNIFICANT DIGIT
1617 004257          .BYTE 161
1618 004260          S2T12: ;SOURCE2 STRING
1619 004260          .BYTE 64      ;MOST SIGNIFICANT DIGIT
1620 004261          .BYTE 165
1621 004262          ANS12: ;EXPECTED ANSWER
1622 004262          .BYTE 166
1623
1624          004264          .EVEN

```

```

1625
1626
1627
1628
1629
1630 004264 000004
1631 004266 004567 010440
1632 004272 000001
1633 004274 004442
1634 004276 000001
1635 004300 004443
1636 004302 000000
1637 004304 004767 010514
1638 004310 004567 01053C
1639 004314 000204
1640 004316 004767 010436
1641 004322 000277
1642 004324 000244
1643 004326 076050
1644
1645 004330 106767 174342
1646 004334 042767 177400 174334
1647 004342 023767 000700 174326
1648 004350 001401
1649 004352 104001
1650
1651
1652
1653 004354
1654 004354 005700
1655 004356 001401
1656 004360 104002
1657
1658 004362 005701
1659 004364 001401
1660 004366 104003
1661
1662 004370 005702
1663 004372 001401
1664 004374 104004
1665
1666 004376 005703
1667 004400 001401
1668 004402 104005
1669
1670 004404 020467 174244
1671 004410 001401
1672 004412 104006
1673
1674 004414 020567 174236
1675 004420 001401
1676 004422 104007
1677
1678 004424 023706 000702
1679 004430 001403
1680 004432 010637 000704

```

```

;*****
;*TEST 13 TEST "ADDN" WITH +SRC1 & -SRC2, ZERO RESULT
;*****
†ST13: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
1 ;SOURCE1 LENGTH
S1T13 ;SOURCE1 ADDRESS
1 ;SOURCE2 LENGTH
S2T13 ;SOURCE2 ADDRESS
0 ;DESTINATION LENGTH
JSR PC,CLBUF ;CLEAR BUFFER AREA
JSR R5,XPSW
.WORD 204
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CLZ
ADDN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 13 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 13 - ERROR 2*****
;R0 SHOULD BE ZERO
65$:
TST R1 ;CHECK R1=0
BEQ 66$
ERROR 3 ;*****TEST 13 - ERROR 3*****
;R1 SHOULD BE ZERO
66$:
TST R2 ;CHECK R2=0
BEQ 67$
ERROR 4 ;*****TEST 13 - ERROR 4*****
;R2 SHOULD BE ZERO
67$:
TST R3 ;CHECK R3=0
BEQ 68$
ERROR 5 ;*****TEST 13 - ERROR 5*****
;R3 SHOULD BE ZERO
68$:
CMP R4,DSTLN ;CHECK R4= DESTINATION LENGTH
BEQ 69$
ERROR 6 ;*****TEST 13 - ERROR 6*****
;R4 SHOULD STILL BE DESTINATION LENGTH
69$:
CMP R5,DSTAD ;CHECK R5 = DESTINATION ADDRESS
BEQ 70$
ERROR 7 ;*****TEST 13 - ERROR 7*****
;R5 SHOULD STILL BE DESTINATION ADDRESS
70$:
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 71$
MOV SP,#BADR6 ;COPY BAD SP VALUE

```

1681 004436 104010  
1682  
1683  
1684  
1685 004440  
1686 004440 000401  
1687 004442  
1688 004442 063  
1689 004443  
1690 004443 163  
1691  
1692

ERROR 10  
  
715:  
S1T13: BR TST14  
.BYTE 63  
S2T13:  
.BYTE 163  
.EVEN

\*\*\*\*\*TEST 13 - ERROR 10\*\*\*\*\*  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED SP IS STORED AT "SAVR6"  
;ERRONEOUS SP VALUE IS AT "BADR6"  
  
:BR TO NEXT TEST  
:SOURCE1 STRING  
:MOST SIGNIFICANT DIGIT  
:SOURCE2 STRING  
:MOST SIGNIFICANT DIGIT

# E04

```

1693
1694
1695
1696
1697
1698 004444 000004
1699 004446 004567 010260
1700 004452 000003
1701 004454 004650
1702 004456 000004
1703 004460 004653
1704 004462 000002
1705 004464 004767 010334
1706 004470 004567 010350
1707 004474 000210
1708 004476 004767 010256
1709 004502 000277
1710 004504 000250
1711 004506 076050
1712
1713 004510 106767 174162
1714 004514 042767 177400 174154
1715 004522 023767 300700 174146
1716 004530 001401
1717 004532 104001
1718
1719
1720
1721 004534
1722 004534 005700
1723 004536 001401
1724 004540 104002
1725
1726 004542 005701
1727 004544 001401
1728 004546 104003
1729
1730 004550 005702
1731 004552 001401
1732 004554 104004
1733
1734 004556 005703
1735 004560 001401
1736 004562 104005
1737
1738 004564 020467 174064
1739 004570 001401
1740 004572 104006
1741
1742 004574 020567 174056
1743 004600 001401
1744 004602 104007
1745
1746 004604 023706 000702
1747 004610 001403
1748 004612 010637 000704

```

```

;*****
;*TEST 14 TEST "ADDN" WITH -SRC1 & +SRC2, S1L .LT. S2L, /S2/ .GT. /S1/
;*****
↑ST14: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      3
      S1T14      ;SOURCE1 LENGTH
      4
      S2T14      ;SOURCE1 ADDRESS
      2
      PC,CLBUF      ;SOURCE2 LENGTH
      JSR      R5,XPSW      ;SOURCE2 ADDRESS
      .WORD 210      ;DESTINATION LENGTH
      JSR      PC,GENR      ;CLEAR BUFFER AREA
      SCC
      CLN
      ADDN
;*****
MFP5  C00DES      ;STORE RESULTANT PSW
BIC   #177400,C00DES ;CLEAR UNUSED BITS
CMP   @#EXPPSW,C00DES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ   64$
ERROR 1
;*****TEST 14 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALJE IS AT "BADR6"
54$:
      TST      R0
      BEQ      65$
      ERROR    2
;*****TEST 14 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$:
      TST      R1
      BEQ      66$
      ERROR    3
;*****TEST 14 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$:
      TST      R2
      BEQ      67$
      ERROR    4
;*****TEST 14 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$:
      TST      R3
      BEQ      68$
      ERROR    5
;*****TEST 14 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4= DESTINATION LENGTH
68$:
      CMP      R4,DSTLN
      BEQ      69$
      ERROR    6
;*****TEST 14 - ERROR 6*****
;R4 SHOULD STILL BE DESTINATION LENGTH
;CHECK R5 = DESTINATION ADDRESS
69$:
      CMP      R5,DSTAD
      BEQ      70$
      ERROR    7
;*****TEST 14 - ERROR 7*****
;R5 SHOULD STILL BE DESTINATION ADDRESS
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
;COPY BAD SP VALJE
70$:
      CMP      @#SAVR6,SP
      BEQ      71$
      MOV      SP,@#BADR6

```

F04

1749	004616	104010		ERROR	10		:*****TEST 14 - ERROR 10*****
1750							:STACK POINTER NOT RESTORED BY INSTRUCTION
1751							:EXPECTED SP IS STORED AT "SAVR6"
1752							:ERRONEOUS SP VALUE IS AT "BADRE"
1753	004620		71\$:				
1754							:CHECK ANSWER
1755	004620	012700		MOV	#ANS14,R0		:POINT R0 TO EXPECTED ANSWER
1756	004624	016701		MOV	DSTAD,R1		:POINT R1 TO RESULTANT ANSWER
1757	004630	016702		MOV	DSTLN,R2		:STORE ANSWER LENGTH IN R1
1758	004634	122021	72\$:	CMPB	(R0)+,(R1)+		:COMPARE EACH DIGIT
1759	004636	001401		SEQ	73\$		:BR IF EQUAL
1760	004640	104011		ERROR	11		:*****TEST 14 - ERROR 11*****
1761							:ERRONEOUS ANSWER
1762							:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1763							:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1764	004642	005302	73\$:	DEC	R2		:DECREMENT ANSWER LENGTH
1765	004644	001373		BNE	72\$		:BR IF NOT FINISHED
1766	004646	000405		BR	TST15		:BR TO NEXT TEST
1767	004650		S1T14:				:SOURCE1 STRING
1768	004650	067		.BYTE	67		:MOST SIGNIFICANT DIGIT
1769	004651	066		.BYTE	66		
1770	004652	064		.BYTE	64		
1771	004653		S2T14:				:SOURCE2 STRING
1772	004653	060		.BYTE	60		:MOST SIGNIFICANT DIGIT
1773	004654	070		.BYTE	70		
1774	004655	063		.BYTE	63		
1775	004656	171		.BYTE	171		
1776	004657		ANS14:				:EXPECTED ANSWER
1777	004657	067		.BYTE	67		:MOST SIGNIFICANT DIGIT
1778	004660	165		.BYTE	165		
1779							
1780							
1781							
1782							
1783							
1784							
1785							
1786							
1787							
1788							
1789							
1790							
1791							
1792							
1793							
1794							
1795							
1796							
1797							
1798							
1799							
1800							
1801							
1802							
1803							
1804							
1805							
1806							
1807							
1808							
1809							
1810							
1811							
1812							
1813							
1814							
1815							
1816							
1817							
1818							
1819							
1820							
1821							
1822							
1823							
1824							
1825							
1826							
1827							
1828							
1829							
1830							
1831							
1832							
1833							
1834							
1835							
1836							
1837							
1838							
1839							
1840							
1841							
1842							
1843							
1844							
1845							
1846							
1847							
1848							
1849							
1850							
1851							
1852							
1853							
1854							
1855							
1856							
1857							
1858							
1859							
1860							
1861							
1862							
1863							
1864							
1865							
1866							
1867							
1868							
1869							
1870							
1871							
1872							
1873							
1874							
1875							
1876							
1877							
1878							
1879							
1880							
1881							
1882							
1883							
1884							
1885							
1886							
1887							
1888							
1889							
1890							
1891							
1892							
1893							
1894							
1895							
1896							
1897							
1898							
1899							
1900							
1901							
1902							
1903							
1904							
1905							
1906							
1907							
1908							
1909							
1910							
1911							
1912							
1913							
1914							
1915							
1916							
1917							
1918							
1919							
1920							
1921							
1922							
1923							
1924							
1925							
1926							
1927							
1928							
1929							
1930							
1931							
1932							
1933							
1934							
1935							
1936							
1937							
1938							
1939							
1940							
1941							
1942							
1943							
1944							
1945							
1946							
1947							
1948							
1949							
1950							
1951							
1952							
1953							
1954							
1955							
1956							
1957							
1958							
1959							
1960							
1961							
1962							
1963							
1964							
1965							
1966							
1967							
1968							
1969							
1970							
1971							

G04

```

1781
1782
1783
1784
1785
1786 004662 000004
1787 004664 004567 010042
1788 004670 000004
1789 004672 005066
1790 004674 000003
1791 004676 005072
1792 004700 000002
1793 004702 004767 010116
1794 004706 004567 010132
1795 004712 000206
1796 004714 004767 010043
1797 004720 000271
1798 004722 000246
1799 004724 076050
1800
1801 004726 106767 173744
1802 004732 042767 177400 173736
1803 004740 023767 000700 173730
1804 004746 001401
1805 004750 104001
1806
1807
1808
1809 004752
1810 004752 005700
1811 004754 001401
1812 004756 104002
1813
1814 004760 005701
1815 004762 001401
1816 004764 104003
1817
1818 004766 005702
1819 004770 001401
1820 004772 104004
1821
1822 004774 005703
1823 004776 001401
1824 005000 104005
1825
1826 005002 020467 173646
1827 005006 001401
1828 005010 104006
1829
1830 005012 020567 173640
1831 005016 001401
1832 005020 104007
1833
1834 005022 023706 000702
1835 005026 001403
1836 005030 010637 000704

```

```

*****
:TEST 15 TEST "ADDN" WITH +SRC1 & -SRC2, S1L .GT. S2L, /S2/ .GT. /S1/. OVERFLOW
*****
TST15: SCOPE
        JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
        4
        ;SOURCE1 LENGTH
        S1T15
        ;SOURCE1 ADDRESS
        3
        ;SOURCE2 LENGTH
        S2T15
        ;SOURCE2 ADDRESS
        2
        ;DESTINATION LENGTH
        ;CLEAR BUFFER AREA
        JSR      PC,CLBUF
        JSR      R5,XPSW
        .WORD    206
        JSR      PC,GENR      ;SET UP GENERAL REGISTERS
        +SEN!SEC
        ;SET UP THE COMPLEMENT OF EXPECTED CC'S
        +CLZ!CLV
        ADDN

        MFPS     CCODES      ;STORE RESULTANT PSW
        BIC      #177400,CCODES ;CLEAR UNUSED BITS
        CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$
        ;R0, IF EQUAL
        ;*****TEST 15 - ERROR 1*****
        ;PSW ERROR
        ;EXPECTED PSW IS STORED AT "SAVR6"
        ;ERRONEOUS SP VALUE IS AT "BADR6"
        64$:
        TST      R0
        ;CHECK R0=0
        BEQ      65$
        ERROR    2
        ;*****TEST 15 - ERROR 2*****
        ;R0 SHOULD BE ZERO
        ;CHECK R1=0
        65$:
        TST      R1
        BEQ      66$
        ERROR    3
        ;*****TEST 15 - ERROR 3*****
        ;R1 SHOULD BE ZERO
        ;CHECK R2=0
        66$:
        TST      R2
        BEQ      67$
        ERROR    4
        ;*****TEST 15 - ERROR 4*****
        ;R2 SHOULD BE ZERO
        ;CHECK R3=0
        67$:
        TST      R3
        BEQ      68$
        ERROR    5
        ;*****TEST 15 - ERROR 5*****
        ;R3 SHOULD BE ZERO
        ;CHECK R4= DESTINATION LENGTH
        68$:
        CMP      R4,DSTLN
        BEQ      69$
        ERROR    6
        ;*****TEST 15 - ERROR 6*****
        ;R4 SHOULD STILL BE DESTINATION LENGTH
        ;CHECK R5 = DESTINATION ADDRESS
        69$:
        CMP      R5,DSTAD
        BEQ      70$
        ERROR    7
        ;*****TEST 15 - ERROR 7*****
        ;R5 SHOULD STILL BE DESTINATION ADDRESS
        ;VERIFY STACK POINTER IS RESTORED
        ;BR IF OK
        ;COPY BAD SP VALUE
        MOV      SP,@#BADR6

```

# H04

1837	005034	104010		ERROR	10		:*****TEST 15 - ERROR 10*****
1838							:STACK POINTER NOT RESTORED BY INSTRUCTION
1839							:EXPECTED SP IS STORED AT "SAVR6"
1840							:ERRONEOUS SP VALUE IS AT "BADR6"
1841	005036		71\$:				:CHECK ANSWER
1842							:POINT R0 TO EXPECTED ANSWER
1843	005036	012700		MOV	#ANS15,R0		:POINT R1 TO RESULTANT ANSWER
1844	005042	016701		MOV	DSTAD,R1		:STORE ANSWER LENGTH IN R1
1845	005046	016702		MOV	DSTLN,R2		:COMPARE EACH DIGIT
1846	005052	122021	72\$:	CMPB	(R0)+,(R1)+		:BR IF EQUAL
1847	005054	001401		BEQ	73\$		:*****TEST 15 - ERROR 11*****
1848	005056	104011		ERROR	11		:ERRONEOUS ANSWER
1849							:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1850							:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1851							:DECREMENT ANSWER LENGTH
1852	005060	005302	73\$:	DEC	R2		:BR IF NOT FINISHED
1853	005062	001373		BNE	72\$		:BR TO NEXT TEST
1854	005064	000405		BR	TST16		:SOURCE1 STRING
1855	005066		S1T15:				:MOST SIGNIFICANT DIGIT
1856	005066	060		.BYTE	60		
1857	005067	060		.BYTE	60		
1858	005070	067		.BYTE	67		
1859	005071	163		.BYTE	163		
1860	005072		S2T15:				:SOURCE2 STRING
1861	005072	061		.BYTE	61		:MOST SIGNIFICANT DIGIT
1862	005073	067		.BYTE	67		
1863	005074	063		.BYTE	63		
1864	005075		ANS15:				:EXPECTED ANSWER
1865	005075	060		.BYTE	60		:MOST SIGNIFICANT DIGIT
1866	005076	060		.BYTE	60		
1867							
1868		005100		.EVEN			
1869							

1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925

005100 000004  
005102 004567 007624  
005106 000003  
005110 005304  
005112 000004  
005114 005307  
005116 000002  
005120 004767 007700  
005124 004567 007714  
005130 000206  
005132 004767 007622  
005136 000271  
005140 000246  
005142 076050  
  
005144 106767 173526  
005150 042767 177400 173520  
005156 023767 000700 173512  
005164 001401  
005166 104001  
  
005170  
005170 005700  
005172 001401  
005174 104002  
  
005176 005701  
005200 001401  
005202 104003  
  
005204 005702  
005206 001401  
005210 104004  
  
005212 005703  
005214 001401  
005216 104005  
  
005220 020467 173430  
005224 001401  
005226 104006  
  
005230 020567 173422  
005234 001401  
005236 104007  
  
005240 023706 000702  
005244 001403  
005246 010637 000704

```
*****
*TEST 16 TEST ADDN WITH -SRC1 & +SRC2,S1L .LT. S2L, /S1/ .GT. /S2/,OVERFLOW
*****
↑ST16: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      3
      ;SOURCE1 LENGTH
      S1T16      ;SOURCE1 ADDRESS
      4
      ;SOURCE2 LENGTH
      S2T16      ;SOURCE2 ADDRESS
      2
      ;DESTINATION LENGTH
      ;CLEAR BUFFER AREA
      JSR      PC,CLBUF
      JSR      R5,XPSW
      .WORD    206
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      +SEN!SEC      ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      +CLZ!CLV
      ADDN

      MFPS     CCODES      ;STORE RESULTANT PSW
      BIC     #177400,CCODES ;CLEAR UNUSED BITS
      CMP     @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ     64$
      ERROR   1
      ;*****TEST 16 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED AT "SAVR6"
      ;ERRONEOUS SP VALJE IS AT "BADR6"

64$:   TST     R0
      BEQ     55$
      ERROR   2
      ;*****TEST 16 - ERROR 2*****
      ;R0 SHOULD BE ZERO
      ;CHECK R1=0

55$:   TST     R1
      BEQ     66$
      ERROR   3
      ;*****TEST 16 - ERROR 3*****
      ;R1 SHOULD BE ZERO
      ;CHECK R2=0

66$:   TST     R2
      BEQ     67$
      ERROR   4
      ;*****TEST 16 - ERROR 4*****
      ;R2 SHOULD BE ZERO
      ;CHECK R3=0

67$:   TST     R3
      BEQ     68$
      ERROR   5
      ;*****TEST 16 - ERROR 5*****
      ;R3 SHOULD BE ZERO
      ;CHECK R4= DESTINATION LENGTH

68$:   CMP     R4,DSTLN
      BEQ     69$
      ERROR   6
      ;*****TEST 16 - ERROR 6*****
      ;R4 SHOULD STILL BE DESTINATION LENGTH
      ;CHECK R5 = DESTINATION ADDRESS

69$:   CMP     R5,DSTAD
      BEQ     70$
      ERROR   7
      ;*****TEST 16 - ERROR 7*****
      ;R5 SHOULD STILL BE DESTINATION ADDRESS
      ;VERIFY STACK POINTER IS RESTORED
      ;BR IF OK
      MCV     SP,@#BADR6 ;COPY BAD SP VALUE
```

1926	005252	104010		ERROR	10		:*****TEST 16 - ERROR 10*****
1927							:STACK POINTER NOT RESTORED BY INSTRUCTION
1928							:EXPECTED SP IS STORED AT "SAVRE"
1929							:ERRONEOUS SP VALLE IS AT "BADR6"
1930	005254			71\$:			
1931							:CHECK ANSWER
1932	005254	012700	005313	MOV	#ANS16,R0		:POINT R0 TO EXPECTED ANSWER
1933	005260	016701	173372	MOV	DSTAD,R1		:POINT R1 TO RESULTANT ANSWER
1934	005264	016702	173364	MOV	DSTLN,R2		:STORE ANSWER LENGTH IN R1
1935	005270	122021		72\$:	CMPB	(R0)+,(R1)+	:COMPARE EACH DIGIT
1936	005272	001401		BEQ	73\$		:BR IF EQUAL
1937	005274	104011		ERROR	11		:*****TEST 16 - ERROR 1:*****
1938							:ERRONEOUS ANSWER
1939							:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1940							:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1941	005276	005302		73\$:	DEC	R2	:DECREMENT ANSWER LENGTH
1942	005300	001373		BNE	72\$		:BR IF NOT FINISHED
1943	005302	000405		BR	TST17		:BR TO NEXT TEST
1944	005304			S1T16:			:SOURCE1 STRING
1945	005304	061		.BYTE	61		:MOST SIGNIFICANT DIGIT
1946	005305	067		.BYTE	67		
1947	005306	162		.BYTE	162		
1948	005307			S2T16:			:SOURCE2 STRING
1949	005307	060		.BYTE	60		:MOST SIGNIFICANT DIGIT
1950	005310	060		.BYTE	60		
1951	005311	067		.BYTE	67		
1952	005312	062		.BYTE	62		
1953	005313			ANS16:			:EXPECTED ANSWER
1954	005313	060		.BYTE	60		:MOST SIGNIFICANT DIGIT
1955	005314	060		.BYTE	60		
1956							
1957		005316		.EVEN			

K04

```

1958
1959
1960
1961
1962
1963 005316 000004
1964 005320 004567 007406
1965 005324 000003
1966 005326 005522
1967 005330 000003
1968 005332 005525
1969 005334 000002
1970 005336 004767 007462
1971 005342 004567 007476
1972 005346 000202
1973 005350 004767 007404
1974 005354 000277
1975 005356 000242
1976 005360 076050
1977
1978 005362 106767 173310
1979 005366 042767 177400 173302
1980 005374 023767 000700 173274
1981 005402 001401
1982 005404 104001
1983
1984
1985
1986 005406
1987 005406 005700
1988 005410 001401
1989 005412 104002
1990
1991 005414 005701
1992 005416 001401
1993 005420 104003
1994
1995 005422 005702
1996 005424 001401
1997 005426 104004
1998
1999 005430 005703
2000 005432 001401
2001 005434 104005
2002
2003 005436 020467 173212
2004 005442 001401
2005 005444 104006
2006
2007 005446 020567 173204
2008 005452 001401
2009 005454 104007
2010
2011 005456 023706 000702
2012 005462 001403
2013 005464 010637 000704

```

```

*****
*TEST 17 TEST ADDN WITH +SRC1 & -SRC2, S1L=S2L, /S1/ .GT./S2/,OVERFLOW
*****
ST17: SCOPE
JSR RS,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3 ;SOURCE1 LENGTH
S1T17 ;SOURCE1 ADDRESS
3 ;SOURCE2 LENGTH
S2T17 ;SOURCE2 ADDRESS
2 ;DESTINATION LENGTH
PC,CLBUF ;CLEAR BUFFER AREA
JSR RS,XPSW
JSR R5,XPSW
.WORD 202
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
C_V
ADDN

MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 17 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"

64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 17 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0

65$:
TST R1
BEQ 66$
ERROR 3 ;*****TEST 17 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0

66$:
TST R2
BEQ 67$
ERROR 4 ;*****TEST 17 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0

67$:
TST R3
BEQ 68$
ERROR 5 ;*****TEST 17 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4= DESTINATION LENGTH

68$:
CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 17 - ERROR 6*****
;R4 SHOULD STILL BE DESTINATION LENGTH
;CHECK R5 = DESTINATION ADDRESS

69$:
CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 17 - ERROR 7*****
;R5 SHOULD STILL BE DESTINATION ADDRESS
;VERIFY STACK POINTER IS RESTORED
;BR IF OK

70$:
CMP #SAVR6,SP
BEQ 71$
MOV SP,#BADR6 ;COPY BAD SP VALUE

```

```

2014 005470 104010 ERROR 10 ;*****TEST 17 - ERROR 10*****
2015 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2016 ;EXPECTED SP IS STORED AT "SAVRS"
2017 ;ERRONEOUS SP VALUE IS AT "BADRE"
2018 005472 71$: ;CHECK ANSWER
2019 ;POINT R0 TO EXPECTED ANSWER
2020 005472 012700 005530 MOV #ANS17,R0 ;POINT R1 TO RESULTANT ANSWER
2021 005476 016701 173154 MOV DSTAD,R1 ;STORE ANSWER LENGTH IN R1
2022 005502 016702 173146 MOV DSTLN,R2 ;COMPARE EACH DIGIT
2023 005506 122021 72$: CMPB (R0)+,(R1)+ ;BR IF EQUAL
2024 005510 001401 BEQ 73$ ;*****TEST 17 - ERROR 1:*****
2025 005512 104011 ERROR 11 ;ERRONEOUS ANSWER
2026 ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2027 ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2028 73$: DEC R2 ;DECREMENT ANSWER LENGTH
2029 005514 005302 BNE 72$ ;BR IF NOT FINISHED
2030 005516 001373 BR TST20 ;BR TO NEXT TEST
2031 005520 000404 S1T17: ;SOURCE1 STRING
2032 005522 ;MOST SIGNIFICANT DIGIT
2033 005522 063 .BYTE 63
2034 005523 065 .BYTE 65
2035 005524 067 .BYTE 67
2036 005525 S2T17: ;SOURCE2 STRING
2037 005525 060 .BYTE 60 ;MOST SIGNIFICANT DIGIT
2038 005526 063 .BYTE 63
2039 005527 165 .BYTE 165
2040 005530 ANS17: ;EXPECTED ANSWER
2041 005530 062 .BYTE 62 ;MOST SIGNIFICANT DIGIT
2042 005531 062 .BYTE 62
2043 .EVEN
2044

```

M04

```

2045
2046
2047
2048
2049
2050 005532 000004
2051 005534 004567 007172
2052 005540 000003
2053 005542 005736
2054 005544 000003
2055 005546 005741
2056 005550 000002
2057 005552 004767 007246
2058 005556 004567 007262
2059 005562 000202
2060 005564 004767 007170
2061 005570 000277
2062 005572 000242
2063 005574 076050
2064
2065 005576 106767 173074
2066 005602 042767 177400 173066
2067 005610 023767 000700 173060
2068 005616 001401
2069 005620 104001
2070
2071
2072
2073 005622
2074 005622 005700
2075 005624 001401
2076 005626 104002
2077
2078 005630 005701
2079 005632 001401
2080 005634 104003
2081
2082 005636 005702
2083 005640 001401
2084 005642 104004
2085
2086 005644 005703
2087 005646 001401
2088 005650 104005
2089
2090 005652 020467 172776
2091 005656 001401
2092 005660 104006
2093
2094 005662 020567 172770
2095 005666 001401
2096 005670 104007
2097
2098 005672 023706 000702
2099 005676 001403
2100 005700 010637 000704

```

```

*****
;TEST 20 TEST ADDN WITH POSITIVE OPERANDS, OVERFLOW, NO CARRY OUT OF OVERFLCW
*****
†ST20: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      3
      S1T20    ;SOURCE1 LENGTH
      3
      S2T20    ;SOURCE1 ADDRESS
      2
      PC,CLBUF ;SOURCE2 LENGTH
      JSR      R5,XPSW      ;SOURCE2 ADDRESS
      .WORD    202          ;DESTINATION LENGTH
      JSR      PC,GENR      ;CLEAR BUFFER AREA
      SCC
      CLV
      ADDN
      MFPS     CCODES      ;STORE RESULTANT PSW
      BIC     #177400,CCODES ;CLEAR UNUSED BITS
      CMP     @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ     64$          ;BR, IF EQUAL
      ERROR   1           ;*****TEST 20 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "SAVR6"
                          ;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
      TST     R0
      BEQ     65$
      ERROR   2           ;*****TEST 20 - ERROR 2*****
                          ;R0 SHOULD BE ZERO
                          ;CHECK R0=0
65$:
      TST     R1
      BEQ     66$
      ERROR   3           ;*****TEST 20 - ERROR 3*****
                          ;R1 SHOULD BE ZERO
                          ;CHECK R1=0
66$:
      TST     R2
      BEQ     67$
      ERROR   4           ;*****TEST 20 - ERROR 4*****
                          ;R2 SHOULD BE ZERO
                          ;CHECK R2=0
67$:
      TST     R3
      BEQ     68$
      ERROR   5           ;*****TEST 20 - ERROR 5*****
                          ;R3 SHOULD BE ZERO
                          ;CHECK R3=0
68$:
      CMP     R4,DSTLN
      BEQ     69$
      ERROR   6           ;*****TEST 20 - ERROR 6*****
                          ;R4 SHOULD STILL BE DESTINATION LENGTH
                          ;CHECK R4= DESTINATION LENGTH
69$:
      CMP     R5,DSTAD
      BEQ     70$
      ERROR   7           ;*****TEST 20 - ERROR 7*****
                          ;R5 SHOULD STILL BE DESTINATION ADDRESS
                          ;CHECK R5 = DESTINATION ADDRESS
70$:
      CMP     @#SAVR6,SP
      BEQ     71$
      MOV     SP,@#BADR6
      ;BR IF OK
      ;COPY BAD SP VALUE

```

# N04

2101	005704	104010		ERROR	10		;*****TEST 20 - ERROR 10*****
2102							;STACK POINTER NOT RESTORED BY INSTRUCTION
2103							;EXPECTED SP IS STORED AT "SAVR6"
2104							;ERRONEOUS SP VALUE IS AT "BADR6"
2105	005706			71\$.			
2106							;CHECK ANSWER
2107	005706	012700	005744			MOV #ANS20,R0	;POINT R0 TO EXPECTED ANSWER
2108	005712	016701	172740			MOV DSTAD,R1	;POINT R1 TO RESULTANT ANSWER
2109	005716	016702	172732			MOV DSTLN,R2	;STORE ANSWER LENGTH IN R1
2110	005722	122021		72\$:		CMPB (R0)+,(R1)+	;COMPARE EACH DIGIT
2111	005724	001401				BEG 73\$	;BR IF EQUAL
2112	005726	104011				ERROR 11	;*****TEST 20 - ERROR 11*****
2113							;ERRONEOUS ANSWER
2114							;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2115							;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2116	005730	005302		73\$:		DEC R2	;DECREMENT ANSWER LENGTH
2117	005732	001373				BNE 72\$	;BR IF NOT FINISHED
2118	005734	000404				BR TST21	;BR TO NEXT TEST
2119	005736			S1T20:			;SOURCE1 STRING
2120	005736	070				.BYTE 70	;MOST SIGNIFICANT DIGIT
2121	005737	062				.BYTE 62	
2122	005740	064				.BYTE 64	
2123	005741			S2T20:			;SOURCE2 STRING
2124	005741	061				.BYTE 61	;MOST SIGNIFICANT DIGIT
2125	005742	070				.BYTE 70	
2126	005743	065				.BYTE 65	
2127	005744			ANS20:			;EXPECTED ANSWER
2128	005744	060				.BYTE 60	;MOST SIGNIFICANT DIGIT
2129	005745	071				.BYTE 71	
2130							
2131						.EVEN	

```

2132
2133
2134
2135
2136
2137 005746 000004
2138 005750 105777 172564
2139 005754 100555
2140 005756 026767 172566 172700
2141 005764 001007
2142 005766 032767 000001 172612
2143 005774 001403
2144 005776 005767 172572
2145 006002 001142
2146 006004
2147 006004 004567 006722
2148 006010 000004
2149 006012 006260
2150 006014 000004
2151 006016 006264
2152 006020 000005
2153 006022 004767 006776
2154 006026 012767 006106 172640
2155 006034 012777 015114 172626
2156 006042 005077 172624
2157 006046 004767 007016
2158 006052 013777 000554 172606
2159 006060 004567 006760
2160 006064 000000
2161 006066 106427 000000
2162 006072 052777 000100 172564
2163 006100 004767 006654
2164 006104 000277
2165 006106 076050
2166
2167 006110 106767 172562
2168 006114 032777 000100 172542
2169 006122 001366
2170 006124 042767 177400 172544
2171 006132 023767 000700 172536
2172 006140 001401
2173 006142 104001
2174
2175
2176
2177 006144
2178 006144 005700
2179 006146 001401
2180 006150 104002
2181
2182 006152 005701
2183 006154 001401
2184 006156 104003
2185
2186 006160 005702
2187 006162 001401

```

```

*****
*TEST 21 TEST INTERRUPTABILITY OF "ADDN"
*****
TST21: SCOPE
TSTB QSWR ;TEST BIT 7 OF SWR
BMI TST22 ;SKIP TO NEXT TEST IF SET
CMP $TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T21CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT #BIT0, $ENV ;IF YES, IS PROGRAM UNDER APT?
BEQ T21CONT ;BR, IF NOT
TST $PASS ;IF YES, CHECK IF NOT ON FIRST PASS
BNE TST22 ;IF NOT, BR & SKIP TEST

T21CONT:
JSR R5, NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T21 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T21 ;SOURCE2 ADDRESS
5 ;DESTINATION LENGTH
JSR PC, CLBUF ;CLEAR BUFFER AREA
MOV #ADDNPC, PCI ;STORE PC OF TEST INSTRUCTION
MOV #INTR, @TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
JSR PC, TDONE ;WAIT FOR SLU READY
MOV @#NULL, @TBUF ;SEND NULL CHARACTER
JSR R5, XPSW ;STORE EXPECTED PSW
.WORD 0
MTPS #0 ;ALLOW INTERRUPTS
BIS #100, @TCSR ;ENABLE TTY INTERRUPTS
READN: JSR PC, GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
ADDNPC: ADDN

MFPS CCODES ;STORE RESULTANT PSW
BIT #100, @TCSR ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
BNE READN ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
BIC #177400, CCODES ;CLEAR UNUSED BITS
CMP @#EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 21 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"

64$: TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 21 - ERROR 2*****
;R0 SHOULD BE ZERO
65$: TST R1 ;CHECK R1=0
BEQ 66$
ERROR 3 ;*****TEST 21 - ERROR 3*****
;R1 SHOULD BE ZERO
66$: TST R2 ;CHECK R2=0
BEQ 67$

```

```

2188 006164 104004          ERROR 4          ;*****TEST 21 - ERROR 4*****
2189                                     ;R2 SHOULD BE ZERO
2190 006166 005703      67$:  TST    R3          ;CHECK R3=0
2191 006170 001401          BEQ    68$
2192 006172 104005          ERROR 5          ;*****TEST 21 - ERROR 5*****
2193                                     ;R3 SHOULD BE ZERO
2194 006174 020467 172454    68$:  CMP    R4,DSTLN    ;CHECK R4= DESTINATION LENGTH
2195 006200 001401          BEQ    69$
2196 006202 104006          ERROR 6          ;*****TEST 21 - ERROR 6*****
2197                                     ;R4 SHOULD STILL BE DESTINATION LENGTH
2198 006204 020567 172446    69$:  CMP    R5,DSTAD    ;CHECK R5 = DESTINATION ADDRESS
2199 006210 001401          BEQ    70$
2200 006212 104007          ERROR 7          ;*****TEST 21 - ERROR 7*****
2201                                     ;R5 SHOULD STILL BE DESTINATION ADDRESS
2202 006214 023706 000702    70$:  CMP    2*SAVR6,SP  ;VERIFY STACK POINTER IS RESTORED
2203 006220 001403          BEQ    71$
2204 006222 010637 000704    MOV    SP,2*BADR6
2205 006226 104010          ERROR 10         ;*****TEST 21 - ERROR 10*****
2206                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
2207                                     ;EXPECTED SP IS STORED AT "SAVR6"
2208                                     ;ERRONEOUS SP VALUE IS AT "BADR6"
2209 006230          71$:
2210                                     ;CHECK ANSWER
2211 006230 012700 006270    MOV    2*ANS21,R0   ;POINT R0 TO EXPECTED ANSWER
2212 006234 016701 172416    MOV    DSTAD,R1    ;POINT R1 TO RESULTANT ANSWER
2213 006240 016702 172410    MOV    DSTLN,R2    ;STORE ANSWER LENGTH IN R1
2214 006244 122021          72$:  CMPB   (R0)+,(R1)+  ;COMPARE EACH DIGIT
2215 006246 001401          BEQ    73$
2216 006250 104011          ERROR 11         ;*****TEST 21 - ERROR 11*****
2217                                     ;ERRONEOUS ANSWER
2218                                     ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2219                                     ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2220 006252 005302          73$:  DEC    R2          ;DECREMENT ANSWER LENGTH
2221 006254 001373          BNE   72$
2222 006256 000407          BR    ENDT21      ;BR IF NOT FINISHED
2223 006260          SIT21:
2224 006260          .BYTE 61
2225 006261          .BYTE 62
2226 006262          .BYTE 63
2227 006263          .BYTE 64
2228 006264          S2T21:
2229 006264          .BYTE 65
2230 006265          .BYTE 66
2231 006266          .BYTE 67
2232 006267          .BYTE 70
2233 006270          ANS21:
2234 006270          .BYTE 60
2235 006271          .BYTE 66
2236 006272          .BYTE 71
2237 006273          .BYTE 61
2238 006274          .BYTE 62
2239
2240          006276          .EVEN
2241 006276 016777 172370 172364 ENDT21: MOV    TPSW,2TVECT
2242 006304 106427 000200          MTPS 200

```

```

2243
2244
2245 ;:*****
2246 ;*TEST 22 TEST "SUBN" WITH POSITIVE OPERANDS, SRC1 .GT. SRC2
2247 ;:*****
2248 006310 000004
2249 006312 004567 006414
2250 006316 000004
2251 006320 006514
2252 006322 000003
2253 006324 006520
2254 006326 000003
2255 006330 004767 006470
2256 006334 004567 006504
2257 006340 000210
2258 006342 004767 006412
2259 006346 000277
2260 006350 000250
2261 006352 076051
2262
2263 006354 106767 172316
2264 006360 042767 177400 172310
2265 006366 023767 000700 172302
2266 006374 001401
2267 006376 104001
2268
2269
2270
2271 006400
2272 006400 005700
2273 006402 001401
2274 006404 104002
2275
2276 006406 005701
2277 006410 001401
2278 006412 104003
2279
2280 006414 005702
2281 006416 001401
2282 006420 104004
2283
2284 006422 005703
2285 006424 001401
2286 006426 104005
2287
2288 006430 020467 172220
2289 006434 001401
2290 006436 104006
2291
2292 006440 020567 172212
2293 006444 001401
2294 006446 104007
2295
2296 006450 023706 000702
2297 006454 001403
2298 006456 010637 000704

```

```

TST22: SCOPE
        JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
        4          ;SOURCE1 LENGTH
        S1T22      ;SOURCE1 ADDRESS
        3          ;SOURCE2 LENGTH
        S2T22      ;SOURCE2 ADDRESS
        3          ;DESTINATION LENGTH
        JSR      PC,CLBUF      ;CLEAR BUFFER AREA
        JSR      R5,XPSW
        .WORD     210
        JSR      PC,GENR      ;SET UP GENERAL REGISTERS
        SCC          ;SET UP THE COMPLEMENT OF EXPECTED CC'S
        CLN
        SUBN
        MFPS      CCODES      ;STORE RESULTANT PSW
        BIC      #177400,CCODES ;CLEAR UNUSED BITS
        CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$
        ERROR     1          ;BR, IF EQUAL
        *****TEST 22 - ERROR 1*****
        PSW ERROR
        EXPECTED PSW IS STORED AT "SAVR6"
        ERRONEOUS SP VALUE IS AT "BADR6"
        64$:
        TST      R0          ;CHECK R0=0
        BEQ      65$
        ERROR     2          ;*****TEST 22 - ERROR 2*****
        ;R0 SHOULD BE ZERO
        65$:
        TST      R1          ;CHECK R1=0
        BEQ      66$
        ERROR     3          ;*****TEST 22 - ERROR 3*****
        ;R1 SHOULD BE ZERO
        66$:
        TST      R2          ;CHECK R2=0
        BEQ      67$
        ERROR     4          ;*****TEST 22 - ERROR 4*****
        ;R2 SHOULD BE ZERO
        67$:
        TST      R3          ;CHECK R3=0
        BEQ      68$
        ERROR     5          ;*****TEST 22 - ERROR 5*****
        ;R3 SHOULD BE ZERO
        68$:
        CMP      R4,DSTLN      ;CHECK R4= DESTINATION LENGTH
        BEQ      69$
        ERROR     6          ;*****TEST 22 - ERROR 6*****
        ;R4 SHOULD STILL BE DESTINATION LENGTH
        69$:
        CMP      R5,DSTAD      ;CHECK R5 = DESTINATION ADDRESS
        BEQ      70$
        ERROR     7          ;*****TEST 22 - ERROR 7*****
        ;R5 SHOULD STILL BE DESTINATION ADDRESS
        CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
        BEQ      71$
        MCY      SP,@#BADR6      ;COPY BAC SP VALUE

```

```

2299 006462 104010          ERROR 10          ;*****TEST 22 - ERROR 10*****
2300                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
2301                                     ;EXPECTED SP IS STORED AT "SAVR6"
2302                                     ;ERRONEOUS SP VALUE IS AT "BADR6"
2303 006464          71$:
2304                                     ;CHECK ANSWER
2305 006464 012700 006523      MOV #ANS22,R0      ;POINT R0 TO EXPECTED ANSWER
2306 006470 016701 172162      MOV DSTAD,R1     ;POINT R1 TO RESULTANT ANSWER
2307 006474 016702 172154      MOV DSTLN,R2     ;STORE ANSWER LENGTH IN R1
2308 006500 122021          72$:      CMPB (R0)+,(R1)+ ;COMPARE EACH DIGIT
2309 006502 001401          BEQ 73$      ;BR IF EQUAL
2310 006504 104011          ERROR 11      ;*****TEST 22 - ERROR 11*****
2311                                     ;ERRONEOUS ANSWER
2312                                     ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2313                                     ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2314 006506 005302          73$:      DEC R2          ;DECREMENT ANSWER LENGTH
2315 006510 001373          BNE 72$      ;BR IF NOT FINISHED
2316 006512 000405          BR TST23     ;BR TO NEXT TEST
2317 006514          S1T22:          ;SOURCE1 STRING
2318 006514 060          .BYTE 60      ;MOST SIGNIFICANT DIGIT
2319 006515 071          .BYTE 71
2320 006516 066          .BYTE 66
2321 006517 062          .BYTE 62
2322 006520          S2T22:          ;SOURCE2 STRING
2323 006520 067          .BYTE 67      ;MOST SIGNIFICANT DIGIT
2324 006521 065          .BYTE 65
2325 006522 064          .BYTE 64
2326 006523          ANS22:          ;EXPECTED ANSWER
2327 006523 062          .BYTE 62      ;MOST SIGNIFICANT DIGIT
2328 006524 060          .BYTE 60
2329 006525 170          .BYTE 170
2330
2331          .EVEN

```

# F05

```

2332
2333
2334      ;*****
2335      ;*TEST 23      TEST "SUBN" WITH NEGATIVE OPERANDS, SRC1 .GTT. SRC2
2336      ;*****
2337      †ST23:  SCOPE
2338      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2339      3              ;SOURCE1 LENGTH
2340      S1T23         ;SOURCE1 ADDRESS
2341      4              ;SOURCE2 LENGTH
2342      S2T23         ;SOURCE2 ADDRESS
2343      3              ;DESTINATION LENGTH
2344      JSR      PC,CLBUF     ;CLEAR BUFFER AREA
2345      JSR      R5,XPSW
2346      .WORD     200
2347      JSR      PC,GENR     ;SET UP GENERAL REGISTERS
2348      SCC
2349      SUBN          ;SET UP THE COMPLEMENT OF EXPECTED CC'S
2350      MFPS        CCODES    ;STORE RESULTANT PSW
2351      BIC        #177400,CCODES ;CLEAR UNUSED BITS
2352      CMP        2#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALLE
2353      BEQ        64$
2354      ERROR      1
2355      ;*****TEST 23 - ERROR 1*****
2356      ;PSW ERROR
2357      ;EXPECTED PSW IS STORED AT "SAVR6"
2358      ;ERRONEOUS SP VALUE IS AT "BADR6"
2359      64$:
2360      TST        R0
2361      BEQ        65$
2362      ERROR      2
2363      ;*****TEST 23 - ERROR 2*****
2364      ;R0 SHOULD BE ZERO
2365      ;CHECK R0=0
2366      65$:
2367      TST        R1
2368      BEQ        66$
2369      ERROR      3
2370      ;*****TEST 23 - ERROR 3*****
2371      ;R1 SHOULD BE ZERO
2372      ;CHECK R1=0
2373      66$:
2374      TST        R2
2375      BEQ        67$
2376      ERROR      4
2377      ;*****TEST 23 - ERROR 4*****
2378      ;R2 SHOULD BE ZERO
2379      ;CHECK R2=0
2380      67$:
2381      TST        R3
2382      BEQ        68$
2383      ERROR      5
2384      ;*****TEST 23 - ERROR 5*****
2385      ;R3 SHOULD BE ZERO
2386      ;CHECK R3=0
2387      68$:
2388      CMP        R4,DSTLN
2389      BEQ        69$
2390      ERROR      6
2391      ;*****TEST 23 - ERROR 6*****
2392      ;R4 SHOULD STILL BE DESTINATION LENGTH
2393      ;CHECK R4 = DESTINATION LENGTH
2394      69$:
2395      CMP        R5,DSTAD
2396      BEQ        70$
2397      ERROR      7
2398      ;*****TEST 23 - ERROR 7*****
2399      ;R5 SHOULD STILL BE DESTINATION ADDRESS
2400      ;VERIFY STACK POINTER IS RESTORED
2401      ;BR IF OK
2402      MOV        SP,2#BADR6
2403      ERROR      10
2404      ;*****TEST 23 - ERROR 10*****
2405      ;STACK POINTER NOT RESTORED BY INSTRUCTION
  
```



```

2419
2420
2421
2422
2423 006742 000004
2424 006744 105777 171570
2425 006750 100556
2426 006752 026767 171572 171704
2427 006760 001007
2428 006762 032767 000001 171616
2429 006770 001403
2430 006772 005767 171576
2431 006776 001143
2432 007000
2433 007000 004567 005726
2434 007004 000004
2435 007006 007256
2436 007010 000004
2437 007012 007262
2438 007014 000005
2439 007016 004767 006002
2440 007022 012767 007104 171644
2441 007030 012777 015114 171632
2442 007036 005077 171630
2443 007042 004767 006022
2444 007046 013777 000554 171612
2445 007054 004567 005764
2446 007060 000010
2447 007062 106427 000000
2448 007066 052777 000100 171570
2449 007074 004767 005660
2450 007100 000277
2451 007102 000250
2452 007104 076051
2453
2454 007106 106767 171564
2455 007112 032777 000100 171544
2456 007120 001365
2457 007122 042767 177400 171546
2458 007130 023767 000700 171540
2459 007136 001401
2460 007140 104001
2461
2462
2463
2464 007142
2465 007142 005700
2466 007144 001401
2467 007146 104002
2468
2469 007150 005701
2470 007152 001401
2471 007154 104002
2472
2473 007156 005702
2474 007160 001401

```

```

*****
*TEST 24 TEST INTERRUPTABILITY OF "SUBN"
*****
TST24: SCOPE
TSTB @SWR ;TEST BIT 7 OF SWR
BMI TST25 ;SKIP TO NEXT TEST IF SET
CMP $TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T24CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT @BIT0, $ENV ;IF YES, IS PROGRAM UNDER APT?
BEQ T24CONT ;BR, IF NOT
TST $PASS ;IF YES, CHECK IF NOT ON FIRST PASS
BNE TST25 ;IF NOT, BR & SKIP TEST

T24CONT:
JSR R5, NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T24 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T24 ;SOURCE2 ADDRESS
5 ;DESTINATION LENGTH
JSR PC, CLBUF ;CLEAR BUFFER AREA
MOV @SUBNPC, PCI ;STORE PC OF TEST INSTRUCTION
MOV @INTR, @TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
JSR PC, TDONE ;WAIT FOR SLU READY
MOV @NULL, @TBUF ;SEND NULL CHARACTER
JSR R5, XPSW ;STORE EXPECTED PSW
WORD 10
MTPS #0 ;ALLOW INTERRUPTS
BIS #100, @TCSR ;ENABLE TTY INTERRUPTS
RESUBN: JSR PC, GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CLN
SUBNPC: SUBN

MFPS CCODES ;STORE RESULTANT PSW
BIT #100, @TCSR ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
BNE RESUBN ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
BIC #177400, CCODES ;CLEAR UNUSED BITS
CMP @EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 24 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"

64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 24 - ERROR 2*****
;R0 SHOULD BE ZERO

65$:
TST R1 ;CHECK R1=0
BEQ 66$
ERROR 3 ;*****TEST 24 - ERROR 3*****
;R1 SHOULD BE ZERO

66$:
TST R2 ;CHECK R2=0
BEQ 67$

```

2475	007162	104004			ERROR	4		*****TEST 24 - ERROR 4*****
2476								:R2 SHOULD BE ZERO
2477	007164	005703		67\$:	TST	R3		:CHECK R3=0
2478	007166	001401			BEQ	68\$		
2479	007170	104005			ERROR	5		*****TEST 24 - ERROR 5*****
2480								:R3 SHOULD BE ZERO
2481	007172	020467	171456	68\$:	CMP	R4,DSTLN		:CHECK R4= DESTINATION LENGTH
2482	007176	001401			BEQ	69\$		
2483	007200	104006			ERROR	6		*****TEST 24 - ERROR 6*****
2484								:R4 SHOULD STILL BE DESTINATION LENGTH
2485	007202	020567	171450	69\$:	CMP	R5,DSTAD		:CHECK R5 = DESTINATION ADDRESS
2486	007206	001401			BEQ	70\$		
2487	007210	104007			ERROR	7		*****TEST 24 - ERROR 7*****
2488								:R5 SHOULD STILL BE DESTINATION ADDRESS
2489	007212	023706	000702	70\$:	CMP	2#SAVR6,SP		:VERIFY STACK POINTER IS RESTORED
2490	007216	001403			BEQ	71\$		:BR IF OK
2491	007220	010637	000704		MOV	SP,2#BADR6		:COPY BAD SP VALUE
2492	007224	104010			ERROR	10		*****TEST 24 - ERROR 10*****
2493								:STACK POINTER NOT RESTORED BY INSTRUCTION
2494								:EXPECTED SP IS STORED AT "SAVR6"
2495								:ERRONEOUS SP VALUE IS AT "BADR6"
2496	007226			71\$:				
2497								:CHECK ANSWER
2498	007226	012700	007266		MOV	#ANS24,R0		:POINT R0 TO EXPECTED ANSWER
2499	007232	016701	171420		MOV	DSTAD,R1		:POINT R1 TO RESULTANT ANSWER
2500	007236	016702	171412		MOV	DSTLN,R2		:STORE ANSWER LENGTH IN R1
2501	007242	122021		72\$:	CMPB	(R0)+,(R1)+		:COMPARE EACH DIGIT
2502	007244	001401			BEQ	73\$		:BR IF EQUAL
2503	007246	104011			ERROR	11		*****TEST 24 - ERROR 11*****
2504								:ERRONEOUS ANSWER
2505								:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2506								:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2507	007250	005302		73\$:	DEC	R2		:DECREMENT ANSWER LENGTH
2508	007252	001373			BNE	72\$		:BR IF NOT FINISHED
2509	007254	000407			BR	ENDT24		:BR TO END OF THIS TEST
2510	007256			S1T24:				:SOURCE1 STRING
2511	007256	061			.BYTE	61		:MOST SIGNIFICANT DIGIT
2512	007257	062			.BYTE	62		
2513	007260	063			.BYTE	63		
2514	007261	064			.BYTE	64		
2515	007262			S2T24:				:SOURCE2 STRING
2516	007262	065			.BYTE	65		:MOST SIGNIFICANT DIGIT
2517	007263	066			.BYTE	66		
2518	007264	067			.BYTE	67		
2519	007265	170			.BYTE	170		
2520	007266			ANS24:				:EXPECTED ANSWER
2521	007266	060			.BYTE	60		:MOST SIGNIFICANT DIGIT
2522	007267	066			.BYTE	66		
2523	007270	071			.BYTE	71		
2524	007271	061			.BYTE	61		
2525	007272	162			.BYTE	162		
2526								
2527		007274			.EVEN			
2528	007274	016777	171372	171366	ENDT24: MOV	TPSW,2#VECT		
2529	007302	106427	000200		MTPS	#200		

J05

2530  
2531  
2532  
2533  
2534  
2535  
2536  
2537  
2538  
2539  
2540  
2541  
2542  
2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585

007306 000004  
007310 004567 005416  
007314 000003  
007316 007460  
007320 000003  
007322 007463  
007324 000377  
007326 004567 005512  
007332 000204  
007334 004767 005420  
007340 000277  
007342 000244  
007344 076052  
007346 106767 171324  
007352 042767 177400 171316  
007360 023767 000700 171310  
007366 001401  
007370 104001  
007372  
007372 005700  
007374 001401  
007376 104002  
007400 005701  
007402 001401  
007404 104003  
007406 005702  
007410 001401  
007412 104004  
007414 005703  
007416 001401  
007420 104005  
007422 020467 171226  
007426 001401  
007430 104006  
007432 020567 171220  
007436 001401  
007440 104007  
007442 023706 000702

```
::*****  
:*TEST 25 TEST "CMPN" WITH ALL ZEROES IN BOTH SOURCE STRINGS  
:*****  
↑ST25: SCOPE  
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST  
3 ;SOURCE1 LENGTH  
S1T25 ;SOURCE1 ADDRESS  
3 ;SOURCE2 LENGTH  
S2T25 ;SOURCE2 ADDRESS  
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN  
JSR R5,XPSW  
.WORD 204  
JSR PC,GENP ;SET UP GENERAL REGISTERS  
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S  
CLZ  
CMPN  
MFPS 000DES ;STORE RESULTANT PSW  
BIC #177400,CCODES ;CLEAR UNUSED BITS  
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 25 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "SAVR6"  
;ERRONEOUS SP VALUE IS AT "BADP6"  
64$:  
TST R0 ;CHECK R0=0  
BEQ 65$  
ERROR 2 ;*****TEST 25 - ERROR 2*****  
;R0 SHOULD BE ZERO  
;CHECK R1=0  
65$:  
TST R1  
BEQ 66$  
ERROR 3 ;*****TEST 25 - ERROR 3*****  
;R1 SHOULD BE ZERO  
;CHECK R2=0  
66$:  
TST R2  
BEQ 67$  
ERROR 4 ;*****TEST 25 - ERROR 4*****  
;R2 SHOULD BE ZERO  
;CHECK R3=0  
67$:  
TST R3  
BEQ 68$  
ERROR 5 ;*****TEST 25 - ERROR 5*****  
;R3 SHOULD BE ZERO  
;CHECK R4 UNCHANGED  
;BR IF OK  
68$:  
CMP R4,DSTLN  
BEQ 69$  
ERROR 6 ;*****TEST 25 - ERROR 6*****  
;R4 CHANGED  
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"  
;CHECK R5 UNCHANGED  
69$:  
CMP R5,DSTAD  
BEQ 70$  
ERROR 7 ;*****TEST 25 - ERROR 7*****  
;R5 CHANGED  
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"  
70$:  
CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
```

K05

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 62  
CVKAJA.P11 20-DEC-76 15:02 T25

TEST "CMPN" WITH ALL ZEROES IN BOTH SOURCE STRINGS

SEG 0064

```

2586 007446 001403      BEQ      71$      ;BR IF OK
2587 007450 010637 000704  MOV      SP,0#BADR6 ;COPY BAD SP VALUE
2588 007454 104010      ERROR    10      ;*****TEST 25 - ERROR 10*****
2589                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
2590                                     ;EXPECTED SP IS STORED AT "SAVR6"
2591                                     ;ERRONEOUS SP VALUE IS AT "BADR6"
2592 007456                                     71$:
2593 007456 000403      BR       TST26    ;BR TO NEXT TEST
2594 007460      S1T25:   .BYTE    60      ;SOURCE1 STRING
2595 007460      .BYTE    60      ;MOST SIGNIFICANT DIGIT
2596 007461      .BYTE    60
2597 007462      .BYTE    60
2598 007463      S2T25:   .BYTE    60      ;SOURCE2 STRING
2599 007463      .BYTE    60      ;MOST SIGNIFICANT DIGIT
2600 007464      .BYTE    60
2601 007465      .BYTE    60
2602
2603      .EVEN

```

```

2604
2605
2606
2607
2608
2609 007466 000004
2610 007470 004567 005236
2611 007474 000004
2612 007476 007640
2613 007500 000004
2614 007502 007644
2615 007504 000377
2616 007506 004567 005332
2617 007512 000204
2618 007514 004767 005240
2619 007520 000277
2620 007522 000244
2621
2622 007524 076052
2623
2624 007526 106767 171144
2625 007532 042767 177490 171136
2626 007540 023767 000700 171130
2627 007546 001401
2628 007550 104001
2629
2630
2631
2632 007552
2633 007552 005700
2634 007554 001401
2635 007556 104002
2636
2637 007560 005701
2638 007562 001401
2639 007564 104003
2640
2641 007566 005702
2642 007570 001401
2643 007572 104004
2644
2645 007574 005703
2646 007576 001401
2647 007600 104005
2648
2649 007602 020467 171046
2650 007606 001401
2651 007610 104006
2652
2653
2654 007612 020567 171040
2655 007616 001401
2656 007620 104007
2657
2658
2659 007622 023706 000702

```

```

*****
*TEST 26 TEST "CMPN" WITH SRC1 = SRC2
*****
TST26: SCOPE
        JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
        4          ;SOURCE1 LENGTH
        S1T26    ;SOURCE1 ADDRESS
        4          ;SOURCE2 LENGTH
        S2T26    ;SOURCE2 ADDRESS
        377       ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
        JSR      R5,XPSW
        .WORD    204
        JSR      PC,GENP      ;SET UP GENERAL REGISTERS
        SCC
        CLZ       ;SET UP THE COMPLEMENT OF EXPECTED CC'S
        CMPN
        MFPS     CCODES      ;STORE RESULTANT PSW
        BIC      #177400,CCODES ;CLEAR UNUSED BITS
        CMP      @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ     64$         ;BR, IF EQUAL
        ERROR   1          ;*****TEST 26 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "SAVR6"
                          ;ERRONEOUS SP VALUE IS AT "BADP6"
64$:
        TST     R0          ;CHECK R0=0
        BEQ     55$
        ERROR   2          ;*****TEST 26 - ERROR 2*****
                          ;R0 SHOULD BE ZERO
                          ;CHECK R1=0
65$:
        TST     R1          ;CHECK R1=0
        BEQ     66$
        ERROR   3          ;*****TEST 26 - ERROR 3*****
                          ;R1 SHOULD BE ZERO
                          ;CHECK R2=0
66$:
        TST     R2          ;CHECK R2=0
        BEQ     67$
        ERROR   4          ;*****TEST 26 - ERROR 4*****
                          ;R2 SHOULD BE ZERO
                          ;CHECK R3=0
67$:
        TST     R3          ;CHECK R3=0
        BEQ     68$
        ERROR   5          ;*****TEST 26 - ERROR 5*****
                          ;R3 SHOULD BE ZERO
                          ;CHECK R4 UNCHANGED
                          ;BR IF OK
68$:
        CMP     R4,DSTLN    ;*****TEST 26 - ERROR 6*****
        BEQ     69$
        ERROR   6          ;R4 CHANGED
                          ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
                          ;CHECK R5 UNCHANGED
69$:
        CMP     R5,DSTAD    ;*****TEST 26 - ERROR 7*****
        BEQ     70$
        ERROR   7          ;R5 CHANGED
                          ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
70$:
        CMP     @SAVR6,SP  ;VERIFY STACK POINTER IS RESTORED

```

M05

MAIN MACY11 27(1006) 21-DEC-76 11:53 PAGE 64  
DYKARA.P11 20-DEC-76 15:02 T26

TEST "CMPN" WITH SRC1 = SRC2

SEQ 0066

2660	007626	001403		BEQ	71\$	;BR IF OK
2661	007630	010637	000704	MOV	SP,2#BADR6	;COPY BAD SP VALUE
2662	007634	104010		ERROR	10	;*****TEST 26 - ERROR 10*****
2663						;STACK POINTER NOT RESTORED BY INSTRUCTION
2664						;EXPECTED SP IS STORED AT "SAVR6"
2665						;ERRONEOUS SP VALUE IS AT "BADR6"
2666	007636			71\$:		
2667	007636	000404		BR	TST27	;BR TO NEXT TEST
2668	007640			S1T26:		;SOURCE1 STRING
2669	007640	071		.BYTE	71	;MOST SIGNIFICANT DIGIT
2670	007641	066		.BYTE	66	
2671	007642	062		.BYTE	62	
2672	007643	063		.BYTE	63	
2673	007644			S2T26:		;SOURCE2 STRING
2674	007644	071		.BYTE	71	;MOST SIGNIFICANT DIGIT
2675	007645	066		.BYTE	66	
2676	007646	062		.BYTE	62	
2677	007647	063		.BYTE	63	
2678						
2679				.EVEN		

N05

```

2680
2681
2682
2683
2684
2685 007650 000004
2686 007652 004567 005054
2687 007656 000004
2688 007660 010020
2689 007662 000004
2690 007664 010024
2691 007666 000377
2692 007670 004567 005150
2693 007674 000200
2694 007676 004767 005056
2695 007702 000277
2696
2697 007704 076052
2698
2699 007706 106767 170764
2700 007712 042767 177400 170756
2701 007720 023767 000700 170750
2702 007726 001401
2703 007730 104001
2704
2705
2706
2707 007732
2708 007732 005700
2709 007734 001401
2710 007736 104002
2711
2712 007740 005701
2713 007742 001401
2714 007744 104003
2715
2716 007746 005702
2717 007750 001401
2718 007752 104004
2719
2720 007754 005703
2721 007756 001401
2722 007760 104005
2723
2724 007762 020467 170666
2725 007766 001401
2726 007770 104006
2727
2728
2729 007772 020567 170660
2730 007776 001401
2731 010000 104007
2732
2733
2734 010002 023706 000702
2735 010006 001403

```

```

*****
: *TEST 27 TEST "CMPN" WITH IDENTICAL NON-ZERO MAGNITUDES, NEGATIVE SOURCE2. POSITI
*****
TST27: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T27 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T27 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
JSR R5,XPSW
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CMPN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 27 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$: TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 27 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$: TST R1
BEQ 66$
ERROR 3 ;*****TEST 27 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$: TST R2
BEQ 67$
ERROR 4 ;*****TEST 27 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$: TST R3
BEQ 68$
ERROR 5 ;*****TEST 27 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4 UNCHANGED
;BR IF OK
68$: CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 27 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
69$: CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 27 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
70$: CMP #SAVR6,SP
BEQ 71$

```

2736	010010	010637	000704	MOV	SP,3#BADR6	:COPY BAD SP VALUE
2737	010014	104010		ERROR	10	;*****TEST 27 - ERROR 10*****
2738						;STACK POINTER NOT RESTORED BY INSTRUCTION
2739						;EXPECTED SP IS STORED AT "SAVR6"
2740						;ERRONEOUS SP VALUE IS AT "BADR6"
2741	010016			71\$:		
2742	010016	000404		BR	TST30	:BR TO NEXT TEST
2743	010020			S1T27:		:SOURCE1 STRING
2744	010020	071		.BYTE	71	:MOST SIGNIFICANT DIGIT
2745	010021	067		.BYTE	67	
2746	010022	065		.BYTE	65	
2747	010023	063		.BYTE	63	
2748	010024			S2^27:		:SOURCE2 STRING
2749	010024	071		.BYTE	71	:MOST SIGNIFICANT DIGIT
2750	010025	067		.BYTE	67	
2751	010026	065		.BYTE	65	
2752	010027	163		.BYTE	163	
2753						
2754				.EVEN		

2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810

010030 000304  
010032 004567 004674  
010036 000004  
010040 010202  
010042 000004  
010044 010206  
010046 000377  
010050 004567 004770  
010054 000210  
010056 004767 004676  
010062 000277  
010064 000250  
010066 076052  
010070 106767 170602  
010074 042767 177400 170574  
010102 023767 000700 170566  
010110 001401  
010112 104001  
010114  
010114 005700  
010116 001401  
010120 104002  
010122 005701  
010124 001401  
010126 104003  
010130 005702  
010132 001401  
010134 104004  
010136 005703  
010140 001401  
010142 104005  
010144 020467 170504  
010150 001401  
010152 104006  
010154 020567 170476  
010160 001401  
010162 104007  
010164 023706 000702

```
::*****  
: *TEST 30 TEST "CMPN" WITH IDENTICAL NON-ZERO MAGNITUDES, POSITIVE SOURCE2, NEGATI  
:*****  
↑ST30: SCOPE  
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST  
4 ;SOURCE1 LENGTH  
S1T30 ;SOURCE1 ADDRESS  
4 ;SOURCE2 LENGTH  
S2T30 ;SOURCE2 ADDRESS  
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN  
JSR R5,XPSW  
.WORD 210  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S  
CLN  
CMPN  
MFPS CCODES ;STORE RESULTANT PSW  
BIC #177400,CCODES ;CLEAR UNUSED BITS  
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 30 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "SAVR6"  
;ERRONEOUS SP VALUE IS AT "BADR6"  
64$: TST R0 ;CHECK R0=0  
BEQ 65$  
ERROR 2 ;*****TEST 30 - ERROR 2*****  
;R0 SHOULD BE ZERO  
;CHECK R1=0  
65$: TST R1  
BEQ 66$  
ERROR 3 ;*****TEST 30 - ERROR 3*****  
;R1 SHOULD BE ZERO  
;CHECK R2=0  
66$: TST R2  
BEQ 67$  
ERROR 4 ;*****TEST 30 - ERROR 4*****  
;R2 SHOULD BE ZERO  
;CHECK R3=0  
67$: TST R3  
BEQ 68$  
ERROR 5 ;*****TEST 30 - ERROR 5*****  
;R3 SHOULD BE ZERO  
;CHECK R4 UNCHANGED  
;BR IF OK  
68$: CMP R4,DSTLN  
BEQ 69$  
ERROR 6 ;*****TEST 30 - ERROR 6*****  
;R4 CHANGED  
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"  
;CHECK R5 UNCHANGED  
69$: CMP R5,DSTAD  
BEQ 70$  
ERROR 7 ;*****TEST 30 - ERROR 7*****  
;R5 CHANGED  
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"  
70$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
```

2811	010170	001403		BEQ	71\$		;BR IF OK
2812	010172	010637	000704	MOV	SP,#BADR6		;COPY BAD SP VALUE
2813	010176	104010		ERROR	10		;*****TEST 30 - ERROR 10*****
2814							;STACK POINTER NOT RESTORED BY INSTRUCTION
2815							;EXPECTED SP IS STORED AT "SAVR6"
2816							;ERRONEOUS SP VALUE IS AT "BADR6"
2817	010200					71\$:	
2818	010200	000404		BR	TST31		;BR TO NEXT TEST
2819	010202			S1T30:			;SOURCE1 STRING
2820	010202	070		.BYTE	70		;MOST SIGNIFICANT DIGIT
2821	010203	066		.BYTE	66		
2822	010204	064		.BYTE	64		
2823	010205	162		.BYTE	162		
2824	010206			S2T30:			;SOURCE2 STRING
2825	010206	070		.BYTE	70		;MOST SIGNIFICANT DIGIT
2826	010207	066		.BYTE	66		
2827	010210	064		.BYTE	64		
2828	010211	062		.BYTE	62		
2829							
2830				.EVEN			

2831  
2832  
2833  
2834  
2835  
2836  
2837  
2838  
2839  
2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886

010212 000004  
010214 004567 004512  
010220 000002  
010222 010364  
010224 000002  
010226 010366  
010230 000377  
010232 004567 004606  
010236 000200  
010240 004767 004514  
010244 000277  
010246 000250  
010250 076052  
010252 106767 170420  
010256 042767 177400 170412  
010264 023767 000700 170404  
010272 001401  
010274 104001  
010276  
010276 005700  
010300 001401  
010302 104002  
010304 005701  
010306 001401  
010310 104003  
010312 005702  
010314 001401  
010316 104004  
010320 005703  
010322 001401  
010324 104005  
010326 020467 170322  
010332 001401  
010334 104006  
010336 020567 170314  
010342 001401  
010344 104007

```

:*****
:*TEST 31 TEST "CMPN" WITH +SRC1 & -SRC2, S1L = S2L, /S1/ .GT. /S2/
:*****
TST31: SCOPE
        JSR      R5, XPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
        2          ;SOURCE1 LENGTH
        S1T31    ;SOURCE1 ADDRESS
        2          ;SOURCE2 LENGTH
        S2T31    ;SOURCE2 ADDRESS
        377       ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
        JSR      R5, XPSW
        .WORD    200
        JSR      PC, GENR      ;SET UP GENERAL REGISTERS
        SCC
        CLN                  ;SET UP THE COMPLEMENT OF EXPECTED CC'S
        CMPN
        MFPS     CCODES        ;STORE RESULTANT PSW
        BIC      #177400, CCODES ;CLEAR UNUSED BITS
        CMP      2#EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$
        ERROR    1             ;*****TEST 31 - ERROR 1*****
                                ;PSW ERROR
                                ;EXPECTED PSW IS STORED AT "SAVR6"
                                ;ERRONEOUS SP VALUE IS AT "BADRE"
64$:   TST      R0              ;CHECK R0=0
        BEQ      65$
        ERROR    2             ;*****TEST 31 - ERROR 2*****
                                ;R0 SHOULD BE ZERO
                                ;CHECK R1=0
65$:   TST      R1
        BEQ      66$
        ERROR    3             ;*****TEST 31 - ERROR 3*****
                                ;R1 SHOULD BE ZERO
                                ;CHECK R2=0
66$:   TST      R2
        BEQ      67$
        ERROR    4             ;*****TEST 31 - ERROR 4*****
                                ;R2 SHOULD BE ZERO
                                ;CHECK R3=0
67$:   TST      R3
        BEQ      68$
        ERROR    5             ;*****TEST 31 - ERROR 5*****
                                ;R3 SHOULD BE ZERO
                                ;CHECK R4 UNCHANGED
                                ;BR IF OK
68$:   CMP      R4, DSTLN
        BEQ      69$
        ERROR    6             ;*****TEST 31 - ERROR 6*****
                                ;R4 CHANGED
                                ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
                                ;CHECK R5 UNCHANGED
69$:   CMP      R5, DSTAD
        BEQ      70$
        ERROR    7             ;*****TEST 31 - ERROR 7*****
                                ;R5 CHANGED
                                ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"

```

F06

MAIN MACY11 27(1006) 21-DEC-76 11:53 PAGE 70  
CVKJJA.P11 20-DEC-76 15:02 T31

TEST "CMPN" WITH +SRC1 & -SRC2, S1L = S2L, /S1/ .GT. /S2/

SEQ 0072

2887	010346	023706	000702	70\$:	CMP	2#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
2888	010352	001403			BEG	71\$	:BR IF OK
2889	010354	010637	000704		MOV	SP,2#BADR6	:COPY BAD SP VALUE
2890	010360	104010			ERROR	10	:*****TEST 31 - ERROR 10*****
2891							:STACK POINTER NOT RESTORED BY INSTRUCTION
2892							:EXPECTED SP IS STORED AT "SAVR6"
2893							:ERRONEOUS SP VALUE IS AT "BADR6"
2894	010362			71\$:			
2895	010362	000402			BR	TST32	:BR TO NEXT TEST
2896	010364			S1T31:			:SOURCE1 STRING
2897	010364	071			.BYTE	71	:MOST SIGNIFICANT DIGIT
2898	010365	066			.BYTE	66	
2899	010366			S2T31:			:SOURCE2 STRING
2900	010366	066			.BYTE	66	:MOST SIGNIFICANT DIGIT
2901	010367	171			.BYTE	171	
2902							
2903					.EVEN		

2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959

010370 000004  
010372 004567 004334  
010376 000003  
010400 010542  
010402 000003  
010404 010545  
010406 000377  
010410 004567 004430  
010414 000210  
010416 004767 004336  
010422 000267  
010424 000250  
010426 076052  
010430 106767 170242  
010434 042767 177400 170234  
010442 023767 000700 170226  
010450 001401  
010452 104001  
010454  
010454 005700  
010456 001401  
010460 104002  
010462 005701  
010464 001401  
010466 104003  
010470 005702  
010472 001401  
010474 104004  
010476 005703  
010500 001401  
010502 104005  
010504 020467 170144  
010510 001401  
010512 104006  
010514 020567 170136  
010520 001401  
010522 104007  
010524 023706 000702

```
*****  
*TEST 32 TEST "CMPN" WITH -SRC1 & +SRC2, S1L = S2L, /S1/ .GT. /S2/  
*****  
*ST32: SCOPE  
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST  
3 ;SOURCE1 LENGTH  
S1T32 ;SOURCE1 ADDRESS  
3 ;SOURCE2 LENGTH  
S2T32 ;SOURCE2 ADDRESS  
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN  
JSR R5,XPSW  
.WORD 210  
JSR PC,GENP ;SET UP GENERAL REGISTERS  
+SEV!SEZ!SEC ;SET UP THE COMPLEMENT OF EXPECTED CC'S  
CLN  
CMPN  
MFPS CCODES ;STORE RESULTANT PSW  
BIC #177400,CCODES ;CLEAR UNUSED BITS  
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE  
BEQ 64$ ;BR, IF EQUAL  
ERROR 1 ;*****TEST 32 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT "SAVR5"  
;ERRONEOUS SP VALUE IS AT "BADR6"  
64$: TST R0 ;CHECK R0=0  
BEQ 65$  
ERROR 2 ;*****TEST 32 - ERROR 2*****  
;R0 SHOULD BE ZERO  
;CHECK R1=0  
65$: TST R1  
BEQ 66$  
ERROR 3 ;*****TEST 32 - ERROR 3*****  
;R1 SHOULD BE ZERO  
;CHECK R2=0  
66$: TST R2  
BEQ 67$  
ERROR 4 ;*****TEST 32 - ERROR 4*****  
;R2 SHOULD BE ZERO  
;CHECK R3=0  
67$: TST R3  
BEQ 68$  
ERROR 5 ;*****TEST 32 - ERROR 5*****  
;R3 SHOULD BE ZERO  
;CHECK R4 UNCHANGED  
;BR IF OK  
68$: CMP R4,DSTLN  
BEQ 69$  
ERROR 6 ;*****TEST 32 - ERROR 6*****  
;R4 CHANGED  
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"  
;CHECK R5 UNCHANGED  
69$: CMP R5,DSTAD  
BEQ 70$  
ERROR 7 ;*****TEST 32 - ERROR 7*****  
;R5 CHANGED  
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"  
70$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
```

H06

TEST "CMPN" WITH -SRC1 & +SRC2, S1L = S2L, /S1/ .GT. /S2/

SEQ 0074

2960	010530	001403		BEQ	71\$	:BR IF OK
2961	010532	010637	000704	MOV	SP,2#BADR6	:COPY BAD SP VALUE
2962	010536	104010		ERRCR	10	:*****TEST 32 - ERRCR 10*****
2963						:STACK POINTER NOT RESTORED BY INSTRUCTION
2964						:EXPECTED SP IS STORED AT "SAVR6"
2965						:ERRONEOUS SP VALUE IS AT "BADR6"
2966	010540		71\$:			
2967	010540	000403		BR	TST33	:BR TO NEXT TEST
2968	010542		S1T32:			:SOURCE1 STRING
2969	010542	070		.BYTE	70	:MOST SIGNIFICANT DIGIT
2970	010543	063		.BYTE	63	
2971	010544	161		.BYTE	161	
2972	010545		S2T22:			:SOURCE2 STRING
2973	010545	061		.BYTE	61	:MOST SIGNIFICANT DIGIT
2974	010546	063		.BYTE	63	
2975	010547	070		.BYTE	70	
2976				.EVEN		

```

2978
2979
2980
2981
2982
2983 010550 000304
2984 010552 004567 004154
2985 010556 000004
2986 010560 010722
2987 010562 000004
2988 010564 010726
2989 010566 000377
2990 010570 004567 004250
2991 010574 000210
2992 010576 004767 004156
2993 010602 000277
2994 010604 000250
2995
2996 010606 076052
2997
2998 010610 106767 170062
2999 010614 042767 177400 170054
3000 010622 023767 000700 170046
3001 010630 001401
3002 010632 104001
3003
3004
3005
3006 010634
3007 010634 005700
3008 010636 001401
3009 010640 104002
3010
3011 010642 005701
3012 010644 001401
3013 010646 104003
3014
3015 010650 005702
3016 010652 001401
3017 010654 104004
3018
3019 010656 005703
3020 010660 001401
3021 010662 104005
3022
3023 010664 020467 167764
3024 010670 001401
3025 010672 104006
3026
3027
3028 010674 020567 167756
3029 010700 001401
3030 010702 104007
3031
3032
3033 010704 023706 000702

```

```

;*****
;TEST 33 TEST "CMPN" WITH S1L = S2L, POSITIVE SOURCE2, SOURCE2 MAGNITUDE IS GREAT
;*****
TST33: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T33 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T33 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
JSR R5,XPSW
.WORD 210
JSR PC,GENP ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CLN
CMPN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 33 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR5"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$: TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 33 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$: TST R1
BEQ 66$
ERROR 3 ;*****TEST 33 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$: TST R2
BEQ 67$
ERROR 4 ;*****TEST 33 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$: TST R3
BEQ 68$
ERROR 5 ;*****TEST 33 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4 UNCHANGED
;BR IF OK
68$: CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 33 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
69$: CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 33 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
70$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED

```

# JOB

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 74  
 SYKARA.P11 20-DEC-76 15:02 T33 TEST "CMPN" WITH S1L = S2L, POSITIVE SOURCE2, SOURCE2 MAGNITUDE IS GREATER SEQ 0076

3034	010710	001403		BEQ	71\$		:BR IF OK
3035	010712	010637	000704	MOV	SP,2#BADR6		:COPY BAD SP VALUE
3036	010716	104010		ERROR	10		:*****TEST 33 - ERROR 10*****
3037							:STACK POINTER NOT RESTORED BY INSTRUCTION
3038							:EXPECTED SP IS STORED AT "SAVR6"
3039							:ERRONEOUS SP VALUE IS AT "BADR6"
3040	010720						
3041	010720	000404		BR	TST34		:BR TO NEXT TEST
3042	010722			S1T33:			:SOURCE1 STRING
3043	010722	067		.BYTE	67		:MOST SIGNIFICANT DIGIT
3044	010723	066		.BYTE	66		
3045	010724	065		.BYTE	65		
3046	010725	064		.BYTE	64		
3047	010726			S2T33:			:SOURCE2 STRING
3048	010726	067		.BYTE	67		:MOST SIGNIFICANT DIGIT
3049	010727	066		.BYTE	66		
3050	010730	065		.BYTE	65		
3051	010731	065		.BYTE	65		
3052							
3053				.EVEN			

K06

```

3054
3055
3056 ::*****
3057 :*TEST 34 TEST "CMPN" WITH S2L .GT. S1L, SUCCESSFUL COMPARE
3058 :*****
3059 TST34: SCOPE
3060 010732 000004 JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3061 010734 004567 003772 JSR 2 ;SOURCE1 LENGTH
3062 010740 000002 S1T34 ;SOURCE1 ADDRESS
3063 010742 011104 4 ;SOURCE2 LENGTH
3064 010744 000004 S2T34 ;SOURCE2 ADDRESS
3065 010746 011106 377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3066 010750 000377 JSR R5,XPSW
3067 010752 004567 004066 .WORD 204
3068 010756 000204 JSR PC,GENF ;SET UP GENERAL REGISTERS
3069 010760 004767 003774 SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
3070 010764 000277 CLZ
3071 010766 000244
3072 010770 076052 CMPN
3073
3074 010772 106767 167700 MFPS CCODES ;STORE RESULTANT PSW
3075 010776 042767 177400 167672 BIC #177400,CCODES ;CLEAR UNUSED BITS
3076 011004 023767 000700 167664 CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
3077 011012 001401 BEQ 64$ ;BR, IF EQUAL
3078 011014 104001 ERROR 1 ;*****TEST 34 - ERROR 1*****
3079 ;PSW ERROR
3080 ;EXPECTED PSW IS STORED AT "SAVR6"
3081 ;ERRONEOUS SP VALUE IS AT "BADR6"
3082 011016 64$: TST R0 ;CHECK R0=0
3083 011016 005700 BEQ 65$
3084 011020 001401 ERROR 2 ;*****TEST 34 - ERROR 2*****
3085 011022 104002 ;R0 SHOULD BE ZERO
3086 ;CHECK R1=0
3087 011024 005701 65$: TST R1
3088 011026 001401 BEQ 66$
3089 011030 104003 ERROR 3 ;*****TEST 34 - ERROR 3*****
3090 ;R1 SHOULD BE ZERO
3091 011032 005702 66$: TST R2
3092 011034 001401 BEQ 67$ ;CHECK R2=0
3093 011036 104004 ERROR 4 ;*****TEST 34 - ERROR 4*****
3094 ;R2 SHOULD BE ZERO
3095 011040 005703 67$: TST R3
3096 011042 001401 BEQ 68$ ;CHECK R3=0
3097 011044 104005 ERROR 5 ;*****TEST 34 - ERROR 5*****
3098 ;R3 SHOULD BE ZERO
3099 011046 020467 167602 68$: CMP R4,DSTLN
3100 011052 001401 BEQ 69$ ;CHECK R4 UNCHANGED
3101 011054 104006 ERROR 6 ;BR IF OK
3102 ;*****TEST 34 - ERROR 6*****
3103 ;R4 CHANGED
3104 011056 020567 167574 69$: CMP R5,DSTAD
3105 011062 001401 BEQ 70$ ;CHECK R5 UNCHANGED
3106 011064 104007 ERROR 7 ;*****TEST 34 - ERROR 7*****
3107 ;R5 CHANGED
3108 ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
3109 011066 023706 000702 70$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED

```

3110	011072	001403		BEQ	715		;BR IF OK
3111	011074	010637	000704	MOV	SP,0#BADR6		;COPY BAD SP VALUE
3112	011100	104010		ERROR	10		;*****TEST 34 - ERROR 10*****
3113							;STACK POINTER NOT RESTORED BY INSTRUCTION
3114							;EXPECTED SP IS STORED AT "SAVR6"
3115							;ERRONEOUS SP VALUE IS AT "BADRE"
3116	011102						
3117	011102	000403		BR	TST35		;BR TO NEXT TEST
3118	011104			S1T34:			;SOURCE1 STRING
3119	011104	067			.BYTE 67		;MOST SIGNIFICANT DIGIT
3120	011105	071			.BYTE 71		
3121	011106			S2T34:			;SOURCE2 STRING
3122	011106	060			.BYTE 60		;MOST SIGNIFICANT DIGIT
3123	011107	060			.BYTE 60		
3124	011110	067			.BYTE 67		
3125	011111	071			.BYTE 71		
3126							
3127					.EVEN		

M06

```

3128
3129
3130
3131
3132
3133 011112 000004
3134 011114 004567 003612
3135 011120 000002
3136 011122 011262
3137 011124 000004
3138 011124 011264
3139 011130 000377
3140 011132 004567 003706
3141 011136 000200
3142 011140 004767 003614
3143 011144 000277
3144
3145 011146 076052
3146
3147 011150 106767 167522
3148 011154 042767 177400 167514
3149 011162 023767 000700 167506
3150 011170 001401
3151 011172 104001
3152
3153
3154
3155 011174
3156 011174 005700
3157 011176 001401
3158 011200 104002
3159
3160 011202 005701
3161 011204 001401
3162 011206 104003
3163
3164 011210 005702
3165 011212 001401
3166 011214 104004
3167
3168 011216 005703
3169 011220 001401
3170 011222 104005
3171
3172 011224 020467 167424
3173 011230 001401
3174 011232 104006
3175
3176
3177 011234 020567 167416
3178 011240 001401
3179 011242 104007
3180
3181
3182 011244 023706 000702
3183 011250 001403

```

```

:*****
:TEST 35 TEST "CMPN" WITH NEGATIVE OPERANDS & SRC2 LENGTH .GT. SRC1 LENGTH. NON-Z
:*****
TST35: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2 ;SOURCE1 LENGTH
S1T35 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T35 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
JSR R5,XPSW
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CMPN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 35 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 35 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$:
TST R1
BEQ 66$
ERROR 3 ;*****TEST 35 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$:
TST R2
BEQ 67$
ERROR 4 ;*****TEST 35 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$:
TST R3
BEQ 68$
ERROR 5 ;*****TEST 35 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4 UNCHANGED
;BR IF OK
68$:
CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 35 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
69$:
CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 35 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
70$:
CMP #SAVR6,SP
BEQ 71$

```

N06

3184	011252	010637	000704	MOV	SP, @#BADR6	; COPY BAD SP VALUE
3185	011256	104010		ERROR	10	; *****TEST 35 - ERROR 10*****
3186						; STACK POINTER NOT RESTORED BY INSTRUCTION
3187						; EXPECTED SP IS STORED AT "SAVR6"
3188						; ERRONEOUS SP VALUE IS AT "BADR6"
3189	011260			715:		
3190	011260	000403		BR	TST36	; BR TO NEXT TEST
3191	011262			S1T35:		; SOURCE1 STRING
3192	011262	070		.BYTE	70	; MOST SIGNIFICANT DIGIT
3193	011263	161		.BYTE	161	
3194	011264			S2T35:		; SOURCE2 STRING
3195	011264	061		.BYTE	61	; MOST SIGNIFICANT DIGIT
3196	011265	060		.BYTE	60	
3197	011266	070		.BYTE	70	
3198	011267	161		.BYTE	161	
3199						
3200				.EVEN		

```

3201
3202
3203
3204
3205
3206 011270 000004
3207 011272 004567 003434
3208 011276 000002
3209 011300 011442
3210 011302 000004
3211 011304 011445
3212 011306 000377
3213 011310 004567 003530
3214 011314 000210
3215 011316 004767 003436
3216 011322 000277
3217 011324 000250
3218
3219 011326 076052
3220
3221 011330 106767 167342
3222 011334 042767 177400 167334
3223 011342 023767 000700 167326
3224 011350 001401
3225 011352 104001
3226
3227
3228
3229 011354
3230 011354 005700
3231 011356 001401
3232 011360 104002
3233
3234 011362 005701
3235 011364 001401
3236 011366 104003
3237
3238 011370 005702
3239 011372 001401
3240 011374 104004
3241
3242 011376 005703
3243 011400 001401
3244 011402 104005
3245
3246 011404 020467 167244
3247 011410 001401
3248 011412 104006
3249
3250
3251 011414 020567 167236
3252 011420 001401
3253 011422 104007
3254
3255
3256 011424 023706 000702

```

```

*****
*TEST 36 TEST "CMPN" WITH POSITIVE OPERANDS & SRC2 LENGTH .GT. SRC1 LENGTH, NON-Z
*****
TST36: SCOPE
        JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
        2          ;SOURCE1 LENGTH
        S1T36     ;SOURCE1 ADDRESS
        4          ;SOURCE2 LENGTH
        S2T36     ;SOURCE2 ADDRESS
        377       ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
        JSR      R5,XPSW
        .WORD    210
        JSR      PC,GENR      ;SET UP GENERAL REGISTERS
        SCC
        CLN                ;SET UP THE COMPLEMENT OF EXPECTED CC'S
        CMPN
        MFPS     CCODES      ;STORE RESULTANT PSW
        BIC      #177400,CCODES ;CLEAR UNUSED BITS
        CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ     64$
        ERROR   1          ;BR, IF EQUAL
                          ;*****TEST 36 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "SAVR6"
                          ;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
        TST     R0          ;CHECK R0=0
        BEQ     65$
        ERROR   2          ;*****TEST 36 - ERROR 2*****
                          ;R0 SHOULD BE ZERO
                          ;CHECK R1=0
65$:
        TST     R1
        BEQ     66$
        ERROR   3          ;*****TEST 36 - ERROR 3*****
                          ;R1 SHOULD BE ZERO
                          ;CHECK R2=0
66$:
        TST     R2
        BEQ     67$
        ERROR   4          ;*****TEST 36 - ERROR 4*****
                          ;R2 SHOULD BE ZERO
                          ;CHECK R3=0
67$:
        TST     R3
        BEQ     68$
        ERROR   5          ;*****TEST 36 - ERROR 5*****
                          ;R3 SHOULD BE ZERO
                          ;CHECK R4 UNCHANGED
                          ;BR IF OK
68$:
        CMP     R4,DSTLN
        BEQ     69$
        ERROR   6          ;*****TEST 36 - ERROR 6*****
                          ;R4 CHANGED
                          ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
                          ;CHECK R5 UNCHANGED
69$:
        CMP     R5,DSTAD
        BEQ     70$
        ERROR   7          ;*****TEST 36 - ERROR 7*****
                          ;R5 CHANGED
                          ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
70$:
        CMP     @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED

```

3257	011430	001403		BEG	71\$		;BR IF OK
3258	011432	010637	000704	MOV	SP,2#BADR6		;COPY BAD SP VALUE
3259	011436	104010		ERROR	10		;*****TEST 36 - ERROR 10*****
3260							;STACK POINTER NOT RESTORED BY INSTRUCTION
3261							;EXPECTED SP IS STORED AT "SAVR6"
3262							;ERRONEOUS SP VALUE IS AT "BADR6"
3263	011440			71\$:			
3264	011440	000403		BR	TST37		;BR TO NEXT TEST
3265	011442			S1T36:			;SOURCE1 STRING
3266	011442	070			.BYTE 70		;MOST SIGNIFICANT DIGIT
3267	011443	061			.BYTE 61		
3268	011444	061			.BYTE 61		
3269	011445			S2T36:			;SOURCE2 STRING
3270	011445	060			.BYTE 60		;MOST SIGNIFICANT DIGIT
3271	011446	070			.BYTE 70		
3272	011447	061			.BYTE 61		
3273							
3274					.EVEN		

```

3275
3276
3277
3278
3279
3280 011450 000004
3281 011452 004567 003254
3282 011456 000004
3283 011460 011622
3284 011462 000002
3285 011464 011626
3286 011466 000377
3287 011470 004567 003350
3288 011474 000204
3289 011476 004767 003256
3290 011502 000277
3291 011504 000244
3292
3293 011506 076052
3294
3295 011510 106767 167162
3296 011514 042767 177400 167154
3297 011522 023767 000700 167146
3298 011530 001401
3299 011532 104001
3300
3301
3302
3303 011534
3304 011534 005700
3305 011536 001401
3306 011540 104002
3307
3308 011542 005701
3309 011544 001401
3310 011546 104003
3311
3312 011550 005702
3313 011552 001401
3314 011554 104004
3315
3316 011556 005703
3317 011560 001401
3318 011562 104005
3319
3320 011564 020467 167064
3321 011570 001401
3322 011572 104006
3323
3324
3325 011574 020567 167056
3326 011600 001401
3327 011602 104007
3328
3329
3330 011604 023706 000702

```

```

:*****
:TEST 37 TEST "CMPN" WITH S1L .GT. S2L, SUCCESSFUL COMPARE
:*****
TST37: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      4          ;SOURCE1 LENGTH
      S1T37    ;SOURCE1 ADDRESS
      2          ;SOURCE2 LENGTH
      S2T37    ;SOURCE2 ADDRESS
      377       ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
      JSR      R5,XPSW
      .WORD    204
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC
      CLZ          ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CMPN
      MFPS      CCODES      ;STORE RESULTANT PSW
      BIC      #177400,CCODES ;CLEAR UNUSED BITS
      CMP      @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
      ERROR    1          ;*****TEST 37 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED IN "SAVR6"
      ;ERRONEOUS SP VALUE IS A "BADR6"
      64$:
      TST      R0          ;CHECK R0=0
      BEQ      65$
      ERROR    2          ;*****TEST 37 - ERROR 2*****
      ;R0 SHOULD BE ZERO
      ;CHECK R1=0
      65$:
      TST      R1
      BEQ      66$
      ERROR    3          ;*****TEST 37 - ERROR 3*****
      ;R1 SHOULD BE ZERO
      ;CHECK R2=0
      66$:
      TST      R2
      BEQ      67$
      ERROR    4          ;*****TEST 37 - ERROR 4*****
      ;R2 SHOULD BE ZERO
      ;CHECK R3=0
      67$:
      TST      R3
      BEQ      68$
      ERROR    5          ;*****TEST 37 - ERROR 5*****
      ;R3 SHOULD BE ZERO
      ;CHECK R4 UNCHANGED
      ;BR IF OK
      68$:
      CMP      R4,DSTLN
      BEQ      69$
      ERROR    6          ;*****TEST 37 - ERROR 6*****
      ;R4 CHANGED
      ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
      ;CHECK R5 UNCHANGED
      69$:
      CMP      R5,DSTAD
      BEQ      70$
      ERROR    7          ;*****TEST 37 - ERROR 7*****
      ;R5 CHANGED
      ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
      70$:
      CMP      @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED

```

3331	011610	001403	
3332	011612	010637	000704
3333	011616	104010	

BEQ	71\$
MOV	SP,2#BADR6
ERROR	10

```

:BR IF OK
;COPY BAD SP VALUE
;*****TEST 37 - ERROR 10*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED SP IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"

```

3337	011620		
3338	011620	000403	
3339	011622		

71\$:	BR	TST40
S1T37:		

```

:BR TO NEXT TEST
;SOURCE1 STRING
;MOST SIGNIFICANT DIGIT

```

3340	011622	060	
3341	011623	060	
3342	011624	071	
3343	011625	070	

.BYTE	60
.BYTE	60
.BYTE	71
.BYTE	70

3344	011626		
3345	011626	071	
3346	011627	070	

S2T37:		
.BYTE	71	
.BYTE	70	

```

;SOURCE2 STRING
;MOST SIGNIFICANT DIGIT

```

3347			
3348			

.EVEN

```

3349
3350
3351
3352
3353
3354 011630 000004
3355 011632 004567 003074
3356 011636 000004
3357 011640 012002
3358 011642 000002
3359 011644 012006
3360 011646 000377
3361 011650 004567 003170
3362 011654 000210
3363 011656 004767 003076
3364 011662 000277
3365 011664 000250
3366
3367 011666 076052
3368
3369 011670 106767 167002
3370 011674 042767 177400 166774
3371 011702 023767 000700 166766
3372 011710 001401
3373 011712 104001
3374
3375
3376
3377 011714
3378 011714 005700
3379 011716 001401
3380 011720 104002
3381
3382 011722 005701
3383 011724 001401
3384 011726 104003
3385
3386 011730 005702
3387 011732 001401
3388 011734 104004
3389
3390 011736 005703
3391 011740 001401
3392 011742 104005
3393
3394 011744 020467 166704
3395 011750 001401
3396 011752 104006
3397
3398
3399 011754 020567 166676
3400 011760 001401
3401 011762 104007
3402
3403
3404 011764 023706 000702

```

```

*****
*TEST 40 TEST "CMPN" WITH SIL .GT. S2L, NEGATIVE SOURCE1, NON-ZERO IN EXCESS
*****
TST40: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      4              ;SOURCE1 LENGTH
      S1T40      ;SOURCE1 ADDRESS
      2              ;SOURCE2 LENGTH
      S2T40      ;SOURCE2 ADDRESS
      377          ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
      JSR      R5,XPSW
      .WORD     210
      JSR      PC,GENP      ;SET UP GENERAL REGISTERS
      SCC              ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CLN
      CMPN
      MFPS      CCODES      ;STORE RESULTANT PSW
      BIC      #177400,CCODES ;CLEAR UNUSED BITS
      CMP      3#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ     64$
      ERROR    1
      ;*****TEST 40 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED AT "SAVR6"
      ;ERRONEOUS SP VALUE IS AT "BADR6"
      54$:
      TST      R0
      BEQ     55$
      ERROR    2
      ;*****TEST 40 - ERROR 2*****
      ;R0 SHOULD BE ZERO
      ;CHECK R1=0
      55$:
      TST      R1
      BEQ     66$
      ERROR    3
      ;*****TEST 40 - ERROR 3*****
      ;R1 SHOULD BE ZERO
      ;CHECK R2=0
      56$:
      TST      R2
      BEQ     67$
      ERROR    4
      ;*****TEST 40 - ERROR 4*****
      ;R2 SHOULD BE ZERO
      ;CHECK R3=0
      67$:
      TST      R3
      BEQ     68$
      ERROR    5
      ;*****TEST 40 - ERROR 5*****
      ;R3 SHOULD BE ZERO
      ;CHECK R4 UNCHANGED
      ;BR IF OK
      ;*****TEST 40 - ERROR 6*****
      ;R4 CHANGED
      ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
      ;CHECK R5 UNCHANGED
      68$:
      CMP      R4,DSTLN
      BEQ     69$
      ERROR    6
      ;*****TEST 40 - ERROR 7*****
      ;R5 CHANGED
      ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
      ;VERIFY STACK POINTER IS RESTORED
      69$:
      CMP      3#SAVR6,SP

```

3405	011770	001403		BEQ	71\$		:BR IF OK
3406	011772	010637	000704	MOV	SP,2#BADR6		:COPY BAD SP VALUE
3407	011776	104010		ERROR	10		:*****TEST 40 - ERROR 10*****
3408							:STACK POINTER NOT RESTORED BY INSTRUCTION
3409							:EXPECTED SP IS STORED AT "SAVR6"
3410							:ERRONEOUS SP VALUE IS AT "BADR6"
3411	012000			71\$:			
3412	012000	000403		BR	TST41		:BR TO NEXT TEST
3413	012002			S1T4C:			:SOURCE1 STRING
3414	012002	061		.BYTE	61		:MOST SIGNIFICANT DIGIT
3415	012003	060		.BYTE	60		
3416	012004	063		.BYTE	63		
3417	012005	167		.BYTE	167		
3418	012006			S2T4C:			:SOURCE2 STRING
3419	012006	063		.BYTE	63		:MOST SIGNIFICANT DIGIT
3420	012007	167		.BYTE	167		
3421							
3422				.EVEN			

```

3423
3424
3425
3426
3427
3428 012010 000004
3429 012012 004567 002714
3430 012016 000004
3431 012020 012160
3432 012022 000002
3433 012024 012164
3434 012026 000377
3435 012030 004567 003010
3436 012034 000200
3437 012036 004767 002716
3438 012042 000277
3439
3440 012044 076052
3441
3442 012046 106767 166624
3443 012052 042767 177400 166616
3444 012060 023767 000700 166610
3445 012066 001401
3446 012070 104001
3447
3448
3449
3450 012072
3451 012072 005700
3452 012074 001401
3453 012076 104002
3454
3455 012100 005701
3456 012102 001401
3457 012104 104003
3458
3459 012106 005702
3460 012110 001401
3461 012112 104004
3462
3463 012114 005703
3464 012116 001401
3465 012120 104005
3466
3467 012122 020467 166526
3468 012126 001401
3469 012130 104006
3470
3471
3472 012132 020567 166520
3473 012136 001401
3474 012140 104007
3475
3476
3477 012142 023706 000702
3478 012146 001403

```

```

*****
*TEST 41 TEST "CMPN" WITH SIL .GT. S2L, POSITIVE SOURCE1, NON-ZERO IN EXCESS
*****
TST41: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T41 ;SOURCE1 ADDRESS
2 ;SOURCE2 LENGTH
S2T41 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
JSR R5,XPSW
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CMPN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 41 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 41 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$:
TST R1
BEQ 66$
ERROR 3 ;*****TEST 41 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$:
TST R2
BEQ 67$
ERROR 4 ;*****TEST 41 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$:
TST R3
BEQ 68$
ERROR 5 ;*****TEST 41 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4 UNCHANGED
68$:
CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 41 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
69$:
CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 41 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
70$:
CMP #SAVR6,SP
BEQ 71$
;VERIFY STACK POINTER IS RESTORED
;BR IF OK

```

3479	012150	010637	000704	MOV	SP,3*BADR6	:COPY BAD SP VALUE
3480	012154	104010		ERROR	10	:*****TEST 41 - ERROR 10*****
3481						:STACK POINTER NOT RESTORED BY INSTRUCTION
3482						:EXPECTED SP IS STORED AT "SAVR6"
3483						:ERRONEOUS SP VALUE IS AT "BADR6"
3484	012156			715:		
3485	012156	000403		BR	TST42	:BR TO NEXT TEST
3486	012160			S1T41:		:SOURCE1 STRING
3487	012160	061		.BYTE	61	:MOST SIGNIFICANT DIGIT
3488	012161	060		.BYTE	60	
3489	012162	063		.BYTE	63	
3490	012163	067		.BYTE	67	
3491	012164			S2T41:		:SOURCE2 STRING
3492	012164	063		.BYTE	63	:MOST SIGNIFICANT DIGIT
3493	012165	067		.BYTE	67	
3494						
3495				.EVEN		

```

3496
3497
3498
3499
3500
3501
3502 012166 000004
3503 012170 004567 002536
3504 012174 000003
3505 012176 012336
3506 012200 000003
3507 012202 012341
3508 012204 000377
3509 012206 004567 002632
3510 012212 000200
3511 012214 004767 002540
3512 012220 000277
3513
3514 012222 076052
3515
3516 012224 106767 166446
3517 012230 042767 177400 166440
3518 012236 023767 000700 166432
3519 012244 001401
3520 012246 104001
3521
3522
3523
3524 012250
3525 012250 005700
3526 012252 001401
3527 012254 104002
3528
3529 012256 005701
3530 012260 001401
3531 012262 104003
3532
3533 012264 005702
3534 012266 001401
3535 012270 104004
3536
3537 012272 005703
3538 012274 001401
3539 012276 104005
3540
3541 012300 020467 166350
3542 012304 001401
3543 012306 104006
3544
3545
3546 012310 020567 166342
3547 012314 001401
3548 012316 104007
3549
3550
3551 012320 023706 000702

```

```

*****
*TEST 42 TEST "CMPN" WITH POSITIVE OPERANDS, /S1/ .GT. /S2/
*****
TST42: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3 ;SOURCE1 LENGTH
S1T42 ;SOURCE1 ADDRESS
3 ;SOURCE2 LENGTH
S2T42 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
JSR R5,XPSW
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CMPN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 42 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR5"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$: TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 42 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$: TST R1
BEQ 66$
ERROR 3 ;*****TEST 42 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$: TST R2
BEQ 67$
ERROR 4 ;*****TEST 42 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$: TST R3
BEQ 68$
ERROR 5 ;*****TEST 42 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4 UNCHANGED
;BR IF OK
68$: CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 42 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
69$: CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 42 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
70$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED

```

# K07

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 88  
 DVKAJA.P11 20-DEC-76 15:02 T42

TEST "CMPN" WITH POSITIVE OPERANDS, /S1/ .GT. /S2/

SEQ 0090

3552	012324	001403		BEG	71\$		;BR IF OK
3553	012326	010637	000704	MOV	SP,#BADR6		;COPY BAD SP VALUE
3554	012332	104010		ERROR	10		;*****TEST 42 - ERROR 10*****
3555							;STACK POINTER NOT RESTORED BY INSTRUCTION
3556							;EXPECTED SP IS STORED AT "SAVR6"
3557							;ERRONEOUS SP VALUE IS AT "BADR6"
3558	012334			71\$:			
3559	012334	000403		BR	TST43		;BR TO NEXT TEST
3560	012336			S1T42:			;SOURCE1 STRING
3561	012336	067		.BYTE	67		;MOST SIGNIFICANT DIGIT
3562	012337	065		.BYTE	65		
3563	012340	064		.BYTE	64		
3564	012341			S2T42:			;SOURCE2 STRING
3565	012341	067		.BYTE	67		;MOST SIGNIFICANT DIGIT
3566	012342	065		.BYTE	65		
3567	012343	063		.BYTE	63		
3568							
3569				.EVEN			

```

3570
3571
3572
3573
3574
3575 012344 000004
3576 012346 004567 002360
3577 012352 000003
3578 012354 012514
3579 012356 000003
3580 012360 012517
3581 012362 000377
3582 012364 004567 002454
3583 012370 000210
3584 012372 004767 002362
3585 012376 000277
3586
3587 012400 076052
3588
3589 012402 106767 166270
3590 012406 042767 177400 166262
3591 012414 023767 000700 166254
3592 012422 001401
3593 012424 104001
3594
3595
3596
3597 012426
3598 012426 005700
3599 012430 001401
3600 012432 104002
3601
3602 012434 005701
3603 012436 001401
3604 012440 104003
3605
3606 012442 005702
3607 012444 001401
3608 012446 104004
3609
3610 012450 005703
3611 012452 001401
3612 012454 104005
3613
3614 012456 020467 166172
3615 012462 001401
3616 012464 104006
3617
3618
3619 012466 020567 166164
3620 012472 001401
3621 012474 104007
3622
3623
3624 012476 023706 000702
3625 012502 001403

```

```

;*****
;TEST 43 TEST "CMPN" WITH NEGATIVE OPERANDS, /S1/ .GT. /S2/
;*****
TST43: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3 ;SOURCE1 LENGTH
S1T43 ;SOURCE1 ADDRESS
3 ;SOURCE2 LENGTH
S2T43 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
JSR R5,XPSW
.WORD 210
JSR PC,GENP ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CMPN
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 43 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 43 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0
65$:
TST R1
BEQ 66$
ERROR 3 ;*****TEST 43 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0
66$:
TST R2
BEQ 67$
ERROR 4 ;*****TEST 43 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0
67$:
TST R3
BEQ 68$
ERROR 5 ;*****TEST 43 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4 UNCHANGED
;BR IF OK
68$:
CMP R4,DSTLN
BEQ 69$
ERROR 6 ;*****TEST 43 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
69$:
CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 43 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
70$:
CMP #SAVR6,SP
BEQ 71$

```

M07

MAIN MACY11 27(1006) 21-DEC-76 11:53 PAGE 90  
 DVKAJA.P11 20-DEC-76 15:02 \*43

TEST "CMPN" WITH NEGATIVE OPERANDS, /S1/ .GT. /S2/

SEQ 0092

3626	012504	010637	00C704	MOV	SP,2#BADR6	;COPY BAD SP VALUE
3627	012510	104010		ERROR	10	;*****TEST 43 - ERROR 10*****
3628						;STACK POINTER NOT RESTORED BY INSTRUCTION
3629						;EXPECTED SP IS STORED AT "SAVR6"
3530						;ERRONEOUS SP VALUE IS AT "BADR6"
3631	012512			71\$:		
3632	012512	000403		BR	TST44	;BR TO NEXT TEST
3633	012514			S1T43:		;SOURCE1 STRING
3634	012514	067		.BYTE	67	;MOST SIGNIFICANT DIGIT
3635	012515	065		.BYTE	65	
3636	012516	164		.BYTE	164	
3637	012517			S2*43:		;SOURCE2 STRING
3638	012517	067		.BYTE	67	;MOST SIGNIFICANT DIGIT
3639	012520	065		.BYTE	65	
3640	012521	163		.BYTE	163	
3641						
3642				.EVEN		

```

3643
3644
3645
3646
3647
3648 012522 000004
3649 012524 105777 166010
3650 012530 100535
3651 012532 026767 166012 166124
3652 012540 001007
3653 012542 032767 000001 166036
3654 012550 001403
3655 012552 005767 166016
3656 012556 001122
3657 012560
3658 012560 004567 002146
3659 012564 000004
3660 012566 013002
3661 012570 000004
3662 012572 013006
3663 012574 000377
3664 012576 012767 012656 166070
3665 012604 012777 015114 166056
3666 012612 005077 166054
3667 012616 004767 002246
3668 012622 013777 000554 166036
3669 012630 004567 002210
3670 012634 000000
3671 012636 106427 000000
3672 012642 052777 000100 166014
3673 012650 004767 002104
3674 012654 000277
3675
3676 012656 076052
3677
3678 012660 106767 166012
3679 012664 032777 000100 165772
3680 012672 001366
3681 012674 042767 177400 165774
3682 012702 023767 000700 165766
3683 012710 001401
3684 012712 104001
3685
3686
3687
3688 012714
3689 012714 005700
3690 012716 001401
3691 012720 104002
3692
3693 012722 005701
3694 012724 001401
3695 012726 104003
3696
3697 012730 005702
3698 012732 001401

```

```

*****
*TEST 44 TEST INTERRUPTABILITY OF "CMPN"
*****
TST44: SCOPE
TSTB @SWR ;TEST BIT 7 OF SWR
BMI TST45 ;SKIP TO NEXT TEST IF SET
CMP $TPS,TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T44CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT @BIT0,$ENV ;IF YES, IS PROGRAM UNDER APT?
BEQ T44CONT ;BR, IF NOT
TST $PASS ;IF YES,CHECK IF NOT ON FIRST PASS
BNE TST45 ;IF NOT, BR & SKIP TEST

T44CONT:
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T44 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T44 ;SOURCE2 ADDRESS
377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
MOV #CMPNPC,PCI ;STORE PC OF TEST INSTRUCTION
MOV @INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
JSR PC,TDONE ;WAIT FOR SLU READY
MOV @#NULL,@TBUF ;SEND NULL CHARACTER
JSR R5,XPSW ;STORE EXPECTED PSW
.WORD 0
MTPS #0 ;ALLOW INTERRUPTS
BIS #100,@TCSR ;ENABLE TTY INTERRUPTS
RECMPN: JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S

CMPNPC: CMPN

MFPS CCODES ;STORE RESULTANT PSW
BIT #100,@TCSR ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
BNE RECMPN ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 44 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BADR6"

64$: TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 44 - ERROR 2*****
;R0 SHOULD BE ZERO
65$: TST R1 ;CHECK R1=0
BEQ 66$
ERROR 3 ;*****TEST 44 - ERROR 3*****
;R1 SHOULD BE ZERO
66$: TST R2 ;CHECK R2=0
BEQ 67$

```

```

3699 012734 104004          ERROR 4          ;*****TEST 44 - ERROR 4*****
3700                                     ;R2 SHOULD BE ZERO
3701 012736 005703          67$:  TST      R3          ;CHECK R3=0
3702 012740 001401          BEQ      68$
3703 012742 104005          ERROR 5          ;*****TEST 44 - ERROR 5*****
3704                                     ;R3 SHOULD BE ZERO
3705 012744 020467 165704          69$:  CMP      R4,DSTLN     ;CHECK R4 UNCHANGED
3706 012750 001401          BEQ      69$
3707 012752 104006          ERROR 6          ;BR IF OK
3708                                     ;*****TEST 44 - ERROR 6*****
3709                                     ;R4 CHANGED
3710 012754 020567 165676          69$:  CMP      R5,DSTAD     ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
3711 012760 001401          BEQ      70$
3712 012762 104007          ERROR 7          ;CHECK R5 UNCHANGED
3713                                     ;*****TEST 44 - ERROR 7*****
3714                                     ;R5 CHANGED
3715 012764 023706 000702          70$:  CMP      @SAVR6,SP     ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
3716 012770 001403          BEQ      71$
3717 012772 010637 000704          MOV      SP,@BADR6
3718 012776 104010          ERROR 10         ;VERIFY STACK POINTER IS RESTORED
3719                                     ;BR IF OK
3720                                     ;COPY BAD SP VALUE
3721                                     ;*****TEST 44 - ERROR 10*****
3722 013000                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
3723 013000 000404          71$:  BR      ENDT44       ;EXPECTED SP IS STORED AT "SAVR6"
3724 013002                                     ;ERRONEOUS SP VALUE IS AT "BADR6"
3725 013002 071          S1T44: .BYTE 71
3726 013003 070          .BYTE 70
3727 013004 067          .BYTE 67
3728 013005 066          .BYTE 66
3729 013006          S2T44:
3730 013006 071          .BYTE 71
3731 013007 070          .BYTE 70
3732 013010 067          .BYTE 67
3733 013011 065          .BYTE 65
3734
3735 .EVEN
3736 013012 016777 165654 165650 ENDT44: MOV      TPSW,@TVECT
3737 013020 106427 000200          MTPS   #200

```

```

3738
3739
3740
3741
3742
3743
3744 013024 000004
3745 013026 004567 001700
3746 013032 000012
3747 013034 013164
3748 013036 177777
3749 013040 177777
3750 013042 000377
3751 013044 004567 001774
3752 013050 000200
3753 013052 004767 001702
3754 013056 000277
3755
3756 013060 076053
3757
3758 013062 106767 165610
3759 013066 042767 177400 165602
3760 013074 023767 000700 165574
3761 013102 001401
3762 013104 104001
3763
3764
3765
3766 013106
3767 013106 005700
3768 013110 001401
3769 013112 104002
3770
3771 013114 005701
3772 013116 001401
3773 013120 104003
3774
3775 013122 020227 077777
3776 013126 001401
3777 013130 104004
3778
3779
3780
3781 013132 020327 177777
3782 013136 001401
3783 013140 104005
3784
3785
3786
3787 013142 020467 165506
3788 013146 001401
3789 013150 104006
3790
3791
3792 013152 020567 165500
3793 013156 001401

```

```

:*****
:*TEST 45 TEST "CVTNL" WITH SRC = +2,147,483,647
:*****
TST45: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      12      ;SOURCE LENGTH
      ST45     ;SOURCE ADDRESS
      177777  ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
      177777
      377     ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
      JSR      R5,XPSW
      .WORD   200
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC      ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CVTNL
      MFPS     CCODES      ;STORE RESULTANT PSW
      BIC     #177400,CCODES ;CLEAR UNUSED BITS
      CMP     @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ     64$
      ERROR   1            ;*****TEST 45 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED AT "SAVR5"
      ;ERRONEOUS SP VALUE IS AT "BADR6"
64$:
      TST     R0            ;CHECK R0=0
      BEQ     65$
      ERROR   2            ;*****TEST 45 - ERROR 2*****
      ;R0 SHOULD BE ZERO
      ;CHECK R1=0
65$:
      TST     R1
      BEQ     66$
      ERROR   3            ;*****TEST 45 - ERROR 3*****
      ;R1 SHOULD BE ZERO
      ;CHECK UPPER WORD OF ANSWER
      ;BR IF OK
      ;*****TEST 45 - ERROR 4*****
      ;UPPER WORD OF ANSWER IS IN ERROR
      ;EXPECTED VALUE IS 077777
      ;ERRONEOUS ANSWER VALUE IS IN R2
      ;CHECK LOWER WORD OF ANSWER
      ;BR IF OK
      ;*****TEST 45 - ERROR 5*****
      ;LOWER WORD OF ANSWER IS IN ERROR
      ;EXPECTED VALUE IS 177777
      ;ERRONEOUS ANSWER IS IN R3
      ;CHECK R4 UNCHANGED
66$:
      CMP     R2,#077777
      BEQ     67$
      ERROR   4
67$:
      CMP     R3,#177777
      BEQ     68$
      ERROR   5
      ;*****TEST 45 - ERROR 6*****
      ;R4 CHANGED
      ;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
      ;CHECK R5 UNCHANGED
      ;BR IF OK
68$:
      CMP     R4,DSTLN
      BEQ     69$
      ERROR   6
69$:
      CMP     R5,DSTAD
      BEQ     70$

```

3794 013160 104007  
 3795  
 3796  
 3797 013162  
 3798 013162 000405  
 3799 013164  
 3800 013164 062  
 3801 013165 061  
 3802 013166 064  
 3803 013167 067  
 3804 013170 064  
 3805 013171 070  
 3806 013172 063  
 3807 013173 066  
 3808 013174 064  
 3809 013175 067  
 3810

705:

ST45:

ERROR 7  
 BR TST46  
 .BYTE 62  
 .BYTE 61  
 .BYTE 64  
 .BYTE 67  
 .BYTE 64  
 .BYTE 70  
 .BYTE 63  
 .BYTE 65  
 .BYTE 64  
 .BYTE 67

:\*\*\*\*\*TEST 45 - ERROR 7\*\*\*\*\*  
 ;RS CHANGED  
 ;RS SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"  
 :BR TO NEXT TEST  
 ;SOURCE STRING  
 ;MOST SIGNIFICANT DIGIT

```

3811
3812
3813
3814
3815
3816
3817 013176 000004
3818 013200 004567 001526
3819 013204 000012
3820 013206 013340
3821 013210 177777
3822 013212 177777
3823 013214 000377
3824 013216 004567 001622
3825 013222 000202
3826 013224 004767 001530
3827 013230 000277
3828 013232 000242
3829
3830 013234 076053
3831
3832 013236 106767 165434
3833 013242 042767 177400 165426
3834 013250 023767 000700 165420
3835 013256 001401
3836 013260 104001
3837
3838
3839
3840 013262
3841 013262 005700
3842 013264 001401
3843 013266 104002
3844
3845 013270 005701
3846 013272 001401
3847 013274 104003
3848
3849 013276 020227 014631
3850 013302 001401
3851 013304 104004
3852
3853
3854
3855 013306 020327 031462
3856 013312 001401
3857 013314 104005
3858
3859
3860
3861 013316 020467 165332
3862 013322 001401
3863 013324 104006
3864
3865
3866 013326 020567 165324

```

```

*****
;TEST 46 TEST "CVTNL" WITH SRC= +2,687,483,648, OVERFLOW WITH CORRECT SIGN
*****
†ST46: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      12          ;SOURCE LENGTH
      ST46       ;SOURCE ADDRESS
      177777     ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
      177777
      377        ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
      JSR      R5,XPSW
      .WORD    202
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC
      CLV        ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CVTNL
      MFPS      CCODES      ;STORE RESULTANT PSW
      BIC      #177400,CCODES ;CLEAR UNUSED BITS
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
      ERROR    1           ;*****TEST 46 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "SAVR6"
                          ;ERRONEOUS SP VALUE IS AT "BADR6"
      54$:
      TST      R0          ;CHECK R0=0
      BEQ      65$
      ERROR    2           ;*****TEST 46 - ERROR 2*****
                          ;R0 SHOULD BE ZERO
                          ;CHECK R1=0
      65$:
      TST      R1
      BEQ      66$
      ERROR    3           ;*****TEST 46 - ERROR 3*****
                          ;R1 SHOULD BE ZERO
                          ;CHECK UPPER WORD OF ANSWER
      66$:
      CMP      R2,#014631 ;CHECK UPPER WORD OF ANSWER
      BEQ      67$
      ERROR    4           ;*****TEST 46 - ERROR 4*****
                          ;UPPER WORD OF ANSWER IS IN ERROR
                          ;EXPECTED VALUE IS 014631
                          ;ERRONEOUS ANSWER VALUE IS IN R2
      67$:
      CMP      R3,#031462 ;CHECK LOWER WORD OF ANSWER
      BEQ      68$
      ERROR    5           ;*****TEST 46 - ERROR 5*****
                          ;LOWER WORD OF ANSWER IS IN ERROR
                          ;EXPECTED VALUE IS 031462
                          ;ERRONEOUS ANSWER IS IN R3
      68$:
      CMP      R4,DSTLN   ;CHECK R4 UNCHANGED
      BEQ      69$
      ERROR    6           ;*****TEST 46 - ERROR 6*****
                          ;R4 CHANGED
                          ;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
      69$:
      CMP      R5,DSTAD   ;CHECK R5 UNCHANGED

```

F08

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 96  
CVKKAJA.P11 20-DEC-76 15:02 T46

TEST "CVTNL" WITH SRC= +2.687.483.648, OVERFLOW WITH CORRECT SIGN

SEQ 0098

3867 013332 001401  
3868 013334 104007  
3869  
3870  
3871 013336  
3872 013336 000405  
3873 013340  
3874 013340 064  
3875 013341 062  
3876 013342 071  
3877 013343 064  
3878 013344 071  
3879 013345 066  
3880 013346 067  
3881 013347 062  
3882 013350 071  
3883 013351 066

BEQ 70\$  
ERROR 7

:BR IF OK  
:\*\*\*\*\*TEST 46 - ERROR 7\*\*\*\*\*  
:R5 CHANGED  
:R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"

70\$:

BR TST47

:BR TO NEXT TEST  
:SOURCE STRING  
:MOST SIGNIFICANT DIGIT

ST46:

.BYTE 64  
.BYTE 62  
.BYTE 71  
.BYTE 64  
.BYTE 71  
.BYTE 56  
.BYTE 67  
.BYTE 62  
.BYTE 71  
.BYTE 56

```

3885
3886
3887
3888
3889
3890 013352 000004
3891 013354 004567 001352
3892 013360 000012
3893 013362 013514
3894 013364 177777
3895 013366 177777
3896 013370 000377
3897 013372 004567 001446
3898 013376 000212
3899 013400 004767 001354
3900 013404 000277
3901 013406 000242
3902
3903 013410 076053
3904
3905 013412 106767 165260
3906 013416 042767 177400 165252
3907 013424 023767 000700 165244
3908 013432 001401
3909 013434 104001
3910
3911
3912
3913 013436
3914 013436 005700
3915 013440 001401
3916 013442 104002
3917
3918 013444 005701
3919 013446 001401
3920 013450 104003
3921
3922 013452 020227 100000
3923 013456 001401
3924 013460 104004
3925
3926
3927
3928 013462 020327 000000
3929 013466 001401
3930 013470 104005
3931
3932
3933
3934 013472 020467 165156
3935 013476 001401
3936 013500 104006
3937
3938
3939 013502 020567 165150
3940 013506 001401

```

```

;*****
;TEST 47 TEST "CVTNL" WITH SRC = +2,147,483,648, OVERFLOW
;*****
†ST47: SCOPE
      JSR    R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      L     12             ;SOURCE LENGTH
      ST47             ;SOURCE ADDRESS
      L     177777        ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
      L     177777
      L     377           ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
      JSR    R5,XPSW
      .WORD 212
      JSR    PC,GENR      ;SET UP GENERAL REGISTERS
      SCC
      CLV                ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CVTNL
      MFPS   CCODES      ;STORE RESULTANT PSW
      BIC   #177400,CCODES ;CLEAR UNUSED BITS
      CMP   @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ   64$          ;BR, IF EQUAL
      ERROR 1            ;*****TEST 47 - ERROR 1*****
                        ;PSW ERROR
                        ;EXPECTED PSW IS STORED AT "SAVR5"
                        ;ERRONEOUS SP VALJE IS AT "BADR6"
64$:
      TST   R0            ;CHECK R0=0
      BEQ   65$
      ERROR 2            ;*****TEST 47 - ERROR 2*****
                        ;R0 SHOULD BE ZERO
                        ;CHECK R1=0
65$:
      TST   R1
      BEQ   66$
      ERROR 3            ;*****TEST 47 - ERROR 3*****
                        ;R1 SHOULD BE ZERO
                        ;CHECK UPPER WORD OF ANSWER
66$:
      CMP   R2,#100000
      BEQ   67$
      ERROR 4            ;*****TEST 47 - ERROR 4*****
                        ;UPPER WORD OF ANSWER IS IN ERROR
                        ;EXPECTED VALUE IS 100000
                        ;ERRONEOUS ANSWER VALUE IS IN R2
67$:
      CMP   R3,#0        ;CHECK LOWER WORD OF ANSWER
      BEQ   68$
      ERROR 5            ;*****TEST 47 - ERROR 5*****
                        ;LOWER WORD OF ANSWER IS IN ERROR
                        ;EXPECTED VALUE IS 0
                        ;ERRONEOUS ANSWER IS IN R3
68$:
      CMP   R4,DSTLN
      BEQ   69$
      ERROR 6            ;*****TEST 47 - ERROR 6*****
                        ;R4 CHANGED
                        ;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
69$:
      CMP   R5,DSTAD
      BEQ   70$
      ;CHECK R5 UNCHANGED
      ;BR IF OK

```

# H08

MAIN. MACY11 27(1036) 21-DEC-76 11:53 PAGE 98  
CVKAJA.P11 20-DEC-76 15:02 T47

TEST "CVTNL" WITH SRC = +2,147,483,648. OVERFLOW

SEG 0100

3941 013510 104007  
3942  
3943  
3944 013512  
3945 013512 000405  
3946 013514  
3947 013514 062  
3948 013515 061  
3949 013516 064  
3950 013517 067  
3951 013520 064  
3952 013521 070  
3953 013522 063  
3954 013523 066  
3955 013524 064  
3956 013525 070  
3957

70\$:  
ST47:

ERROR 7  
BR TST50  
.BYTE 62  
.BYTE 61  
.BYTE 64  
.BYTE 67  
.BYTE 64  
.BYTE 70  
.BYTE 63  
.BYTE 66  
.BYTE 64  
.BYTE 70

;\*\*\*\*\*TEST 47 - ERROR 7\*\*\*\*\*  
;RS CHANGED  
;RS SHOULD STILL EQUAL THE CONTENTS OF "DETAC"  
;BR TO NEXT TEST  
;SOURCE STRING  
;MOST SIGNIFICANT DIGIT

```

3958
3959
3960
3961
3962
3963 013526 000304
3964 013530 004567 001176
3965 013534 000012
3966 013536 013670
3967 013540 177777
3968 013542 177777
3969 013544 000377
3970 013546 004567 001272
3971 013552 000210
3972 013554 004767 001200
3973 013560 000277
3974 013562 000250
3975
3976 013564 076053
3977
3978 013566 106767 165104
3979 013572 042767 177400 165076
3980 013600 023767 000700 165070
3981 013606 001401
3982 013610 104001
3983
3984
3985
3986 013612
3987 013612 005700
3988 013614 001401
3989 013616 104002
3990
3991 013620 005701
3992 013622 001401
3993 013624 104003
3994
3995 013626 020227 100000
3996 013632 001401
3997 013634 104004
3998
3999
4000
4001 013636 020327 000000
4002 013642 001401
4003 013644 104005
4004
4005
4006
4007 013646 020467 165302
4008 013652 001401
4009 013654 104006
4010
4011
4012 013656 020567 164774
4013 013662 001401

```

```

:*****
:TEST 50 TEST "CVTNL" WITH SRC = -2,147,483,648
:*****
TST50: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      .WORD   12            ;SOURCE LENGTH
      JSR      ST50         ;SOURCE ADDRESS
      JSR      177777       ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
      JSR      177777       ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
      JSR      R5,XPSW
      JSR      210
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC
      CLN                  ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      CVTNL
      MFPS      C00DES      ;STORE RESULTANT PSW
      BIC      #177400,C00DES ;CLEAR UNUSED BITS
      CMP      0#EXPPSW,C00DES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$
      ERROR    1            ;BR, IF EQUAL
                          ;*****TEST 50 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "SAVR6"
                          ;ERRONEOUS SP VALUE IS AT "BADR6"
64$:  TST      R0            ;CHECK R0=0
      BEQ      65$
      ERROR    2            ;*****TEST 50 - ERROR 2*****
                          ;R0 SHOULD BE ZERO
                          ;CHECK R1=0
65$:  TST      R1
      BEQ      66$
      ERROR    3            ;*****TEST 50 - ERROR 3*****
                          ;R1 SHOULD BE ZERO
                          ;CHECK UPPER WORD OF ANSWER
66$:  CMP      R2,#100000   ;CHECK UPPER WORD OF ANSWER
      BEQ      67$
      ERROR    4            ;BR IF OK
                          ;*****TEST 50 - ERROR 4*****
                          ;UPPER WORD OF ANSWER IS IN ERROR
                          ;EXPECTED VALUE IS 100000
                          ;ERRONEOUS ANSWER VALUE IS IN R2
67$:  CMP      R3,#0       ;CHECK LOWER WORD OF ANSWER
      BEQ      68$
      ERROR    5            ;BR IF OK
                          ;*****TEST 50 - ERROR 5*****
                          ;LOWER WORD OF ANSWER IS IN ERROR
                          ;EXPECTED VALUE IS 0
                          ;ERRONEOUS ANSWER IS IN R3
68$:  CMP      R4,DSTLN    ;CHECK R4 UNCHANGED
      BEQ      69$
      ERROR    6            ;*****TEST 50 - ERROR 6*****
                          ;R4 CHANGED
                          ;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
69$:  CMP      R5,DSTAD    ;CHECK R5 UNCHANGED
      BEQ      70$
      ERROR    7            ;BR IF OK

```

J08

MAIN MACY11 27110061 21-DEC-76 11:53 PAGE 100  
CVKASA.P11 20-DEC-76 15:02 150

TEST "CVTNL" WITH SRC = -2,147,483,648

SEG 0102

4014	013664	104007
4015		
4016		
4017	013666	
4018	013666	000405
4019	013670	
4020	013670	062
4021	013671	061
4022	013672	064
4023	013673	067
4024	013674	064
4025	013675	070
4026	013675	063
4027	013677	056
4028	013700	054
4029	013701	170
030		

70\$:

ST50:

```

ERROR 7
BR TST51
.BYTE 62
.BYTE 61
.BYTE 64
.BYTE 67
.BYTE 64
.BYTE 70
.BYTE 63
.BYTE 64
.BYTE 170

```

```

:*****TEST 50 - ERROR 7*****
:R5 CHANGED
:R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAC"
:BR TO NEXT TEST
:SOURCE STRING
:MOST SIGNIFICANT DIGIT

```

K08

```

4031
4032
4033
4034
4035
4036 013702 000004
4037 013704 004567 001022
4038 013710 000012
4039 013712 014044
4040 013714 177777
4041 013716 177777
4042 013720 000377
4043 013722 004567 001116
4044 013726 000202
4045 013730 004767 001024
4046 013734 000265
4047 013736 000252
4048
4049 013740 076053
4050
4051 013742 106767 164730
4052 013746 042767 177400 164722
4053 013754 023767 000700 164714
4054 013762 001401
4055 013764 104001
4056
4057
4058
4059 013766
4060 013766 005700
4061 013770 001401
4062 013772 104002
4063
4064 013774 005701
4065 013776 001401
4066 014000 104003
4067
4068 014002 020227 077777
4069 014006 001401
4070 014010 104004
4071
4072
4073
4074 014012 020327 177777
4075 014016 001401
4076 014020 104005
4077
4078
4079
4080 014022 020467 164626
4081 014026 001401
4082 014030 104006
4083
4084
4085 014032 020567 164620
4086 014036 001401

```

```

*****
*TEST 51 TEST "CVTNL" WITH SRC = -2,147,483,649, OVERFLOW
*****
†ST51: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      12          ;SOURCE LENGTH
      ST51       ;SOURCE ADDRESS
      177777     ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
      177777     ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
      377
      JSR      R5,XPSW
      .WORD    202
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      +SEZ!SEC          ;SET UP THE COMPLEMENT OF EXPECTED CC'S
      +CLN!CLV
      CVTNL
      MFPS     CCODES      ;STORE RESULTANT PSW
      BIC     #177400,CCODES ;CLEAR UNUSED BITS
      CMP     @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALJE
      BEQ     64$          ;BR, IF EQUAL
      ERROR   1          ;*****TEST 51 - ERROR 1*****
                        ;PSW ERROR
                        ;EXPECTED PSW IS STORED AT "SAVR6"
                        ;ERRONEOUS SP VALUE IS AT "BADR6"
      64$:
      TST     R0          ;CHECK R0=0
      BEQ     65$
      ERROR   2          ;*****TEST 51 - ERROR 2*****
                        ;R0 SHOULD BE ZERO
                        ;CHECK R1=0
      65$:
      TST     R1
      BEQ     66$
      ERROR   3          ;*****TEST 51 - ERROR 3*****
                        ;R1 SHOULD BE ZERO
                        ;CHECK UPPER WORD OF ANSWER
      66$:
      CMP     R2,#077777 ;CHECK UPPER WORD OF ANSWER
      BEQ     67$
      ERROR   4          ;BR IF OK
                        ;*****TEST 51 - ERROR 4*****
                        ;UPPER WORD OF ANSWER IS IN ERROR
                        ;EXPECTED VALUE IS 077777
                        ;ERRONEOUS ANSWER VALUE IS IN R2
      67$:
      CMP     R3,#177777 ;CHECK LOWER WORD OF ANSWER
      BEQ     68$
      ERROR   5          ;BR IF OK
                        ;*****TEST 51 - ERROR 5*****
                        ;LOWER WORD OF ANSWER IS IN ERROR
                        ;EXPECTED VALUE IS 177777
                        ;ERRONEOUS ANSWER IS IN R3
      68$:
      CMP     R4,DSTLN   ;CHECK R4 UNCHANGED
      BEQ     69$
      ERROR   6          ;*****TEST 51 - ERROR 6*****
                        ;R4 CHANGED
                        ;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
      69$:
      CMP     R5,DSTAD   ;CHECK R5 UNCHANGED
      BEQ     70$
      ERROR   7

```

L08

MAIN MACY11 27(1005) 21-DEC-76 11:53 PAGE 102  
OVKAJA.P:1 20-DEC-76 15:02 T51

TEST "CVTNL" WITH SRC = -2,147,483,649, OVERFLOW

SEQ 0104

4087	014040	104007
4088		
4089		
4090	014042	
4091	014042	000405
4092	014044	
4093	014044	062
4094	014045	061
4095	014046	064
4096	014047	067
4097	014050	064
4098	014051	070
4099	014052	063
4100	014053	066
4101	014054	064
4102	014055	171
4103		

70\$:  
ST51:

ERROR	7
BR	T5152
.BYTE	62
.BYTE	61
.BYTE	64
.BYTE	67
.BYTE	64
.BYTE	70
.BYTE	63
.BYTE	66
.BYTE	64
.BYTE	171

:\*\*\*\*\*TEST 51 - ERROR 7\*\*\*\*\*  
 :RS CHANGED  
 :RS SHOULD STILL EQUAL THE CONTENTS OF "DSTAC"  
 :BR TO NEXT TEST  
 :SOURCE STRING  
 :MOST SIGNIFICANT DIGIT

:

# MO8

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 103  
 DVKAJA.P11 20-DEC-76 15:02 TS1

TEST "CVTNL" WITH SRC = -2,147,483,649, OVERFLOW

SEG C105

```

4104
4105
4106
4107
4108
4109 014056 000004
4110 014060 004567 000646
4111 014064 000002
4112 014066 014220
4113 014070 177777
4114 014072 177777
4115 014074 000377
4116 014076 004567 000742
4117 014102 000204
4118 014104 004767 000650
4119 014110 000277
4120 014112 000244
4121
4122 014114 076053
4123
4124 014116 106767 164554
4125 014122 042767 177400 164546
4126 014130 023767 000700 164540
4127 014136 001401
4128 014140 104001
4129
4130
4131
4132 014142
4133 014142 005700
4134 014144 001401
4135 014146 104002
4136
4137 014150 005701
4138 014152 001401
4139 014154 104003
4140
4141 014156 020227 000000
4142 014162 001401
4143 014164 104004
4144
4145
4146
4147 014166 020327 000000
4148 014172 001401
4149 014174 104005
4150
4151
4152
4153 014176 020467 164452
4154 014202 001401
4155 014204 104006
4156
4157
4158 014206 020567 164444
4159 014212 001401

```

```

;*****
;TEST 52 TEST "CVTNL" WITH SRC LENGTH = 1, SOURCE= 50,60
;*****
†ST52: SCOPE
      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
      2
      ST52
      177777
      177777
      377
      JSR      R5,XPSW      ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
      .WORD   204
      JSR      PC,GENR     ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
      SCC
      CLZ
      CVTNL
      MFPS     CCODES      ;STORE RESULTANT PSW
      BIC     #177400,CCODES ;CLEAR UNUSED BITS
      CMP     @#EXPPSW.CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ     64$
      ERROR   1
      ;*****TEST 52 - ERROR 1*****
      PSW ERROR
      EXPECTED PSW IS STORED AT "SAVR6"
      ERRONEOUS SP VALUE IS AT "BADR6"
64$:
      TST     R0
      BEQ     65$
      ERROR   2
      ;*****TEST 52 - ERROR 2*****
      ;R0 SHOULD BE ZERO
      ;CHECK R1=0
65$:
      TST     R1
      BEQ     66$
      ERROR   3
      ;*****TEST 52 - ERROR 3*****
      ;R1 SHOULD BE ZERO
      ;CHECK UPPER WORD OF ANSWER
      BEQ     67$
      ERROR   4
      ;*****TEST 52 - ERROR 4*****
      ;UPPER WORD OF ANSWER IS IN ERROR
      ;EXPECTED VALUE IS 0
      ;ERRONEOUS ANSWER VALUE IS IN R2
67$:
      CMP     R3,#0
      BEQ     68$
      ERROR   5
      ;*****TEST 52 - ERROR 5*****
      ;LOWER WORD OF ANSWER IS IN ERROR
      ;EXPECTED VALUE IS 0
      ;ERRONEOUS ANSWER IS IN R3
      ;CHECK R4 UNCHANGED
68$:
      CMP     R4,DSTLN
      BEQ     69$
      ERROR   6
      ;*****TEST 52 - ERROR 6*****
      ;R4 CHANGED
      ;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
      ;CHECK R5 UNCHANGED
      BEQ     70$
      ;BR IF OK

```

# N08

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 104  
CVKAJA.P11 20-DEC-76 15:02 T52 TEST "CVTNL" WITH SRC LENGTH = 1. SOURCE= 60.60

SEG 3106

4160	014214	104007	ERROR	7	:*****TEST 52 - ERROR 7*****
4161					:R5 CHANGED
4162					:R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAC"
4163	014216		70\$:		
4164	014216	000401	BR	TST53	:BR TO NEXT TEST
4165	014220		ST52:		:SOURCE STRING
4166	014220	060	.BYTE	60	:MOST SIGNIFICANT DIGIT
4167	014221	060	.BYTE	60	
4168					

```

4169
4170
4171
4172
4173 014222 000004
4174 014224 105777 164310
4175 014230 100532
4176 014232 026767 164312 164424
4177 014240 001007
4178 014242 032767 000001 164336
4179 014250 001403
4180 014252 005767 164316
4181 014256 001117
4182 014260
4183 014260 004567 000446
4184 014264 000012
4185 014266 014472
4186 014270 177777
4187 014272 177777
4188 014274 000377
4189 014276 012767 014356 164370
4190 014304 012777 015114 164356
4191 014312 005077 164354
4192 014316 004767 000546
4193 014322 013777 000554 164336
4194 014330 004567 000510
4195 014334 000000
4196 014336 106427 000000
4197 014342 052777 000100 164314
4198 014350 004767 000404
4199 014354 000277
4200
4201 014356 076053
4202
4203 014360 106767 164312
4204 014364 032777 000100 164272
4205 014372 001366
4206 014374 042767 177400 164274
4207 014402 023767 000700 164266
4208 014410 001401
4209 014412 104001
4210
4211
4212
4213 014414
4214 014414 005700
4215 014416 001401
4216 014420 104002
4217
4218 014422 005701
4219 014424 001401
4220 014426 104003
4221
4222 014430 020227 052525
4223 014434 001401
4224 014436 104004

```

```

;*****
;*TEST 53 TEST INTERRUPTABILITY OF "CVTNL"
;*****
TST53: SCOPE
TSTB @SWR ;TEST BIT 7 OF SWR
BMI @EOP ;SKIP TO NEXT TEST IF SET
CMP @TPS,@TCR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE CVTCONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT @BITO,@ENV ;IF YES, IS PROGRAM UNDER APT?
BEQ CVTCONT ;BR, IF NOT
TST @PASS ;IF YES,CHECK IF NOT ON FIRST PASS
BNE @EOP ;IF NOT, BR & SKIP TEST

CVTCONT: JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
12 ;SOURCE LENGTH
ST53 ;SOURCE ADDRESS
177777 ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
177777
377 ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
MOV @CVTPC,@PCI ;STORE PC OF TEST INSTRUCTION
MOV @INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
JSR PC,@DONE ;WAIT FOR SLU READY
MOV @@NULL,@TBUF ;SEND NULL CHARACTER
JSR R5,@PSW ;STORE EXPECTED PSW
.WORD 0
MTPS @0 ;ALLOW INTERRUPTS
BIS @100,@TCR ;ENABLE TTY INTERRUPTS
RECVTN: JSR PC,@GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S

CVTPC: CVTNL

MFPS CCODES ;STORE RESULTANT PSW
BIT @100,@TCR ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
BNE RECVTN ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
BIC @177400,CCODES ;CLEAR UNUSED BITS
CMP @@EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 53 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "SAVR6"
;ERRONEOUS SP VALUE IS AT "BACR6"

64$: TST R0 ;CHECK R0=0
BEQ 65$
ERROR 2 ;*****TEST 53 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0

65$: TST R1
BEQ 66$
ERROR 3 ;*****TEST 53 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK UPPER WORD OF ANSWER
;BR IF OK

66$: CMP R2,@052525
BEQ 67$
ERROR 4 ;*****TEST 53 - ERROR 4*****

```

```

4225
4226
4227
4228 014440 020327 125252      67$:  CMP      R3,#125252
4229 014444 001401              BEQ      68$
4230 014446 104005              ERROR   5
4231
4232
4233
4234 014450 020467 164200      68$:  CMP      R4,DSTLN
4235 014454 001401              BEQ      69$
4236 014456 104006              ERROR   6
4237
4238
4239 014460 020567 164172      69$:  CMP      R5,DSTAD
4240 014464 001401              BEQ      70$
4241 014466 104007              ERROR   7
4242
4243
4244 014470              70$:
4245 014470 000405              BR      ENDT53
4246 014472              ST53:
4247 014472 061              .BYTE  61
4248 014473 064              .BYTE  64
4249 014474 063              .BYTE  63
4250 014475 061              .BYTE  61
4251 014476 066              .BYTE  66
4252 014477 067              .BYTE  67
4253 014500 067              .BYTE  67
4254 014501 066              .BYTE  66
4255 014502 061              .BYTE  61
4256 014503 060              .BYTE  60
4257
4258 014504 016777 164162 164156 ENDT53: MOV     TFSW,@TVECT
4259 014512 106427 000200              MTPS   #200
4260

```

```

;UPPER WORD OF ANSWER IS IN ERROR
;EXPECTED VALUE IS 052525
;ERRONEOUS ANSWER VALUE IS IN R2
;CHECK LOWER WORD OF ANSWER
;BR IF OK
;*****TEST 53 - ERROR 5*****
;LOWER WORD OF ANSWER IS IN ERROR
;EXPECTED VALUE IS 125252
;ERRONEOUS ANSWER IS IN R3
;CHECK R4 UNCHANGED
;*****TEST 53 - ERROR 6*****
;R4 CHANGED
;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
;CHECK R5 UNCHANGED
;BR IF OK
;*****TEST 53 - ERROR 7*****
;R5 CHANGED
;R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"
;BR TO END OF THIS TEST
;SOURCE STRING
;MOST SIGNIFICANT DIGIT

```

```

4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271 014516
4272 014516 000004
4273 014520 005067 163756
4274 014524 005267 164044
4275 014530 042767 100000 164036
4276 014536 005327
4277 014540 000001
4278 014542 003024
4279 014544 012737
4280 014546 000001
4281 014550 014540
4282 014552 104401 C14664
4283 014556 013700 000042
4284 014562 001414
4285 014564 005046
4286 014566 012746 014574
4287 014572 000426
4288
4289 014574
4290 014574 013700 000042
4291 014600 001405
4292 014602 000005
4293 014604 004710
4294 014606 000240
4295 014610 000240
4296 014612 000240
4297 014614
4298 014614 104400
4299 014616 042716 000020
4300 014622 032777 010000 163710
4301 014630 001005
4302 014632 005167 000020
4303 014636 100402
4304 014640 052716 000020
4305 014644 012746 014652
4306 014650 000002
4307
4308
4309 014652
4310 014652 000137
4311 014654 001300
4312 014656 000000
4313 014660 377 377 000
4314 014664
4315 014664 005016 047105 020104
4316 014672 040520 051523 020040

.SBTTL END OF PASS ROUTINE
;*****
;*INCREMENT THE PASS NUMBER ($PASS)
;*IF SW12=1 INHIBIT TRACE TRAP
;*IF THERES A MONITOR GO TO IT
;*IF THERE ISN'T JUMP TO BEGIN

$EOP:
SCOPE
CLR $STNM ;;ZERO THE TEST NUMBER
INC $PASS ;;INCREMENT THE PASS NUMBER
BIC #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
DEC (PC)+ ;;LOOP?
$EOPCT: .WORD 1
BGT $DOAGN ;;YES
MOV (PC)+,2(PC)+ ;;RESTORE COUNTER
$ENDCT: .WORD 1
$EOPCT
TYPE, ENDMMSG ;TYPE "END PASS"
$GET+2: MOV 2#42,RO ;;GET MONITOR ADDRESS
BEQ $DOAGN ;;BRANCH IF NO MONITOR
CLR -(SP) ;;INSURE THE "T" BIT IS CLEAR
MOV #SCLR.T,-(SP) ;;SETUP FOR AN RTI OR RTT
BR $RTN ;;GO DO AN RTI OR RTT TO LOAD THE PSW
;;WITH A CLEARED "T" BIT

$CLR.T:
MOV 2#42,RO ;;INSURE RO CONTAINS THE MONITORS
BEQ $DOAGN ;;RETURN ADDRESS
RESET ;;CLEAR THE WORLD
$ENDAD: JSR PC,(RO) ;;GO TO MONITOR
NOP ;;SAVE ROOM
NOP ;;FOR
NOP ;;ACT11

$DOAGN:
TRAP ;;PUSH OLD PSW AND PC ON STACK
BIC #20,(SP) ;;CLEAR THE "T" BIT
BIT #BIT12,2SWR ;;RUN WITH TRACE TRAP?
BNE 1$ ;;BR IF NO
COM $TBIT ;;IS IT TIME FOR TRACE TRAP
BMI 1$ ;;BR IF NO
BIS #20,(SP) ;;SET TRACE TRAP
1$: MOV #SLOOP,-(SP) ;;JUMP TO START OF TEST
$RTN: RTI ;;RETURN--THIS IS CHANGED TO
;;AN "RTT" IF "RTT" IS A LEGAL
;;INSTRUCTION

$LOOP:
JMP 2(PC)+ ;;RETURN
$RTNAD: .WORD BEGIN
$TBIT: .WORD 0 ;;"T" BIT STATE INDICATOR
$ENULL: .BYTE -1,-1,0 ;;NULL CHARACTER STRING
ENDMSG: .ASCIZ <15><12>/END PASS /

```

E09

MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 108  
SVKACJA.P11 20-DEC-76 15:02 END OF PASS ROUTINE

SEG 0110

4317 014700 000  
4318 014702  
4319

.EVEN

```

4320
4321
4322      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 0
4323 014702 011637 000706  †ZERO: MOV      (SP),0#OLDPC      ;GET PC+2 WHERE UNEXPECTED TRAP OCCURRED
4324 014706 104200          ERROR 200      ;*****ERROR 200*****
4325          ;UNEXPECTED TRAP TO LOCATION 0
4326          ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4327 014710 000000          HALT          ;PROGRAM MUST BE RESTARTED AT THIS POINT
4328
4329
4330      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 4
4331 014712 011637 000706  †IMTRP: MOV      (SP),0#OLDPC      ;GET PC+2 WHERE UNEXPECTED TIMEOUT TRAP OCCURRED
4332 014716 104204          ERROR 204      ;*****ERROR 204*****
4333          ;UNEXPECTED TRAP TO LOCATION 4
4334          ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4335 014720 000000          HALT          ;PROGRAM MUST BE RESTARTED AT THIS POINT
4336
4337
4338      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 10
4339 014722 011637 000706  †ILLTRP: MOV      (SP),0#OLDPC      ;GET PC+2 WHERE UNEXPECTED ILLEGAL INSTRUCTION TRAP OCCUR
4340 014726 104210          ERROR 210      ;*****ERROR 210*****
4341          ;UNEXPECTED TRAP TO LOCATION 10
4342          ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4343 014730 000000          HALT          ;PROGRAM MUST BE RESTARTED AT THIS POINT
4344
4345
4346
4347
4348      ;SUBROUTINE TO SET OPERAND VALUES
4349
4350 014732 012567 163706  NPREP: MOV      (R5)+,S1LN      ;STORE INSTRUCTION TEST ARGUMENTS
4351 014736 012567 163704          MOV      (R5)+,S1ADR
4352 014742 012567 163702          MOV      (R5)+,S2LN
4353 014746 012567 163700          MOV      (R5)+,S2ADR
4354 014752 012567 163676          MOV      (R5)+,DSTLN
4355 014756 000205          RTS      R5
4356
4357
4358      ;SUBROUTINE TO SET UP GENERAL REGISTERS
4359
4360 01476C 016700 163660  GENR: MOV      S1LN,R0      ;TRANSFER INSTRUCTION TEST ARGUMENTS TO
4361 014764 016701 163656          MOV      S1ADR,R1      ; THE GENERAL REGISTERS
4362 014770 016702 163654          MOV      S2LN,R2
4363 014774 016703 163652          MOV      S2ADR,R3
4364 015000 016704 163650          MOV      DSTLN,R4
4365 015004 016705 163646          MOV      DSTAD,R5
4366 015010 010637 000702          MOV      SP,0#SAVRE      ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
4367 015014 062737 000002 000702  ADD      #2,0#SAVRE      ;ADJUST SAVED SP BECAUSE OF JSR TO THIS ROUTINE
4368 015022 000207          RTS      PC
4369

```

```

4370
4371
4372      ;SUBROUTINE TO CLEAR BUFFER AREA
4373 015024 012700 017000  CLBUF:  MOV    #BUF,R0      ;POINT R0 TO BUFFER AREA
4374 015030 012701 000020      MOV    #20,R1      ;STORE BUFFER LENGTH IN R1
4375 015034 005020      IS:    CLR    (R0)+      ;CLEAR BUFFER
4376 015036 005301      DEC    R1          ;DECREMENT BUFFER LENGTH
4377 015040 001375      BNE    IS         ;BR IF NOT FINISHED
4378 015042 000207      RTS     PC         ;RETURN
4379
4380      ;SUBROUTINE TO RECORD EXPECTED PSW
4381 015044 012537 000700  XPSW:  MOV    (R5)+,R#EXPPSW ;STORE EXPECTED PSW VALUE IN "EXPPSW"
4382 015050 106700      MFPS   R0          ;GET PRESENT PSW
4383 015052 032700 000020      BIT    #TBIT,R0   ;IS T-BIT SET?
4384 015056 001403      BEQ    IS         ;BR IF NOT
4385 015060 052737 000020 000700  BIS    #TBIT,R#EXPPSW ;OTHERWISE SET T-BIT IN EXPECTED PSW VALUE
4386 015066 000205      IS:    RTS     R5         ;RETURN
4387
4388
4389
4390
4391      ;SUBROUTINE TO TEST FOR TRANSMIT DONE FLAG
4392 015070 005077 163614  TDONE: CLR    @TEMP      ;CLEAR TIMER
4393 015074 105777 163564      IS:    TSTB   @TCSR    ;IS SLU READY?
4394 015100 100404      BMI    RETN      ;BR IF READY
4395 015102 005277 163602      INC    @TEMP      ;OTHERWISE INCREMENT TIMER
4396 015106 001372      BNE    IS         ;BR IF NOT TIMED OUT
4397 015110 104300      ERROR  300       ;*****ERROR 300*****
4398
4399 015112 000207      RETN:  RTS     PC         ;RETURN
4400
4401
4402      ;SUBROUTINE TO HANDLE TTY INTERRUPTS IN INSTRUCTION
4403      ;INTERRUPTABILITY TESTS
4404
4405 015114      INTR:
4406 015114 021667 163554      CMP    (SP),PC1   ;WAS PC AT INSTRUCTION UNDER TEST?
4407 015120 001003      BNE    SEND      ;BR. IF NO
4408 015122 032704 177400      CKR4:  BIT    #177400,R4 ;IF YES, CHECK UPPER BYTE OF R4
4409 015126 001004      BNE    CLRINT    ;IF ZERO, INSTRUCTION WAS NOT INTERRUPTED-TRY AGAIN
4410 015130 013777 000554 163530  SEND:  MOV    @#NULL,@#BUF ;SEND ANOTHER CHARACTER
4411 015136 000002      RTI
4412 015140 042777 000100 163516  CLRINT: BIC    #100,@TCSR ;IF NON-ZERO, CLEAR INTERRUPT ENABLE
4413 015146 000002      RTI
4414

```

4415  
4416  
4417  
4418  
4419  
4420  
4421  
4422  
4423  
4424  
4425  
4426  
4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466

.SBTTL SCOPE HANDLER ROUTINE

::\*\*\*\*\*  
::THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT  
::AND LOAD THE TEST NUMBER(\$TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)  
::AND LOAD THE ERROR FLAG (\$ERFLG) INTO DISPLAY<15:08>  
::THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
::\*SW14=1 LOOP ON TEST  
::\*SW09=1 LOOP ON ERROR  
::\*SW08=1 LOOP ON TEST IN SWR<5:0>  
::\*CALL  
::\* SCOPE ::SCOPE=IOT

```
015150 $SCOPE:
015150 032777 040000 163362 1$: BIT #BIT14,$SWR ::LOOP ON PRESENT TEST?
015156 0C1065 BNE $OVER ::YES IF SW14=1
:*****START OF CODE FOR THE XOR TESTER*****
015160 000416 $XTSTR: BR 6$ ::IF RUNNING ON THE "XOR" TESTER CHANGE
:THIS INSTRUCTION TO A "NOP" (NOP=240)
015162 013746 000004 MOV @#ERRVEC,-(SP) ::SAVE THE CONTENTS OF THE ERROR VECTOR
015166 012737 015206 000004 MOV #5,$@#ERRVEC ::SET FOR TIMEOUT
015174 005737 177060 TST @#177060 ::TIME OUT ON XOR?
015200 012637 000004 MOV (SP)+,@#ERRVEC ::RESTORE THE ERROR VECTOR
015204 000434 BR $SVLAD ::GO TO THE NEXT TEST
015206 022626 5$: CMP (SP)+,(SP)+ ::CLEAR THE STACK AFTER A TIME OUT
015210 012637 000004 MOV (SP)+,@#ERRVEC ::RESTORE THE ERROR VECTOR
015214 000422 BR 7$ ::LOOP ON THE PRESENT TEST
015216 032777 000400 163314 6$::*****END OF CODE FOR THE XOR TESTER*****
015224 001407 BIT #BIT08,$SWR ::LOOP ON SPEC. TEST?
015226 017746 163306 BEQ 2$ ::BR IF NO
015232 042716 000300 MOV @SWR,-(SP) ::SET DESIRED TEST NUM. FROM SWR
015236 122667 163240 BIC #5,$SWR ::STRIP AWAY UNDESIREC BITS
015242 001433 CMPS (SP)+,$TSTNM ::ON THE RIGHT TEST?
015244 105767 163233 BEQ $OVER ::BR IF YES
015250 001412 2$: TSTB $ERFLG ::HAS AN ERROR OCCURRED?
015252 032777 001000 163260 BEQ $SVLAD ::BR IF NO
015260 001404 BIT #BIT09,$SWR ::LOOP ON ERROR?
015262 016767 163222 163216 7$: MOV $LPERR,$LPADR ::SET LOOP ADDRESS TO LAST SCOPE
015270 000420 BR $OVER
015272 105067 163205 4$: CLRB $ERFLG ::ZERO THE ERROR FLAG
015276 105267 163200 $SVLAD: INCB $TSTNM ::COUNT TEST NUMBERS
015302 116767 163174 163262 MOVB $TSTNM,$TESTN ::SET TEST NUMBER IN APT MAILBOX
015310 011667 163172 MOV (SP),$LPADR ::SAVE SCOPE LOOP ADDRESS
015314 011667 163170 MOV (SP),$LPERR ::SAVE ERROR LOOP ADDRESS
015320 005067 163234 CLR $ESCAPE ::CLEAR THE ESCAPE FROM ERROR ADDRESS
015324 112767 000001 163163 MOVB #1,$ERMAX ::ONLY ALLOW ONE(1) ERROR ON NEXT TEST
015332 016777 163144 163202 $OVER: MOV $TSTNM,@DISPLAY ::DISPLAY TEST NUMBER
015340 016716 163142 MOV $LPADR,(SP) ::FUDGE RETURN ADDRESS
015344 000002 RTI ::FIXES PS
```

4467  
4468  
4469  
4470  
4471  
4472  
4473  
4474  
4475  
4476  
4477  
4478  
4479  
4480  
4481  
4482  
4483  
4484  
4485  
4486  
4487  
4488  
4489  
4490  
4491  
4492  
4493  
4494  
4495  
4496  
4497  
4498  
4499  
4500  
4501  
4502  
4503  
4504  
4505  
4506  
4507  
4508  
4509  
4510  
4511  
4512  
4513  
4514  
4515

.SBTTL POWER DOWN AND UP ROUTINES

\*\*\*\*\*  
:POWER DOWN ROUTINE

```
$PWRDN: MOV    $ILLUP, @#PWRVEC    ;; SET FOR FAST UP
          MOV    #340, @#PWRVEC+2 ;; Prio:7
          RO, -(SP)                ;; PUSH R0 ON STACK
          MOV    R1, -(SP)          ;; PUSH R1 ON STACK
          MOV    R2, -(SP)          ;; PUSH R2 ON STACK
          MOV    R3, -(SP)          ;; PUSH R3 ON STACK
          MOV    R4, -(SP)          ;; PUSH R4 ON STACK
          MOV    R5, -(SP)          ;; PUSH R5 ON STACK
          MOV    @SWR, -(SP)        ;; PUSH @SWR ON STACK
          MOV    SP, $SAVR6        ;; SAVE SP
          MOV    $PWRUP, @#PWRVEC  ;; SET UP VECTOR
          HALT
          BR     .-2                ;; HANG UP
```

\*\*\*\*\*  
:POWER UP ROUTINE

```
$PWRUP: MOV    $ILLUP, @#PWRVEC  ;; SET FOR FAST DOWN
          MOV    $SAVR6, SP        ;; GET SP
          CLR    $SAVR6            ;; WAIT LOOP FOR THE TTY
1$:      INC    $SAVR6            ;; WAIT FOR THE INC
          BNE   1$                ;; OF WORD
          CLR    $STNM
          MOV    (SP)+, @SWR       ;; POP STACK INTO @SWR
          MOV    (SP)+, R5         ;; POP STACK INTO R5
          MOV    (SP)+, R4         ;; POP STACK INTO R4
          MOV    (SP)+, R3         ;; POP STACK INTO R3
          MOV    (SP)+, R2         ;; POP STACK INTO R2
          MOV    (SP)+, R1         ;; POP STACK INTO R1
          MOV    (SP)+, R0         ;; POP STACK INTO R0
          MOV    $PWRDN, @#PWRVEC  ;; SET UP THE POWER DOWN VECTOR
          MOV    #340, @#PWRVEC+2 ;; Prio:7
          TYPE   $POWER            ;; REPORT THE POWER FAILURE
          $PWRMG: .WORD $POWER     ;; POWER FAIL MESSAGE POINTER
          MOV    (PC)+, (SP)       ;; RESTART AT $LOOP
          $PWRAD: .WORD $LOOP      ;; RESTART ADDRESS
          BIC    #20, 2(SP)        ;; CLEAR "T" BIT
          CLR    $TBIT            ;; CLEAR THE "T" BIT FLAG
          RTI
          $ILLUP: HALT             ;; THE POWER UP SEQUENCE WAS STARTED
          BR     .-2              ;; BEFORE THE POWER DOWN WAS COMPLETE
          $SAVR6: 0                ;; PUT THE SP HERE
          $POWER: .ASCIZ <15><12>"POWER"
          .EVEN
```

4516  
4517  
4518  
4519  
4520  
4521  
4522  
4523  
4524  
4525  
4526  
4527  
4528  
4529  
4530  
4531  
4532  
4533  
4534  
4535  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555  
4556  
4557  
4558  
4559  
4560  
4561  
4562  
4563  
4564  
4565  
4566  
4567

.SETTL ERROR HANDLER ROUTINE

::\*\*\*\*\*  
::\*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.  
::\*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL  
::\*AND TYPE OUT THE PC OF THE ERROR INSTRUCTION  
::\*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:  
::\*SW15=1 HALT ON ERROR  
::\*SW13=1 INHIBIT ERROR TYPEOUTS  
::\*SW09=1 LOOP ON ERROR  
::\*CALL  
::\* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER

\$ERROR:  
7\$: INCB \$ERFLG ;;SET THE ERROR FLAG  
BEQ 7\$ ;;DON'T LET THE FLAG GO TO ZERO  
MOV \$STNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG  
INC \$ERTTL ;;INC THE ERROR COUNT  
MOV (SP),\$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION  
SUB #2,\$ERRPC  
MOV @ERRPC,\$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE  
BIT #BIT13,@SWR ;;SKIP TYPEOUT IF SET  
BNE 20\$ ;;SKIP TYPEOUTS  
TYPE \$CRLF  
MOV \$ERRPC,-(SP) ;;SAVE \$ERRPC FOR TYPEOUT  
 ;;ERROR ADDRESS  
 ;;GO TYPE--OCTAL ASCII(ALL DIGITS)  
20\$: TYPC  
TYPE .CRLF  
21\$: CMPB #APTENV,\$ENV ;;RUNNING IN APT MODE  
BNE 2\$ ;;NO SKIP APT ERROR REPORT  
MOV \$ITEMB,21\$ ;;SET ITEM NUMBER AS ERROR NUMBER  
JSR PC,\$ATY4 ;;REPORT FATAL ERROR TO APT  
21\$: .BYTE 0  
 .BYTE 0  
22\$: BR 22\$ ;;APT ERROR LOOP  
2\$: TST @SWR ;;HALT ON ERROR  
BPL 3\$ ;;SKIP IF CONTINUE  
HALT ;;HALT ON ERROR!  
3\$: BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?  
BEQ 4\$ ;;BR IF NO  
MOV \$LPERR,(SP) ;;FUDGE RETURN FOR LOOPING  
4\$: TST \$ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS  
BEQ 5\$ ;;BR IF NONE  
MOV \$ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE  
5\$: CMP #SENDAD,@#42 ;;ACT-11 AUTO-ACCEPT?  
BNE 6\$ ;;BRANCH IF NO  
HALT ;;YES  
6\$: RTI ;;RETURN

015546 105267 162731  
015546 001775  
015552 016777 162722 162760  
015562 005267 162724  
015566 011667 162724  
015572 162767 000002 162716  
015600 117767 162712 162706  
015606 032777 020000 162724  
015614 001007  
015616 104401 000563  
015622 016746 162670  
  
015626 104402  
015630 104401 000563  
015634  
015634 122767 000001 162744  
015642 001007  
015644 116767 162644 000004  
015652 004767 000604  
015656 000  
015657 000  
015660 000777  
015662 005777 162652  
015666 100001  
015670 000000  
015672 032777 001000 162640  
015700 001402  
015702 016716 162602  
015706 005767 162646  
015712 001402  
015714 016716 162640  
015720  
015720 022737 014604 000042  
015726 001001  
015730 000000  
015732  
015732 000002

4568  
4569  
4570  
4571  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600  
4601  
4602  
4603  
4604  
4605  
4606  
4607  
4608  
4609  
4610  
4611  
4612  
4613  
4614  
4615  
4616  
4617  
4618  
4619  
4620  
4621  
4622  
4623

.SBTTL TYPE ROUTINE

```

*****
*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
*
*CALL:
*1) USING A TRAP INSTRUCTION
* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
*OR
* TYPE
* MESADR
*
$TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
BPL 1$ ;; BR IF YES
HALT ;; HALT HERE IF NO TERMINAL
BR 3$ ;; LEAVE
1$: MOV RO, -(SP) ;; SAVE RO
MOV 22(SP), RO ;; GET ADDRESS OF ASCIZ STRING
CMPB #APTENV, $ENV ;; RUNNING IN APT MODE
BNE 62$ ;; NO, GO CHECK FOR APT CONSOLE
BITB #APTSPOOL, $ENVM ;; SPOOL MESSAGE TO APT
BEQ 62$ ;; NO, GO CHECK FOR CONSOLE
MOV RO, 61$ ;; SETUP MESSAGE ADDRESS FOR APT
JSR PC, $ATY3 ;; SPOOL MESSAGE TO APT
61$: .WORD 0 ;; MESSAGE ADDRESS
62$: BITB #APTCSUP, $ENVM ;; APT CONSOLE SUPPRESSED
BNE 60$ ;; YES, SKIP TYPE OUT
2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
60$: MOV (SP)+, RO ;; RESTORE RO
3$: ADD #2, (SP) ;; ADJUST RETURN PC
RTI ;; RETURN
4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
BEQ 8$
CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
BNE 5$
TST (SP)+ ;; POP <CR><LF> EQUIV
TYPE ;; TYPE A CR AND LF
$CRLF
CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
BR 2$ ;; GET NEXT CHARACTER
5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
BNE 2$ ;; IF NO GO GET NEXT CHAR.
MOV $NULL, -(SP) ;; GET # OF FILLER CHARS. NEEDED
AND THE NULL CHAR.
7$: DECB 1(SP) ;; DOES A NULL NEED TO BE TYPED?
BLT 6$ ;; BR IF NO--GO POP THE NULL OFF OF STACK
JSR PC, $TYPEC ;; GO TYPE A NULL

```

```

4624 016114 105367 000072          DECB  $CHARCNT      ;;DO NOT COUNT AS A CCUNT
4625 016120 000770          BR      7$          ;;LOOP
4626
4627          :HORIZONTAL TAB PROCESSOR
4628
4629 016122 112716 000040      8$:      MOVB  #' (SP)      ;;REPLACE TAB WITH SPACE
4630 016125 004767 000014      9$:      JSR   PC,$TYPEC      ;;TYPE A SPACE
4631 016132 132767 000007 000052  BITB  #7,$CHARCNT      ;;BRANCH IF NOT AT
4632 016140 001372          BNE   9$          ;;TAB STOP
4633 016142 005726          TST   (SP)+        ;;POP SPACE OFF STACK
4634 016144 000724          BR    2$          ;;GET NEXT CHARACTER
4635 016146 105777 162376  $TYPEC: TSTB  @STPS      ;;WAIT UNTIL PRINTER IS READY
4636 016152 100375          BPL  $TYPEC
4637 016154 116677 000002 162370  MOVB  2(SP),@STPB      ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4638 016162 122766 000015 000002  CMPB  #CR,2(SP)      ;;IS CHARACTER A CARRIAGE RETURN?
4639 016170 001003          BNE  1$          ;;BRANCH IF NO
4640 016172 105067 000014          CLRB  $CHARCNT      ;;YES--CLEAR CHARACTER COUNT
4641 016176 000406          BR    $TYPEX      ;;EXIT
4642 016200 122766 000012 000002  1$:    CMPB  #LF,2(SP)      ;;IS CHARACTER A LINE FEED?
4643 016206 001402          BEQ  $TYPEX      ;;BRANCH IF YES
4644 016210 105227          INCB  (PC)+        ;;COUNT THE CHARACTER
4645 016212 000000          $CHARCNT: .WORD 0      ;;CHARACTER COUNT STORAGE
4646 016214 000207          $TYPEX: RTS   PC
4647

```

4648  
4649  
4650  
4651  
4652  
4653  
4654  
4655  
4656  
4657  
4658  
4659  
4660  
4661  
4662  
4663  
4664  
4665  
4666  
4667  
4668  
4669  
4670  
4671  
4672  
4673  
4674  
4675  
4676  
4677  
4678  
4679  
4680  
4681  
4682  
4683  
4684  
4685  
4686  
4687  
4688  
4689  
4690  
4691  
4692  
4693  
4694  
4695  
4696  
4697  
4698  
4699  
4700  
4701  
4702  
4703

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOS    ;;CALL FOR TYPEOUT
*      .BYTE   N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
*      .BYTE   M              ;;M=1 OR 0
*                               ;;1=TYPE LEADING ZEROS
*                               ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPON    ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
*      TYPOC    ;;CALL FOR TYPEOUT
*$TYPOS: MOV      2(SP),-(SP)    ;;PICKUP THE MODE
        MOV      1(SP), $OFILL  ;;LOAD ZERO FILL SWITCH
        MOV      (SP)+, $OMODE+1 ;;NUMBER OF DIGITS TO TYPE
        ADD      #2, (SP)      ;;ADJUST RETURN ADDRESS
        BR       $TYPON
*$TYPOC: MOV      #1, $OFILL    ;;SET THE ZERO FILL SWITCH
        MOV      #6, $OMODE+1   ;;SET FOR SIX(6) DIGITS
*$TYPON: MOV      #5, $OCNT     ;;SET THE ITERATION COUNT
        MOV      R3,-(SP)      ;;SAVE R3
        MOV      R4,-(SP)      ;;SAVE R4
        MOV      R5,-(SP)      ;;SAVE R5
        MOV      $OMODE+1, R4   ;;GET THE NUMBER OF DIGITS TO TYPE
        NEG      R4
        ADD      #6, R4        ;;SUBTRACT IT FOR MAX. ALLOWED
        MOV      R4, $OMODE    ;;SAVE IT FOR USE
        MOV      $OFILL, R4    ;;GET THE ZERO FILL SWITCH
        MOV      12(SP), R5    ;;PICKUP THE INPUT NUMBER
        CLR      R3           ;;CLEAR THE OUTPUT WORD
        ROL     R5           ;;ROTATE MSB INTO "C"
        BR      3$          ;;GO DO MSB
        BR      3$          ;;FORM THIS DIGIT
        ROL     R5
        ROL     R5
        ROL     R5
        MOV     R5, R3
        ROL     R3           ;;GET LSB OF THIS DIGIT
        DECB   $OMODE       ;;TYPE THIS DIGIT
        BPL    7$           ;;BR IF NO
        BIC   #177770, R3   ;;GET RID OF JUNK
        BNE   4$           ;;TEST FOR 0
        ST    R4           ;;SUPPRESS THIS 0

```

```

016216 017646 000000
016222 116667 000001 000211
016230 112667 000207
016234 062716 000002
016240 000406
016242 112767 000001 000171
016250 112767 000006 000165
016256 112767 000005 000154
016264 010346
016266 010446
016270 010546
016272 116704 000145
016276 005404
016300 062704 000006
016304 110467 000132
016310 116704 000125
016314 016605 000012
005003
006105
000404
006105
006105
006105
010503
006103
016340 105367 000076
016344 100016
016346 042703 177770
016352 001002
016354 005704

```

1\$:  
2\$:  
3\$:



```

4726
4727      .SBTTL  APT COMMUNICATIONS ROUTINE
4728
4729      ;:*****
4730 016444 112767 000001 000236 $ATY1:  MOVB  #1,$FFLG      ;;TO REPORT FATAL ERROR.
4731 016452 112767 000001 000226 $ATY3:  MOVB  #1,$MFLG      ;;TO TYPE A MESSAGE
4732 016460 000403          $ATYC:  BR      $ATYC
4733 016462 112767 000001 000220 $ATY4:  MOVB  #1,$FFLG      ;;TO ONLY REPORT FATAL ERROR
4734 016470
4735 016470 010046          $ATYC:  MOV      RO,-(SP)      ;;PUSH RO ON STACK
4736 016472 010146          MOV      R1,-(SP)      ;;PUSH R1 ON STACK
4737 016474 105767 000206          TSTB   $MFLG      ;;SHOULD TYPE A MESSAGE?
4738 016500 001450          BEQ     5$      ;;IF NOT: BR
4739 016502 122767 000001 162076          CMPB   #APTENV,$ENV      ;;OPERATING UNDER APT?
4740 016510 001031          BNE     3$      ;;IF NOT: BR
4741 016512 132767 000100 162067          BITB   #APTPOOL,$ENVM      ;;SHOULD SPOOL MESSAGES?
4742 016520 001425          BEQ     3$      ;;IF NOT: BR
4743 016522 017600 000004          MOV      24(SP),RO      ;;GET MESSAGE ADDR.
4744 016526 062766 000002 000004          ADD      #2,4(SP)      ;;BUMP RETURN ADDR.
4745 016534 005767 162026          1$:  TST     $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
4746 016540 001375          BNE     1$      ;;IF NOT: WAIT
4747 016542 010067 162034          MOV      RO,$MSGAD      ;;PUT ADDR IN MAILBOX
4748 016546 105720          2$:  TSTB   (RO)+      ;;FIND END OF MESSAGE
4749 016550 001376          BNE     2$
4750 016552 166700 162024          SUB     $MSGAD,RO      ;;SUB START OF MESSAGE
4751 016556 006200          ASR     RO      ;;GET MESSAGE LNTH IN WORDS
4752 016560 010067 162020          MOV      RO,$MSGGLT      ;;PUT LENGTH IN MAILBOX
4753 016564 012767 000004 161774          MOV      #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
4754 016572 000413          BR      5$
4755 016574 017667 000004 000016 3$:  MOV      24(SP),4$      ;;PUT MSG ADDR IN JSR LINKAGE
4756 016602 062766 000002 000004          ADD      #2,4(SP)      ;;BUMP RETURN ADDRESS
4757 016610 016746 161162          MOV      177776,-(SP)      ;;PUSH 177776 ON STACK
4758 016614 004767 177114          JSR     PC,$TYPE      ;;CALL TYPE MACRO
4759 016620 000000          4$:  .WORD  0
4760 016622
4761 016622 105767 000062          5$:  10$:  TSTB   $FFLG      ;;SHOULD REPORT FATAL ERROR?
4762 016626 001416          BEQ     12$      ;;IF NOT: BR
4763 016630 005767 161752          TST     $ENV      ;;RUNNING UNDER APT?
4764 016634 001413          BEQ     12$      ;;IF NOT: BR
4765 016636 005767 161724          11$:  TST     $MSGTYPE      ;;FINISHED LAST MESSAGE?
4766 016642 001375          BNE     11$      ;;IF NOT: WAIT
4767 016644 017667 000004 161716          MOV      24(SP),$FATAL      ;;GET ERROR #
4768 016652 062766 000002 000004          ADD      #2,4(SP)      ;;BUMP RETURN ADDR.
4769 016660 005267 161702          INC     $MSGTYPE      ;;TELL APT TO TAKE ERROR
4770 016664 105067 000020          12$:  CLRB   $FFLG      ;;CLEAR FATAL FLAG
4771 016670 105067 000013          CLRB   $LFLG      ;;CLEAR LOG FLAG
4772 016674 105067 000006          CLRB   $MFLG      ;;CLEAR MESSAGE FLAG
4773 016700 012601          MOV      (SP)+,R1      ;;POP STACK INTO R1
4774 016702 012600          MOV      (SP)+,RO      ;;POP STACK INTO RO
4775 016704 000207          RTS     PC      ;;RETURN
4776 016706 000          $MFLG: .BYTE 0      ;;MESSG. FLAG
4777 016707 000          $LFLG: .BYTE 0      ;;LOG FLAG
4778 016710 000          $FFLG: .BYTE 0      ;;FATAL FLAG
4779          016712          .EVEN
4780          000200          APTSIZE=200
4781          000001          APTENV=001

```

C10

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 119  
DVKAJA.P11 20-DEC-76 15:02 APT COMMUNICATIONS ROUTINE

SEQ 0121

4782 000100 APTSPool=100  
4783 000040 APTCSUP=04C

4784  
4785  
4786  
4787  
4788  
4789  
4790  
4791  
4792  
4793  
4794  
4795  
4796  
4797  
4798  
4799  
4800  
4801  
4802  
4803  
4804  
4805  
4806  
4807  
4808  
4809  
4810  
4811  
4812  
4813  
4814  
4815  
4816  
4817  
4818  
4819  
4820  
4821  
4822  
4823  
4824  
4825  
4826  
4827  
4828

016712 010046  
016714 016600 000002  
016720 035740  
016722 111000  
016724 006300  
016726 016000 016746  
016732 000200

000004 000002

.SBTTL TRAP DECODER

\*\*\*\*\*  
\*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION  
\*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS  
\*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL  
\*GO TO THAT ROUTINE.

```
$TRAP:  MOV    RO, -(SP)           ;; SAVE RO
        MOV    2(SP), RO         ;; GET TRAP ADDRESS
        TST   -(RO)             ;; BACKUP BY 2
        MOVB  (RO), RO          ;; GET RIGHT BYTE OF TRAP
        ASL   RO                ;; POSITION FOR INDEXING
        MOV   $TRPAD(RO), RO    ;; INDEX TO TABLE
        RTS   RO                ;; GO TO ROUTINE
```

:: THIS IS USE TO HANDLE THE "GETPRI" MACRO

```
$TRAP2: MOV   (SP), -(SP)        ;; MOVE THE PC DOWN
        MOV   4(SP), 2(SP)      ;; MOVE THE PSW DOWN
        RTI                          ;; RESTORE THE PSW
```

.SBTTL TRAP TABLE

\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
\*BY THE "TRAP" INSTRUCTION.

	ROUTINE
\$TRPAD:	-----
.WORD	\$TRAP2
\$TYPE	::CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
\$TYPOC	::CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
\$TYPOS	::CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
\$TYPON	::CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)

005015 C-2115 030455  
016766 026461 053104 040513  
016774 026512 000101  
017000 J0020  
J00001

NAME: .ASCIZ <15><12> /MD-11-DVKAJ-A/

BUF: .BLKW 20  
.END





# G10

MAIN: MACY11 27(1006) 21-DEC-76 11:53 PAGE 124  
 CVKAR.F11 20-DEC-76 15:02 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0125

CLBUF	015024	753	849	945	1041	1137	1206	1292	1381	1468	1553	1637	1705	1793
CLRINT	015140	1882	1970	2057	2153	2255	2343	2439	4373*					
CMFN	= 076052	4409	4412*											
		463*	2548	2622	2597	2773	2850	2922	2996	3072	3145	3219	3293	3367
		3440	3514	3587	3676									
CMFNPC	012656	3664	3676*											
CR	= 000015	363*	4638	4648										
CRLF	= 000200	364*	4609	4648										
CVTCON	014260	4177	4179	4192*										
CVTNL	= 076053	463*	3756	3830	3903	3976	4049	4122	4201					
CVTPC	014356	4189	4201*											
DDISP	= 177570	370*	573											
DISPLA	000542	573*	716*	4464*	4533*									
DISPRE	000174	487*	716											
DISAD	000656	655*	796	892	988	1084	1174	1242	1256	1329	1343	1418	1432	1505
		1519	1590	1604	1674	1742	1756	1830	1844	1919	1933	2007	2021	2094
		2108	2198	2212	2292	2306	2379	2393	2485	2499	2580	2654	2729	2805
		2992	2954	3028	3104	3177	3251	3325	3399	3472	3546	3619	3710	3792
		3866	3939	4012	4085	4158	4239	4365						
DETLN	000654	654*	1170	1238	1257	1325	1344	1414	1433	1501	1520	1586	1605	1670
		1738	1757	1826	1845	1915	1934	2003	2022	2090	2109	2194	2213	2288
		2307	2375	2394	2491	2500	2575	2649	2724	2803	2877	2949	3023	3099
		3172	3246	3320	3394	3467	3541	3614	3705	3787	3861	3934	4007	4080
		4153	4234	4354*	4364									
DSWR	= 177570	369*	572											
EMTVEC	= 000030	458*	688*	689*										
ENDMSG	014664	4282	4315*											
ENDT1	001552	758	825	835*										
ENDT2	002034	864	921	931*										
ENDT21	006276	2222	2241*											
ENDT24	007274	2509	2528*											
ENDT3	002316	960	1017	1027*										
ENDT4	002600	1056	1113	1123*										
ENDT44	013012	3723	3736*											
ENDT53	014504	4245	4258*											
ERRVEC	= 000004	451*	4436	4437*	4439*	4442*								
EXPPSW	000700	664*	804	826	900	922	996	1018	1092	1114	1147	1215	1302	1391
		1478	1563	1647	1715	1803	1892	1980	2067	2171	2265	2352	2459	2553
		2626	2701	2777	2854	2926	3000	3076	3149	3223	3297	3371	3444	3518
		3591	3682	3760	3834	3907	3980	4053	4126	4207	4381*	4385*		
FILL	000660	656*	791	887	983	1079								
GENR	014760	760	856	952	1046	1140	1209	1295	1384	1471	1556	1640	1708	1796
		1885	1972	2060	2163	2258	2346	2449	2544	2618	2694	2769	2846	2918
		2992	3068	3142	3215	3289	3363	3437	3511	3584	3673	3753	3826	3899
		3972	4045	4118	4198	4360*								
		486	4816	4817	4818	4819								
GNS	= *****													
HV	= 000011	361*	4607	4648										
ILLTRP	014722	496	4339*											
INTR	015114	2155	2441	3665	4190	4435*								
IOIVEC	= 000020	456*	686*	687*										
IV	= 000012	362*	4642	4648										
IV	= 000010	471*	747*	765	766*	773	774*	778	779*	783	784*	789	789*	793
		794*	798	799*	806	807*	813	814*	821	822*	843*	861	862*	869
		870*	874	875*	879	880*	884	885*	889	890*	994	995*	902	903*
		909	910*	917	919*	939*	957	958*	965	966*	970	971*	975	976*
		980	981*	925	986*	990	991*	998	999*	1005	1006*	1013	1014*	1035*

H10

1053	1054*	1061	1062*	1066	1067*	1071	1072*	1076	1077*	1081	1082*	1086
1087*	1094	1095*	1101	1102*	1109	1110*	1131*	1149	1150*	1156	1157*	1160
1161*	1164	1165*	1168	1169*	1172	1173*	1176	1177*	1181	1182*	1200*	1217
1218*	1224	1225*	1228	1229*	1232	1233*	1236	1237*	1240	1241*	1244	1245*
1249	1250*	1260	1261*	1286*	1304	1305*	1311	1312*	1315	1316*	1319	1320*
1323	1324*	1327	1328*	1331	1332*	1336	1337*	1347	1348*	1375*	1393	1394*
1400	1401*	1404	1405*	1408	1409*	1412	1413*	1416	1417*	1420	1421*	1425
1426*	1436	1437*	1462*	1480	1481*	1487	1488*	1491	1492*	1495	1496*	1499
1500*	1503	1504*	1507	1508*	1512	1513*	1523	1524*	1547*	1565	1566*	1572
1573*	1576	1577*	1580	1581*	1584	1585*	1588	1589*	1592	1593*	1597	1598*
1608	1609*	1631*	1649	1650*	1656	1657*	1660	1661*	1664	1665*	1668	1669*
1672	1673*	1676	1677*	1681	1682*	1699*	1717	1718*	1724	1725*	1728	1729*
1732	1733*	1736	1737*	1740	1741*	1744	1745*	1749	1750*	1760	1761*	1787*
1805	1806*	1812	1813*	1816	1817*	1820	1821*	1824	1825*	1828	1829*	1832
1833*	1837	1838*	1848	1849*	1876*	1894	1895*	1901	1902*	1905	1906*	1909
1910*	1913	1914*	1917	1918*	1921	1922*	1925	1927*	1937	1938*	1964*	1982
1983*	1989	1990*	1993	1994*	1997	1998*	2001	2002*	2005	2006*	2009	2010*
2014	2015*	2025	2026*	2051*	2069	2070*	2076	2077*	2080	2081*	2084	2085*
2088	2089*	2092	2093*	2096	2097*	2101	2102*	2112	2113*	2138*	2173	2174*
2180	2181*	2184	2185*	2188	2189*	2192	2193*	2196	2197*	2200	2201*	2205
2206*	2216	2217*	2249*	2267	2268*	2274	2275*	2278	2279*	2282	2283*	2286
2287*	2290	2291*	2294	2295*	2299	2300*	2310	2311*	2337*	2354	2355*	2361
2362*	2365	2366*	2369	2370*	2373	2374*	2377	2378*	2381	2382*	2386	2387*
2397	2398*	2424*	2460	2461*	2467	2468*	2471	2472*	2475	2476*	2479	2480*
2483	2484*	2487	2488*	2492	2493*	2503	2504*	2536*	2554	2555*	2561	2562*
2565	2566*	2569	2570*	2573	2574*	2577	2578*	2582	2583*	2588	2589*	2610*
2628	2629*	2635	2636*	2639	2640*	2643	2644*	2647	2648*	2651	2652*	2656
2657*	2662	2663*	2686*	2703	2704*	2710	2711*	2714	2715*	2718	2719*	2722
2723*	2726	2727*	2731	2732*	2737	2738*	2761*	2779	2780*	2786	2787*	2790
2791*	2794	2795*	2798	2799*	2802	2803*	2807	2808*	2813	2914*	2938*	2856
2857*	2863	2864*	2867	2868*	2871	2872*	2875	2876*	2879	2880*	2884	2885*
2890	2891*	2910*	2928	2929*	2935	2936*	2939	2940*	2943	2944*	2947	2948*
2951	2952*	2956	2957*	2962	2963*	2984*	3002	3003*	3009	3010*	3013	3014*
3017	3018*	3021	3022*	3025	3026*	3030	3031*	3036	3037*	3060*	3079	3079*
3085	3086*	3089	3090*	3093	3094*	3097	3098*	3101	3102*	3106	3107*	3112
3113*	3134*	3151	3152*	3158	3159*	3162	3163*	3166	3167*	3170	3171*	3174
3175*	3179	3180*	3185	3186*	3207*	3225	3226*	3232	3233*	3236	3237*	3240
3241*	3244	3245*	3248	3249*	3253	3254*	3259	3260*	3281*	3299	3300*	3306
3307*	3310	3311*	3314	3315*	3318	3319*	3322	3323*	3327	3328*	3333	3334*
3355*	3373	3374*	3380	3381*	3384	3385*	3388	3389*	3392	3393*	3396	3397*
3401	3402*	3407	3408*	3429*	3446	3447*	3453	3454*	3457	3458*	3461	3462*
3465	3466*	3469	3470*	3474	3475*	3480	3481*	3503*	3520	3521*	3527	3528*
3531	3532*	3535	3536*	3539	3540*	3543	3544*	3548	3549*	3554	3555*	3576*
3593	3594*	3600	3601*	3604	3605*	3608	3609*	3612	3613*	3616	3617*	3621
3622*	3627	3628*	3649*	3684	3685*	3691	3692*	3695	3696*	3699	3700*	3703
3704*	3707	3708*	3712	3713*	3718	3719*	3745*	3762	3763*	3769	3770*	3773
3774*	3777	3778*	3783	3784*	3789	3790*	3794	3795*	3818*	3836	3837*	3843
3844*	3847	3848*	3851	3852*	3857	3858*	3863	3864*	3868	3869*	3891*	3909
3910*	3916	3917*	3920	3921*	3924	3925*	3930	3931*	3936	3937*	3941	3942*
3964*	3982	3983*	3989	3990*	3993	3994*	3997	3999*	4003	4004*	4009	4010*
4014	4015*	4037*	4055	4056*	4062	4063*	4066	4067*	4070	4071*	4076	4077*
4082	4083*	4087	4088*	4110*	4128	4129*	4135	4136*	4139	4140*	4143	4144*
4149	4150*	4155	4156*	4160	4161*	4174*	4209	4210*	4216	4217*	4220	4221*
4224	4225*	4230	4231*	4236	4237*	4241	4242*					
725	4823*											
747	843	939	1035	1131	1200	1286	1375	1462	1547	1631	1699	1797

NAME 015760  
SYMBOL 014700



# J10

.MAIN. MACY11 27(1036) 21-DEC-76 11:53 PAGE 127  
 DVKAJA.P11 20-DEC-76 15:02 CROSS REFERENCE TABLE -- USER SYMBOLS

SEG 0129

SW10	=	002000	400#						
SW11	=	004000	399#						
SW12	=	010000	398#						
SW13	=	020000	397#						
SW14	=	040000	396#						
SW15	=	100000	395#						
SW2	=	000004	418#						
SW3	=	000010	417#						
SW4	=	000020	416#						
SW5	=	000040	415#						
SW6	=	000100	414#						
SW7	=	000200	413#						
SW8	=	000400	412#						
SW9	=	001000	411#						
S1ADR		000646	651#	776	872	968	1064	4351*	4361
S1LN		000644	650#	771	867	963	1059	4350*	4360
S1T1		001550	749	829#					
S1T10		003630	1377	1443#					
S1T11		004044	1464	1530#					
S1T12		004256	1549	1615#					
S1T13		004442	1633	1687#					
S1T14		004650	1701	1767#					
S1T15		005066	1789	1855#					
S1T16		005304	1978	1944#					
S1T17		005522	1966	2032#					
S1T2		002032	845	925#					
S1T20		005736	2053	2119#					
S1T21		006260	2149	2223#					
S1T22		006514	2251	2317#					
S1T23		006730	2339	2404#					
S1T24		007256	2435	2510#					
S1T25		007460	2538	2594#					
S1T26		007640	2612	2668#					
S1T27		010020	2688	2743#					
S1T3		002314	941	1021#					
S1T30		010202	2763	2819#					
S1T31		010364	2840	2896#					
S1T32		010542	2912	2968#					
S1T33		010722	2986	3042#					
S1T34		011104	3062	3118#					
S1T35		011262	3136	3191#					
S1T36		011442	3209	3265#					
S1T37		011622	3283	3339#					
S1T4		002576	1037	1117#					
S1T40		012002	3357	3413#					
S1T41		012160	3431	3486#					
S1T42		012336	3505	3560#					
S1T43		012514	3578	3633#					
S1T44		013002	3660	3724#					
S1T5		002772	1133	1187#					
S1T6		003176	1202	1267#					
S1T7		003412	1288	1354#					
S2ADR		000652	653#	786	882	978	1074	4353*	4363
S2LN		000650	652#	781	877	973	1069	4352*	4362
S2T1		001551	751	831#					
S2T10		003634	1379	1448#					



TST15	004662	1766	1786*																	
TST16	005100	1854	1875*																	
TST17	005316	1943	1963*																	
TST2	001566	942*																		
TST20	005532	2031	2050*																	
TST21	005746	2118	2137*																	
TST22	006310	2139	2145	2248*																
TST23	006526	2316	2336*																	
TST24	006742	2403	2423*																	
TST25	007306	2425	2431	2535*																
TST26	007466	2593	2609*																	
TST27	007650	2667	2685*																	
TST3	002050	938*																		
TST30	010030	2742	2760*																	
TST31	010212	2818	2837*																	
TST32	010370	2895	2909*																	
TST33	010550	2967	2983*																	
TST34	010732	3041	3059*																	
TST35	011112	3117	3133*																	
TST36	011270	3190	3206*																	
TST37	011450	3264	3280*																	
TST4	002332	1034*																		
TST40	011630	3338	3354*																	
TST41	012010	3412	3428*																	
TST42	012166	3485	3502*																	
TST43	012344	3559	3575*																	
TST44	012522	3632	3648*																	
TST45	013024	3650	3656	3744*																
TST46	013176	3798	3817*																	
TST47	013352	3872	3890*																	
TST5	002614	1130*																		
TST50	013526	3945	3963*																	
TST51	013702	4018	4036*																	
TST52	014056	4091	4109*																	
TST53	014222	4164	4173*																	
TST6	002774	1186	1199*																	
TST7	003206	1266	1285*																	
TVECT	000670	660*	734*	2155*	2241*	2441*	2528*	3665*	3736*	4190*	4258*									
TYPE =	104401	725	4282	4503	4540	4544	4612	4709	4816*											
TYPOC =	104402	4543	4817*																	
TYPON =	104404	4819*																		
TYPOS =	104403	4818*																		
TZERO	014702	492	4323*																	
T1	001374	763*	810																	
T1CONT	001402	756	770*																	
T2	001656	859*	906																	
T2CONT	001664	852	866*																	
T21CON	006004	2141	2143	2146*																
T24CON	007000	2427	2429	2432*																
T3	002140	955*	1002																	
T3CONT	002146	948	962*																	
T4	002422	1051*	1098																	
T4CONT	002430	1044	1058*																	
T44CON	012560	3652	3654	3657*																
X =	000000	472*																		
XPSW	015044	758	854	950	1046	1138	1207	1293	1382	1469	1554	1639	1706	1794						

	1883	1971	2058	2159	2256	2344	2445	2542	2616	2692	2767	2844	2916
	2990	3066	3140	3213	3287	3361	3435	3509	3582	3669	3751	3824	3897
	3970	4043	4116	4194	4381*								
\$APTHD 000400	530	536*											
\$ASTAT= ***** U	4761	4776											
\$ATYC 016470	4732	4734*											
\$ATY1 016444	4730*												
\$ATY3 016452	4597	4731*											
\$ATY4 016462	4549	4733*											
\$AUTOB 000534	569*												
\$BASE 000642	631*	727											
\$BDADR 000522	564*	812*	908*	1004*	1100*								
\$BDDAT 000526	566*												
\$CHARC 016212	4614*	4624*	4631	4640*	4645*								
\$CKSWR= ***** U	4822												
\$CLR.T 014574	4286	4289*											
\$CMTAG 000500	552*	680											
\$CM3 = 000000	582*												
\$CPUOP 000614	605*												
\$CRLF 000563	584*	4540	4544	4568	4613	4648							
\$DEVCT 000576	596*												
\$DOAGN 014614	4278	4284	4291	4297*									
\$ENDAD 014604	517	723	4293*	4563									
\$ENDCT 014546	694	4280*											
\$ENULL 014660	4313*												
\$ENV 000606	601*	2142	2428	3653	4178	4546	4592	4739	4763				
\$ENVM 000607	602*	719	4594	4599	4741								
\$EOP 014516	4175	4181	4271*										
\$EOPCT 014540	694*	4277*	4281										
\$ERFLG 000503	555*	4422	4451	4457*	4467	4531*	4568						
\$ERMAX 000515	561*	696*	4463*	4467									
\$ERROR 015546	688	4530*											
\$ERRPC 000516	562*	4535*	4536*	4537	4541	4568							
\$ERRTB 000644	648*												
\$ERTTL 000512	559*	4534*	4568										
\$ESCAP 000560	582*	695*	4462*	4559	4561	4568							
\$ETABL 000606	600*												
\$ETEND 000644	542	632*											
\$FATAL 000570	593*	675*	4767*										
\$FFLG 016710	4730*	4733*	4761	4770*	4778*								
\$FILLC 000556	580*	4617	4648										
\$FILLS 000555	579*	4648											
\$GDADR 000520	563*												
\$GDDAT 000524	565*												
\$GET42 014556	4283*												
\$GTSWR= ***** U	4821												
\$HIBTS 000400	537*												
\$ICNT 000504	556*												
\$ILLUP 015530	4472	4488	4510*										
\$INTAG 000535	570*												
\$ITEMB 000514	560*	4537*	4548	4568									
\$LF 000564	585*	4568	4648										
\$LFLG 016707	4771*	4777*											
\$LOOP 014652	4305	4309*	4506										
\$LPADR 000506	557*	712*	4455*	4460*	4465	4467							
\$LPERR 000510	558*	713*	4455	4461*	4467	4558							







CATCH	329#	4323	4331	4339											
CMPNIN	345#	3649													
CMPNTS	344#	2536	2610	2686	2761	2838	2910	2984	3060	3134	3207	3281	3355	3429	3503
	3576														
CMPREP	331#	2536	2610	2686	2761	2838	2910	2984	3060	3134	3207	3281	3355	3429	3503
	3576	3658													
COMMEN	463#														
CVPREP	345#	3745	3818	3891	3964	4037	4110	4182							
CVTINT	347#	4174													
CVTSRC	342#	3799	3873	3946	4019	4092	4165	4246							
CVTTST	346#	3745	3818	3891	3964	4037	4110								
DECTRP	348#	747	843	939	1035										
EHLT	327#	765	773	778	783	788	793	798	806	813	821	861	869	874	879
	884	889	894	902	909	917	957	965	970	975	980	985	990	998	1005
	1013	1053	1061	1066	1071	1076	1081	1086	1094	1101	1109	1149	1156	1160	1164
	1168	1172	1176	1181	1217	1224	1228	1232	1236	1240	1244	1249	1260	1304	1311
	1315	1319	1323	1327	1331	1336	1347	1393	1400	1404	1408	1412	1416	1420	1425
	1436	1480	1487	1491	1495	1499	1503	1507	1512	1523	1565	1572	1576	1580	1584
	1588	1592	1597	1608	1649	1656	1660	1664	1668	1672	1676	1681	1717	1724	1728
	1732	1736	1740	1744	1749	1760	1805	1812	1816	1820	1824	1828	1832	1837	1848
	1894	1901	1905	1909	1913	1917	1921	1926	1937	1982	1989	1993	1997	2001	2005
	2009	2014	2025	2069	2076	2080	2084	2088	2092	2096	2101	2112	2173	2180	2184
	2188	2192	2196	2200	2205	2216	2267	2274	2278	2282	2286	2290	2294	2299	2310
	2354	2361	2365	2369	2373	2377	2381	2386	2397	2460	2467	2471	2475	2479	2483
	2487	2492	2503	2554	2561	2565	2569	2573	2577	2582	2588	2628	2635	2639	2643
	2647	2651	2656	2662	2703	2710	2714	2718	2722	2726	2731	2737	2779	2786	2790
	2794	2798	2802	2807	2813	2856	2863	2867	2871	2875	2879	2884	2890	2928	2935
	2939	2943	2947	2951	2956	2962	3002	3009	3013	3017	3021	3025	3030	3036	3078
	3085	3089	3093	3097	3101	3106	3112	3151	3158	3162	3166	3170	3174	3179	3185
	3225	3232	3236	3240	3244	3248	3253	3259	3299	3306	3310	3314	3318	3322	3327
	3333	3373	3380	3384	3388	3392	3396	3401	3407	3446	3453	3457	3461	3465	3469
	3474	3480	3520	3527	3531	3535	3539	3543	3548	3554	3593	3600	3604	3608	3612
	3616	3621	3627	3684	3691	3695	3699	3703	3707	3712	3718	3762	3769	3773	3777
	3783	3789	3794	3836	3843	3847	3851	3857	3863	3868	3909	3916	3920	3924	3930
	3936	3941	3982	3989	3993	3997	4003	4009	4014	4055	4062	4066	4070	4076	4082
	4087	4128	4135	4139	4143	4149	4155	4160	4209	4216	4220	4224	4230	4236	4241
ENDCOM	463#														
ENDPAS	325#	4282													
ERR	328#	765	773	778	783	788	793	798	806	813	821	361	869	874	879
	884	389	894	902	909	917	957	965	970	975	980	985	990	998	1005
	1013	1053	1061	1066	1071	1076	1081	1086	1094	1101	1109	1149	1156	1160	1164
	1168	1172	1176	1181	1217	1224	1228	1232	1236	1240	1244	1249	1260	1304	1311
	1315	1319	1323	1327	1331	1336	1347	1393	1400	1404	1408	1412	1416	1420	1425
	1436	1480	1487	1491	1495	1499	1503	1507	1512	1523	1565	1572	1576	1580	1584
	1588	1592	1597	1608	1649	1656	1660	1664	1668	1672	1676	1681	1717	1724	1728
	1732	1736	1740	1744	1749	1760	1805	1812	1816	1820	1824	1828	1832	1837	1848
	1894	1901	1905	1909	1913	1917	1921	1926	1937	1982	1989	1993	1997	2001	2005
	2009	2014	2025	2069	2076	2080	2084	2088	2092	2096	2101	2112	2173	2180	2184
	2188	2192	2196	2200	2205	2216	2267	2274	2278	2282	2286	2290	2294	2299	2310
	2354	2361	2365	2369	2373	2377	2381	2386	2397	2460	2467	2471	2475	2479	2483
	2487	2492	2503	2554	2561	2565	2569	2573	2577	2582	2588	2628	2635	2639	2643
	2647	2651	2656	2662	2703	2710	2714	2718	2722	2726	2731	2737	2779	2786	2790
	2794	2798	2802	2807	2813	2856	2863	2867	2871	2875	2879	2884	2890	2928	2935
	2939	2943	2947	2951	2956	2962	3002	3009	3013	3017	3021	3025	3030	3036	3078
	3085	3089	3093	3097	3101	3106	3112	3151	3158	3162	3166	3170	3174	3179	3185
	3225	3232	3236	3240	3244	3248	3253	3259	3299	3306	3310	3314	3318	3322	3327

# E11

	3333	3373	3380	3384	3388	3392	3396	3401	3407	3446	3452	3457	3461	3465	3469
	3474	3480	3520	3527	3531	3535	3539	3543	3548	3554	3593	3600	3604	3608	3612
	3616	3621	3627	3684	3691	3695	3699	3703	3707	3712	3718	3762	3769	3773	3777
	3783	3789	3794	3836	3843	3847	3851	3857	3863	3868	3909	3916	3920	3924	3930
	3936	3941	3982	3989	3993	3997	4003	4009	4014	4055	4062	4066	4070	4076	4082
	4087	4128	4135	4139	4143	4149	4155	4160	4209	4216	4220	4224	4230	4236	4241
ERROR	357#	765	773	778	783	788	793	798	806	813	821	861	869	874	879
	884	889	894	902	909	917	957	965	970	975	980	985	990	998	1005
	1013	1053	1061	1066	1071	1076	1081	1086	1094	1101	1109	1149	1156	1160	1164
	1169	1172	1176	1181	1217	1224	1228	1232	1236	1240	1244	1249	1260	1304	1311
	1315	1319	1323	1327	1331	1336	1347	1393	1400	1404	1408	1412	1416	1420	1425
	1436	1480	1487	1491	1495	1499	1503	1507	1512	1523	1565	1572	1576	1580	1584
	1588	1592	1597	1608	1649	1656	1660	1664	1668	1672	1676	1681	1717	1724	1729
	1732	1736	1740	1744	1749	1760	1805	1812	1816	1820	1824	1828	1832	1837	1848
	1894	1901	1905	1909	1913	1917	1921	1926	1937	1982	1989	1993	1997	2001	2005
	2009	2014	2025	2069	2076	2080	2084	2088	2092	2096	2101	2112	2173	2180	2184
	2188	2192	2196	2200	2205	2216	2267	2274	2278	2282	2286	2290	2294	2299	2310
	2354	2361	2365	2369	2373	2377	2381	2386	2397	2460	2467	2471	2475	2479	2483
	2487	2492	2503	2554	2561	2565	2569	2573	2577	2582	2588	2628	2635	2639	2643
	2647	2651	2656	2662	2703	2710	2714	2718	2722	2726	2731	2737	2779	2786	2790
	2794	2798	2802	2807	2813	2856	2863	2867	2871	2875	2879	2884	2890	2928	2935
	2939	2943	2947	2951	2956	2962	3002	3009	3013	3017	3021	3025	3030	3036	3078
	3085	3089	3093	3097	3101	3106	3112	3151	3158	3162	3166	3170	3174	3179	3195
	3225	3232	3236	3240	3244	3248	3253	3259	3299	3306	3310	3314	3318	3322	3327
	3333	3373	3380	3384	3388	3392	3396	3401	3407	3446	3452	3457	3461	3465	3469
	3474	3480	3520	3527	3531	3535	3539	3543	3548	3554	3593	3600	3604	3608	3612
	3616	3621	3627	3684	3691	3695	3699	3703	3707	3712	3718	3762	3769	3773	3777
	3783	3789	3794	3836	3843	3847	3851	3857	3863	3868	3909	3916	3920	3924	3930
	3936	3941	3982	3989	3993	3997	4003	4009	4014	4055	4062	4066	4070	4076	4082
	4087	4128	4135	4139	4143	4149	4155	4160	4209	4216	4220	4224	4230	4236	4241
	4324	4332	4340	4397											
ESCAPE	463#														
GETPRI	463#	4298													
GETSWR	463#														
INT	335#	4405													
MANS	338#	1275	1362	1451	1536	1621	1776	1864	1953	2040	2127	2233	2326	2413	2520
MNPREP	330#	747	843	939	1035	1131	1200	1286	1375	1462	1547	1631	1699	1787	1876
	1964	2051	2147	2249	2337	2433									
MSRC	337#	829	925	1021	1117	1187	1267	1354	1443	1530	1615	1687	1767	1855	1944
	2032	2119	2223	2317	2404	2510	2594	2668	2743	2819	2896	2968	3042	3118	3191
	3265	3339	3413	3486	3560	3633	3724								
MULT	463#														
NEWTST	463#	743	839	935	1031	1127	1196	1282	1371	1458	1543	1627	1695	1783	1872
	1960	2047	2134	2245	2333	2420	2532	2606	2682	2757	2834	2906	2980	3056	3130
	3203	3277	3351	3425	3499	3572	3645	3741	3814	3897	3960	4033	4106	4170	
NUMERI	341#	1131	1200	1286	1375	1462	1547	1631	1699	1787	1876	1964	2051	2249	2337
NUMINT	342#	2138	2424												
PGP	323#	463#	4494	4495	4773	4774									
PJSH	323#	463#	4474	4480	4734	4736	4757								
REPORT	463#														
RESETN	329#	747	843	939	1035	1131	1200	1286	1375	1462	1547	1631	1699	1787	1876
	1964	2051	2138	2249	2337	2424	2536	2610	2686	2761	2838	2910	2984	3060	3134
	3207	3281	3355	3429	3503	3576	3649	3745	3818	3891	3964	4037	4110	4174	
SCOPE	358#	746	842	938	1034	1130	1199	1285	1374	1461	1545	1630	1698	1786	1875
	1963	2050	2137	2248	2336	2423	2535	2609	2685	2760	2837	2909	2983	3059	3133
	3206	3280	3354	3428	3502	3575	3648	3744	3817	3890	3963	4036	4109	4173	4272



G11

.MAIN. MACY11 27(1006) 21-DEC-76 11:53 PAGE 138  
DVKAJA.P11 20-DEC-76 15:02 CROSS REFERENCE TABLE -- MACRO NAMES

SEG 0:39

. ABS. 017040 000

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

DVKAJA.DVKAJA/SOL/CRF=DVKAJA.P11  
RUN-TIME: 67 58 6 SECONDS  
RUN-TIME RATIO: 522/132=3.9  
CORE USED: 23K (45 PAGES)

H11