

LSI-11

DIS DECIMAL INST TST
CVKAJBO

AH-8220B MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 77-82
MADE IN USA



A grid of 10 columns and 15 rows of small, dense text blocks, likely representing data or instructions for the LSI-11 system. The text is too small to read clearly but appears to be organized in a structured format.



.REM %

IDENTIFICATION

PRODUCT CODE: AC-8218B-MC
PRODUCT NAME: CVKAJB0 DIS DECMAL INSTR TST
PRODUCT DATE: JANUARY 1982
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1977, 1982 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS
ALL RIGHTS RESERVED

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY
BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS
OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE
COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES
THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAIL-
ABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP
OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE
WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COM-
MITMENT BY DIGITAL EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR
RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
NOT SUPPLIED BY DIGITAL.

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DEC	PDP	UNIBUS	MASSBUS
DECUS	DECTAPE	VAX	

D I G I T A L

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
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
96
97
98
99
100
101
102
103
104
105
106
107
108

1.0 GENERAL INFORMATION

1.1 ABSTRACT

THIS DIAGNOSTIC VERIFIES THE OPERATION OF THE DIBOL DECIMAL INSTRUCTIONS OF THE LSI-11 [ADDN,SUBN,CMFN,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 (2.3.1). 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:

- CVKAI* DIS MOVE & STRING INSTRUCTION TEST
- CVKAJ* [THIS DIAGNOSTIC]
- CVKAB* LSI-11 EIS INSTRUCTION TEST

WHERE "*" IS THE LASTED 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 0000C0 TO 017104.

1.2.3 PRELIMINARY PROGRAMS

IT IS ASSUMED THAT THE FOLLOWING DIAGNOSTICS HAVE BEEN RUN:

- LSI-11 BASIC CPU TEST CVKAA*
- LSI-11 TRAPS TEST CVKAD*

WHERE "*" IS THE LASTED REVISION

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
141
142
143
144
145
146
147
148
149
150
151
152
153

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURES

CAN BE LOADED UNDER XXDP+ OR
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 SENVM
HIGH.

154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187

2.3.2 RUNNING UNDER APT

THE APT MAILBOX-ETABLE 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' & '\$VECT1' 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*

188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218

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

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
247
248
249
250
251
252
253
254
255
256
257
258
259

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 "\$TYPE" & "\$TYPOCT" 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.

260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287

4.5 NPREP

THIS ROUTINE IS USED TO STORE A COPY OF THE INSTRUCTION TEST ARGUMENTS TO BE STORED IN R0-->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.

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
331
332
333
334
335
336
337
338
339
340
341
342
343

5.0 MISCELLANEOUS

5.1 STACK POINTER

STACK POINTER IS INITIALLY SET TO 500 (OCTAL).

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 '\$TSTNM' 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

344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

.SBTTL BASIC DEFINITIONS

```

001100  ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS
HT= 11                ;;CODE FOR HORIZONTAL TAB
LF= 12                ;;CODE FOR LINE FEED
CR= 15                ;;CODE FOR CARRIAGE RETURN
CRLF= 200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776           ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774        ;;STACK LIMIT REGISTER
PIRQ= 177772         ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570         ;;HARDWARE SWITCH REGISTER
DDISP= 177570        ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0                ;;GENERAL REGISTER
R1= %1                ;;GENERAL REGISTER
R2= %2                ;;GENERAL REGISTER
R3= %3                ;;GENERAL REGISTER
R4= %4                ;;GENERAL REGISTER
R5= %5                ;;GENERAL REGISTER
R6= %6                ;;GENERAL REGISTER
R7= %7                ;;GENERAL REGISTER
SP= %6                ;;STACK POINTER
PC= %7                ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS
PRO= 0                ;;PRIORITY LEVEL 0
```


400	000040	PR1=	40	::PRIORITY LEVEL 1
401	000100	PR2=	100	::PRIORITY LEVEL 2
402	000140	PR3=	140	::PRIORITY LEVEL 3
403	000200	PR4=	200	::PRIORITY LEVEL 4
404	000240	PR5=	240	::PRIORITY LEVEL 5
405	000300	PR6=	300	::PRIORITY LEVEL 6
406	000340	PR7=	340	::PRIORITY LEVEL 7

407
408 :*'SWITCH REGISTER' SWITCH DEFINITIONS

409	100000	SW15=	100000
410	040000	SW14=	40000
411	020000	SW13=	20000
412	010000	SW12=	10000
413	004000	SW11=	4000
414	002000	SW10=	2000
415	001000	SW09=	1000
416	000400	SW08=	400
417	000200	SW07=	200
418	000100	SW06=	100
419	000040	SW05=	40
420	000020	SW04=	20
421	000010	SW03=	10
422	000004	SW02=	4
423	000002	SW01=	2
424	000001	SW00=	1
425		.EQUIV	SW09,SW9
426		.EQUIV	SW08,SW8
427		.EQUIV	SW07,SW7
428		.EQUIV	SW06,SW6
429		.EQUIV	SW05,SW5
430		.EQUIV	SW04,SW4
431		.EQUIV	SW03,SW3
432		.EQUIV	SW02,SW2
433		.EQUIV	SW01,SW1
434		.EQUIV	SW00,SW0

435
436 :*DATA BIT DEFINITIONS (BIT00 TO BIT15)

437	100000	BIT15=	100000
438	040000	BIT14=	40000
439	020000	BIT13=	20000
440	010000	BIT12=	10000
441	004000	BIT11=	4000
442	002000	BIT10=	2000
443	001000	BIT09=	1000
444	000400	BIT08=	400
445	000200	BIT07=	200
446	000100	BIT06=	100
447	000040	BIT05=	40
448	000020	BIT04=	20
449	000010	BIT03=	10
450	000004	BIT02=	4
451	000002	BIT01=	2
452	000001	BIT00=	1
453		.EQUIV	BIT09,BIT9
454		.EQUIV	BIT08,BIT8
455		.EQUIV	BIT07,BIT7

BASIC DEFINITIONS

```

456 .EQUIV BIT06,BIT6
457 .EQUIV BIT05,BIT5
458 .EQUIV BIT04,BIT4
459 .EQUIV BIT03,BIT3
460 .EQUIV BIT02,BIT2
461 .EQUIV BIT01,BIT1
462 .EQUIV BIT00,BIT0

```

```

463
464 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
465 ERRVEC= 4          ;;TIME OUT AND OTHER ERRORS
466 RESVEC= 10        ;;RESERVED AND ILLEGAL INSTRUCTIONS
467 TBITVEC=14        ;;"T" BIT
468 TRTVEC= 14        ;;TRACE TRAP
469 BPTVEC= 14        ;;BREAKPOINT TRAP (BPT)
470 IOTVEC= 20        ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
471 PWRVEC= 24        ;;POWER FAIL
472 EMTVEC= 30        ;;EMULATOR TRAP (EMT) **ERROR**
473 TRAPVEC=34        ;;"TRAP" TRAP
474 TKVEC= 60         ;;TTY KEYBOARD VECTOR
475 TPVEC= 64         ;;TTY PRINTER VECTOR
476 PIRQVEC=240       ;;PROGRAM INTERRUPT REQUEST VECTOR
477                   APTSIZE= 200
478                   APTENV= 001
479                   APTSPool= 100
480                   APTCSUP= 040
481                   $SWR= 171400
482                   $SWRMK= 300
483                   TBIT= 20
484                   $TN= 1
485                   N= 1
486                   X= $TN-1
487                   NXM= 177777
488                   ABASE= 177560
489                   AVECT1= 60
490                   .=0
491
492
493
494

```

```

495 .SBTTL TRAP CATCHER
496
497         .=0
498 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
499 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
500 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
501         .=174
502 000174 000000  DISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
503 000176 000000  SWREG:   .WORD 0          ;;SOFTWARE SWITCH REGISTER
504
505
506         .=0
507 000000 014702  TZERO          ;;SET LOCATIONS 0,4,6 TO ERROR REPORTS
508 000002 000340
509 000004 014712  TIMTRP
510 000006 000340
511 000010 014722  ILLTRP

```


512 000012 000340
513
514
515
516 000100 000100
517 000100 000102
518 000102 000002
519
520
521 000200 000200
522 000200 000167 000512
523
524
525 000400
526
527
528
529
530 000400
531 000046 000046
532 000046 014604
533 000052 000052
534 000052 000000
535 000400
536
537
538
539
540
541 000400
542 000024 000024
543 000024 000200
544 000044 000044
545 000044 000400
546 000400
547
548
549
550
551 000400
552 000400 000000
553 000402 000566
554 000404 000030
555 000406 000040
556 000410 000000
557 000412 000027
558
559

```
340  

.=100  
.WORD 102 ;HANDLE EVENT LINE INTERRUPTS  
.WORD 2  

.=200  
JMP START ;STARTING LOCATION FOR PROGRAM  

.=400  
.SBTTL ACT11 HOOKS  

:*****  
:HOOKS REQUIRED BY ACT11  

$SVPC= ;SAVE PC  

.=46  
$ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP  

.=52  
.WORD 0 ;;2)SET LOC.52 TO ZERO  

.$SVPC ;; RESTORE PC  

.SBTTL APT PARAMETER BLOCK  

:*****  
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  

:*****  

.$X= ;;SAVE CURRENT LOCATION  

.=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM  

200 ;;FOR APT START UP  

.=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.  

$APTHDR ;;POINT TO APT HEADER BLOCK  

.=.$X ;;RESET LOCATION COUNTER  

:*****  
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
:INTERFACE SPEC.  

$APTHD:  
$SHIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
$MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)  
$TSTM: .WORD 30 ;;RUN TIM OF LONGEST TEST  
$PASTM: .WORD 40 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
$UNITM: .WORD ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  

.WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

560
561
562
563
564
565
566 000500
567 000500 000500
568 000500 000000
569 000502 000
570 000503 000
571 000504 000000
572 000506 000000
573 000510 000000
574 000512 000000
575 000514 000
576 000515 001
577 000516 000000
578 000520 000000
579 000522 000000
580 000524 000000
581 000526 000000
582 000530 000000
583 000532 000000
584 000534 000
585 000535 000
586 000536 000000
587 000540 177570
588 000542 177570
589 000544 177560
590 000546 177562
591 000550 177564
592 000552 177566
593 000554 000
594 000555 002
595 000556 012
596 000557 000
597 000560 000000
598 000562 077
599 000563 015
600 000564 000012
601
602
603
604
605
606 000566
607 000566 000000
608 000570 000000
609 000572 000000
610 000574 000000
611 000576 000000
612 000600 000000
613 000602 000000
614 000604 000000
615 000606

.SBTTL COMMON TAGS

*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
*USED IN THE PROGRAM.

SCMTAG: . =500
\$STNM: .WORD 0
\$ERFLG: .BYTE 0
\$ICNT: .WORD 0
\$LPADR: .WORD 0
\$LPERR: .WORD 0
\$ERTTL: .WORD 0
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 0
\$BDADR: .WORD 0
\$GDDAT: .WORD 0
\$BDDAT: .WORD 0
\$AUTOB: .BYTE 0
\$INTAG: .BYTE 0
\$SWR: .WORD DSWR
\$DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$TPFLG: .BYTE 0
\$ESCAPE: 0
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCII <12>

:::START OF COMMON TAGS
:::CONTAINS THE TEST NUMBER
:::CONTAINS ERROR FLAG
:::CONTAINS SUBTEST ITERATION COUNT
:::CONTAINS SCOPE LOOP ADDRESS
:::CONTAINS SCOPE RETURN FOR ERRORS
:::CONTAINS TOTAL ERRORS DETECTED
:::CONTAINS ITEM CONTROL BYTE
:::CONTAINS MAX. ERRORS PER TEST
:::CONTAINS PC OF LAST ERROR INSTRUCTION
:::CONTAINS ADDRESS OF 'GOOD' DATA
:::CONTAINS ADDRESS OF 'BAD' DATA
:::CONTAINS 'GOOD' DATA
:::CONTAINS 'BAD' DATA
:::RESERVED--NOT TO BE USED
:::AUTOMATIC MODE INDICATOR
:::INTERRUPT MODE INDICATOR
:::ADDRESS OF SWITCH REGISTER
:::ADDRESS OF DISPLAY REGISTER
:::TTY KBD STATUS
:::TTY KBD BUFFER
:::TTY PRINTER STATUS REG. ADDRESS
:::TTY PRINTER BUFFER REG. ADDRESS
:::CONTAINS NULL CHARACTER FOR FILLS
:::CONTAINS # OF FILLER CHARACTERS REQUIRED
:::INSERT FILL CHARS. AFTER A 'LINE FEED'
:::'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
:::ESCAPE ON ERROR ADDRESS
:::QUESTION MARK
:::CARRIAGE RETURN
:::LINE FEED

.SBTTL APT MAILBOX-ETABLE

\$EVEN
\$MAIL: .WORD
\$MSGTY: .WORD AMSGTY
\$FATAL: .WORD AFATAL
\$TESTN: .WORD ATESTN
\$PASS: .WORD APASS
\$DEVCT: .WORD ADEVCT
\$UNIT: .WORD AUNIT
\$MSGAD: .WORD AMSGAD
\$MSGLG: .WORD AMSGLG
\$ETABLE: .WORD

:::APT MAILBOX
:::MESSAGE TYPE CODE
:::FATAL ERROR NUMBER
:::TEST NUMBER
:::PASS COUNT
:::DEVICE COUNT
:::I/O UNIT NUMBER
:::MESSAGE ADDRESS
:::MESSAGE LENGTH
:::APT ENVIRONMENT TABLE

616	000606	000	SENV:	.BYTE	AENV	::ENVIRONMENT BYTE
617	000607	000	SENV:	.BYTE	AENV	::ENVIRONMENT MODE BITS
618	000610	000000	SSWREG:	.WORD	ASWREG	::APT SWITCH REGISTER
619	000612	000000	SUSWR:	.WORD	AUSWR	::USER SWITCHES
620	000614	000000	SCPUOP:	.WORD	ACPUOP	::CPU TYPE,OPTIONS
621			:*			BITS 15-11=CPU TYPE
622			:*			11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
623			:*			11/70=06,PDQ=07,Q=10
624			:*			BIT 10=REAL TIME CLOCK
625			:*			BIT 9=FLOATING POINT PROCESSOR
626			:*			BIT 8=MEMORY MANAGEMENT
627	000616	000	\$MAMS1:	.BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
628	000617	000	\$MTYP1:	.BYTE	AMTYP1	::MEM. TYPE,BLK#1
629			:*			MEM.TYPE BYTE -- (HIGH BYTE)
630			:*			900 NSEC CORE=001
631			:*			300 NSEC BIPOLAR=002
632			:*			500 NSEC MOS=003
633	000620	000000	\$MADR1:	.WORD	AMADR1	::HIGH ADDRESS,BLK#1
634			:*			MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
635	000622	000	\$MAMS2:	.BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTF
636	000623	000	\$MTYP2:	.BYTE	AMTYP2	::MEM. TYPE,BLK#2
637	000624	000000	\$MADR2:	.WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
638	000626	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
639	000627	000	\$MTYP3:	.BYTE	AMTYP3	::MEM. TYPE,BLK#3
640	000630	000000	\$MADR3:	.WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
641	000632	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
642	000633	000	\$MTYP4:	.BYTE	AMTYP4	::MEM. TYPE,BLK#4
643	000634	000000	\$MADR4:	.WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
644	000636	000060	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
645	000640	000C00	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
646	000642	177560	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
647	000644		\$ETEND:			
648			.MEXIT			

ERROR POINTER TABLE

SEQ 0015

```
649 .SBTTL ERROR POINTER TABLE
650
651 :*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
652 :*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
653 :*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
654 :*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
655 :*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
656
657 :* EM ;:POINTS TO THE ERROR MESSAGE
658 :* DH ;:POINTS TO THE DATA HEADER
659 :* DT ;:POINTS TO THE DATA
660 :* DF ;:POINTS TO THE DATA FORMAT
661
662
663 000644 $ERRTB:
664
665 000644 000000 S1LN: .WORD 0 ;SOURCE1 LENGTH
666 000646 000000 S1ADR: .WORD ;SOURCE1 ADDRESS
667 000650 000000 S2LN: .WORD 0 ;SOURCE2 LENGTH
668 000652 000000 S2ADR: .WORD 0 ;SOURCE2 ADDRESS
669 000654 000000 DSTLN: .WORD ;DESTINATION LENGTH
670 000656 017044 DSTAD: .WORD BUF ;DESTINATION ADDRESS
671 000660 000000 FILL: .WORD ;FILL CHARACTER
672 000662 000000 TABLE: .WORD 0 ;TRANSLATION TABLE ADDRESS
673 000664 177564 TCSR: 177564 ;TCSR ADDRESS OF SLU USED FOR INTERRUPTS
674 000666 177566 TBUF: 177566 ;TBUF ADDRESS
675 000670 000064 TVECT: 64 ;TRANSMIT INTERRUPT VECTOR
676 000672 000066 TPSW: 66 ; AND PSW LOCATION
677 000674 000000 PCI: 0 ;ADDRESS OF TEST INSTRUCTION TO INTERRUPT
678 000676 000000 CCODES: 0 ;CONDITION CODES AFTER EXECUTION OF TEST INSTRUCTION
679 000700 000000 EXPPSW: 0 ;EXPECTED PSW
680 000702 000000 SAVR6: 0 ;STACK POINTER VALUE BEFORE TEST INSTRUCTION EXECUTION
681 000704 000000 BADR6: 0 ;BAD STACK POINTER VALUE
682 000706 000000 OLDPC: 0 ;ADDRESS WHERE UNEXPECTED TRAP OR INTERRUPT OCCURRED
683 000710 000000 TEMP: 0
684 000712 000000 TEMP1: 0
685 000714 000000 TEMP2: 0
686
687
688
689
690 000716 005067 177646 START: CLR $FATAL ;CLEAR ERROR NO.
691 000722 005067 177640 CLR $MSGTYP ;CLEAR MESSAGE TYPE
692 000726 005067 177640 CLR $TESTN ;CLEAR TEST NO.
693
694 .SBTTL INITIALIZE THE COMMON TAGS
695 000732 012706 000500 MOV #SCMTAG,R6 ;CLEAR THE COMMON TAGS ($CMTAG) AREA
696 000736 005026 CLR (R6)+ ;FIRST LOCATION TO BE CLEARED
697 000740 022706 000540 CMP #SWR,R6 ;DONE? ;CLEAR MEMORY LOCATION
698 000744 001374 BNE -6 ;LOOP BACK IF NO
699 000746 012706 000500 MOV #500,SP ;SETUP THE STACK POINTER
700 ;INITIALIZE A FEW VECTORS
701 000752 012737 015150 000020 MOV #SSCOPE,#IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
702 000760 012737 000340 000022 MOV #340,#IOTVEC+2 ;LEVEL 7
703 000766 012737 015546 000030 MOV #SERROR,#EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
704 000774 012737 000340 000032 MOV #340,#EMTVEC+2 ;LEVEL 7
```


INITIALIZE THE COMMON TAGS

```
705 001002 012737 016764 000034 MOV #STRAP,@TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
706 001010 012737 000340 000036 MOV #340,@TRAPVEC+2;LEVEL 7
707 001016 012737 015346 000024 MOV #SPWRDN,@PWRVEC ;POWER FAILURE VECTOR
708 001024 012737 000340 000026 MOV #340,@PWRVEC+2 ;LEVEL 7
709 001032 016767 013510 013500 MOV SENDCT,SEOPCT ;SETUP END-OF-PROGRAM COUNTER
710 001040 005067 177514 CLR $ESCAPE ;CLEAR THE ESCAPE ON ERROR ADDRESS
711 001044 112767 000001 177443 MOVB #1,$ERMAX ;ALLOW ONE ERROR PER TEST
712 ;INITIALIZE THE 'T-BIT' TRAP VECTOR. THEN LOAD LOCATION.
713 ;THE 'END-OF-PASS' (SEOP) ROUTINE, WITH A 'RTI' OR 'RTT'.
714 001052 012737 014650 000014 MOV #SRTN,@TBITVEC ;SET 'T' BIT VECTOR TO SRTN
715 001060 012737 000340 000016 MOV #340,@TBITVEC+2 ;LEVEL 7
716 001066 012767 000002 013554 MOV #RTI,$SRTN ;SET SRTN TO A RTI
717 001074 013767 000010 177606 MOV @RESVEC,TEMP ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR
718 001102 012737 001130 000010 MOV #2,$@RESVEC ;TRY TO DO A RTT
719 001110 005046 CLR -(SP) ;DUMMY PS
720 001112 012746 001120 MOV #1,$-(SP) ;AND PC
721 001116 000006 RTT ;TRY THE RTT
722 001120 012767 000006 013522 1$: MOV #RTT,$SRTN ;RTT IS LEGAL--SET SRTN TO A RTT
723 001126 000402 BR 3$
724 001130 062706 000010 2$: ADD #10,SP ;RTT ILLEGAL--CLEAN OFF THE STACK
725 001134 016737 177550 000010 3$: MOV TEMP,@RESVEC ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR
726 001142 005067 013510 CLR $TBIT ;CLEAR 'T' BIT SWITCH
727 001146 012767 001146 177332 MOV #,$SLPADR ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
728 001154 012767 001154 177326 MOV #,$SLPERR ;SETUP THE ERROR LOOP ADDRESS
729 ;SETUP FOR A SOFTWARE SWITCH REGISTER.
730 001162 012767 000176 177350 MOV #SWREG,SWR ;POINT TO SOFTWARE SWR
731 001170 012767 000174 177344 MOV #DISPREG,DISPLAY
732
733 001176 005067 177372 CLR $PASS ;CLEAR PASS COUNT
734 001202 132767 000200 177377 BITB #APTSIZE,$ENVM ;TEST USER SIZE UNDER APT
735 001210 001403 BEQ 4$ ;YES,USE NON-APT SWITCH
736 001212 012767 000610 177320 MOV #SSWREG,SWR ;NO,USE APT SWITCH REGISTER
737 001220 4$:
738 001220 026737 013360 000042 CMP SENDAD,@#42 ;UNDER ACT11?
739 001226 001424 BEQ BEGIN ;BR, IF YES (SKIP PROGRAM ID TYPEOUT WHEN UNDER AACT)
740 001230 104401 017032 TYPE, NAME
741 ;SET UP ADDRESSES OF SLU TO USE FOR INTERRUPTABILITY TES
742 001234 013700 000642 MOV @NSBASE,R0 ;GET ADDRESS OF THE SLU
743 001240 062700 000004 ADD #4,R0 ;ADJUST TO TCSR ADDRESS
744 001244 010037 000664 MOV R0,@TCSR ;STORE TCSR ADDRESS
745 001250 005720 TST (R0)+ ;ADJUST TO TBUF ADDRESS
746 001252 010037 000666 MOV R0,@TBUF ;STORE TBUF ADDRESS
747 001256 013700 000636 MOV @SVECT1,R0 ;GET SLU INTERRUPT VECTOR
748 001262 062700 000004 ADD #4,R0 ;ADJUST TO TRANSMIT INTERRUPT VECTOR
749 001266 010037 000670 MOV R0,@TVECT ;STORE TRANSMIT INTERRUPT VECTOR
750 001272 005720 TST (R0)+ ;ADJUST TO TRANSMIT INTERRUPT PSW
751 001274 010037 000672 MOV R0,@TPSW ;STORE TRANSMIT INTERRUPT PSW LOCATION
752
753
754 001300 106427 000200 BEGIN: MTPS #200 ;SET PRIORITY TO 7
755
756
757
758
759
760
```

```
::*****  
:*TEST 1 TEST 'ADDN' WITH SOURCE1 LENGTH =0 & SOURCE2 STRING VALID  
:*****
```

```

761 001304 000004          TST1: SCOPE
762 001306 004567 013420 JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
763 001312 000000          0          ;SOURCE1 LENGTH
764 001314 001550          S1T1      ;SOURCE1 ADDRESS
765 001316 000001          1          ;SOURCE2 LENGTH
766 001320 001551          S2T1      ;SOURCE2 ADDRESS
767 001322 000000          0          ;DESTINATION LENGTH
768 001324 004767 013474 JSR      PC,CLBUF    ;CLEAR BUFFER AREA
769 001330 013767 000010 177354 MOV     @#10,TEMP1  ;SAVE TIMEOUT TRAP VECTOR CONTENTS
770 001336 013767 000012 177350 MOV     @#12,TEMP2
771 001344 012737 001402 000010 MOV     #T1CONT,@#10 ;POINT TIMEOUT VECTOR TO TEST CONTINUATION
772 001352 012737 000340 000012 MOV     #340,@#12
773 001360 004567 013460 JSR      R5,XPSW    ;STORE EXPECTED PSW VALUE
774 001364 000217          .WORD    217
775 001366 004767 013366 JSR      PC,GENR    ;SET UP GENERAL REGISTERS
776 001372 000277          SCC
777
778 001374 076050          T1:      ADDN      ;EXECUTE 'ADDN' INSTRUCTION
779
780 001376 104001          ERROR    1      ;*****TEST 1 - ERROR 1*****
781                                     ;DECIMAL INSTRUCTION DID NOT TRAP ON
782                                     ;A ZERO SOURCE1 LENGTH
783 001400 000464          BR      ENDT1
784
785 001402          T1CONT:
786 001402 020067 177236 CMP     R0,S1LN    ;CHECK R0 UNCHANGED
787 001406 001401          BEQ     64$
788 001410 104002          ERROR    2      ;*****TEST 1 - ERROR 2*****
789                                     ;R0 CHANGED
790                                     ;R0 SHOULD STILL EQUAL CONTENTS OF 'S1LN'
791 001412 020167 177230 64$:    CMP     R1,S1ADR  ;CHECK R1 UNCHANGED
792 001416 001401          BEQ     65$
793 001420 104003          ERROR    3      ;*****TEST 1 - ERROR 3*****
794                                     ;R1 CHANGED
795                                     ;R1 SHOULD STILL EQUAL CONTENTS OF 'S1ADR'
796 001422 020267 177222 65$:    CMP     R2,S2LN  ;CHECK R2 UNCHANGED
797 001426 001401          BEQ     66$
798 001430 104004          ERROR    4      ;*****TEST 1 - ERROR 4*****
799                                     ;R2 CHANGED
800                                     ;R2 SHOULD STILL EQUAL THE CONTENTS OF 'S2LN'
801 001432 020367 177214 66$:    CMP     R3,S2ADR  ;CHECK R3 UNCHANGED
802 001436 001401          BEQ     67$
803 001440 104005          ERROR    5      ;*****TEST 1 - ERROR 5*****
804                                     ;R3 CHANGED
805                                     ;R3 SHOULD STILL EQUAL THE CONTENTS OF 'S2ADR'
806 001442 020467 177212 67$:    CMP     R4,FILL   ;CHECK R4 UNCHANGED
807 001446 001401          BEQ     68$
808 001450 104006          ERROR    6      ;*****TEST 1 - ERROR 6*****
809                                     ;R4 CHANGED
810                                     ;R4 SHOULD STILL EQUAL THE CONTENTS OF 'FILL'
811 001452 020567 177200 68$:    CMP     R5,DSTAD  ;CHECK R5 UNCHANGED
812 001456 001401          BEQ     69$
813 001460 104007          ERROR    7      ;*****TEST 1 - ERROR 7*****
814                                     ;R5 CHANGED
815                                     ;R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
816 001462          69$:

```



```

817 001462 016600 000002      MOV      2(SP),R0      ;CHECK PSW AT TIME OF TRAP
818 001466 042700 177400      BIC      #177400,R0
819 001472 020067 177202      CMP      R0,EXPPSW
820 001476 001401              BEQ      1$
821 001500 104010              ERROR    10
822                                ;*****TEST 1 - ERROR 10*****
823                                ;PSW ERROR
824                                ;EXPECTED PSW IS STORED AT 'EXPPSW'
825 001502 021627 001376      1$:     CMP      (SP),#T1+2
826 001506 001403              BEQ      2$
827 001510 011637 000522      MOV      (SP),@#SBDADR ;STORE BAD ADDRESS
828 001514 104011              ERROR    11
829                                ;*****TEST 1 - ERROR 11*****
830                                ;TRAP ADDRESS ERROR
831                                ;EXPECTED ADDRESS IS 'T1+?'
832 001516 010600              2$:     MOV      SP,R0
833 001520 062700 000004      ADD      #4,R0
834 001524 020037 000702      CMP      R0,@#SAVR6
835 001530 001401              BEQ      3$
836 001532 104012              ERROR    12
837                                ;*****TEST 1 - ERROR 12*****
838                                ;STACK POINTER WAS NOT RESTORED
839                                ;EXPECTED SP VALUE IS STORED AT 'SAVR6'
840 001534 012716 001552      3$:     MOV      #ENDT1,(SP)
841 001540 013766 000700 000002  MOV      @#EXPPSW,2(SP)
842 001546 000002              RTI
843                                ;RESTORE SP & PSW
844 001550              S1T1:   ;SOURCE1 STRING
845 001550          060          .BYTE    60          ;MOST SIGNIFICANT DIGIT
846 001551              S2T1:   ;SOURCE2 STRING
847 001551          060          .BYTE    60          ;MOST SIGNIFICANT DIGIT
848                                ;EVEN
849
850 001552 016737 177134 000010  ENDT1:  MOV      TEMP1,@#10 ;RESTORE TIMEOUT VECTOR CONTENTS
851 001560 016737 177130 000012  MOV      TEMP2,@#12
852
853
854                                ;*****
855                                ;*TEST 2 TEST 'ADDN' WITH A POSITIVE LEADING SIGN SOURCE1, & VALID SOURCE2
856                                ;*****
857 001566 000004      TST2:   SCOPE
858 001570 004567 013136      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
859 001574 000001              1          ;SOURCE1 LENGTH
860 001576 002032              S1T2      ;SOURCE1 ADDRESS
861 001600 000001              1          ;SOURCE2 LENGTH
862 001602 002033              S2T2      ;SOURCE2 ADDRESS
863 001604 000000              0          ;DESTINATION LENGTH
864 001606 004767 013212      JSR      PC,CLBUF      ;CLEAR BUFFER AREA
865 001612 013767 000010 177072  MOV      @#10,TEMP1    ;SAVE TIMEOUT TRAP VECTOR CONTENTS
866 001620 013767 000012 177066  MOV      @#12,TEMP2
867 001626 012737 001664 000010  MOV      #T2CONT,@#10 ;POINT TIMEOUT VECTOR TO TEST CONTINUATION
868 001634 012737 000340 000012  MOV      #340,@#12
869 001642 004567 013176      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
870 001646 000217              .WORD    217
871 001650 004767 013104      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
872 001654 000277              SCC

```

```
873  
874 001656 076050 T2: ADDN :EXECUTE 'ADDN' INSTRUCTION  
875  
876 001660 104001 ERROR 1 :*****TEST 2 - ERROR 1*****  
877 :DECIMAL INSTRUCTION DID NOT TRAP ON  
878 :A POSITIVE LEADING SIGN SOURCE1  
879 001662 000464 BR ENDT2  
880  
881 001664 T2CONT:  
882 001664 020067 176754 CMP R0,S1LN :CHECK R0 UNCHANGED  
883 001670 001401 BEQ 64$  
884 001672 104002 ERROR 2 :*****TEST 2 - ERROR 2*****  
885 :R0 CHANGED  
886 :R0 SHOULD STILL EQUAL CONTENTS OF 'S1LN'  
887 001674 020167 176746 64$: CMP R1,S1ADR :CHECK R1 UNCHANGED  
888 001700 001401 BEQ 65$  
889 001702 104003 ERROR 3 :*****TEST 2 - ERROR 3*****  
890 :R1 CHANGED  
891 :R1 SHOULD STILL EQUAL CONTENTS OF 'S1ADR'  
892 001704 020267 176740 65$: CMP R2,S2LN :CHECK R2 UNCHANGED  
893 001710 001401 BEQ 66$  
894 001712 104004 ERROR 4 :*****TEST 2 - ERROR 4*****  
895 :R2 CHANGED  
896 :R2 SHOULD STILL EQUAL THE CONTENTS OF 'S2LN'  
897 001714 020367 176732 66$: CMP R3,S2ADR :CHECK R3 UNCHANGED  
898 001720 001401 BEQ 67$  
899 001722 104005 ERROR 5 :*****TEST 2 - ERROR 5*****  
900 :R3 CHANGED  
901 :R3 SHOULD STILL EQUAL THE CONTENTS OF 'S2ADR'  
902 001724 020467 176730 67$: CMP R4,FILL :CHECK R4 UNCHANGED  
903 001730 001401 BEQ 68$  
904 001732 104006 ERROR 6 :*****TEST 2 - ERROR 6*****  
905 :R4 CHANGED  
906 :R4 SHOULD STILL EQUAL THE CONTENTS OF 'FILL'  
907 001734 020567 176716 68$: CMP R5,DSTAD :CHECK R5 UNCHANGED  
908 001740 001401 BEQ 69$  
909 001742 104007 ERROR 7 :*****TEST 2 - ERROR 7*****  
910 :R5 CHANGED  
911 :R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'  
912 001744 69$:  
913 001744 016600 000002 MOV 2(SP),R0 :CHECK PSW AT TIME OF TRAP  
914 001750 042700 177400 BIC #177400,R0  
915 001754 020067 176720 CMP R0,EXPPSW  
916 001760 001401 BEQ 1$  
917 001762 104010 ERROR 10 :*****TEST 2 - ERROR 10*****  
918 :PSW ERROR  
919 :EXPECTED PSW IS STORED AT 'EXPPSW'  
920 :RESULTANT PSW IS IN R0  
921 001764 021627 001660 1$: CMP (SP),#T2+2 :CHECK ADDRESS OF TRAP  
922 001770 001403 BEQ 2$  
923 001772 011637 000522 MOV (SP),#SBDADR :STORE BAD ADDRESS  
924 001776 104011 ERROR 11 :*****TEST 2 - ERROR 11*****  
925 :TRAP ADDRESS ERROR  
926 :EXPECTED ADDRESS IS 'T2+2'  
927 :RESULTANT ADDRESS IS STORED AT 'SBDADR'  
928 002000 010600 2$: MOV SP,R0 :CHECK SP RESTORATION
```



```

929 002002 062700 000004      ADD    #4,R0
930 002006 020037 000702      CMP    R0,#SAVR6
931 002012 001401              BEQ    3$
932 002014 104012              ERROR  12
933                                :*****TEST 2 - ERROR 12*****
934                                :STACK POINTER WAS NOT RESTORED
935                                :EXPECTED SP VALUE IS STORED AT 'SAVR6'
936 002016 012716 002034 000002 3$:  MOV    #ENDT2,(SP)
937 002022 013766 000700          MOV    @#EXPPSW,2(SP)
938 002030 000002          RTI
939                                :SOURCE1 STRING
940 002032              S1T2:  .BYTE  55          :MOST SIGNIFICANT DIGIT
941 002032              S2T2:  .BYTE  60          :SOURCE2 STRING
942 002033              .EVEN          :MOST SIGNIFICANT DIGIT
943 002033              060
944
945
946 002034 016737 176652 000010  ENDT2: MOV    TEMP1,@#10      :RESTORE TIMEOUT VECTOR CONTENTS
947 002042 016737 176646 000012  MOV    TEMP2,@#12
948
949
950                                :*****
951                                :*TEST 3      TEST 'ADDN' WITH A NEGATIVE LEADING SIGN SOURCE1, & VALID SOURCE2
952                                :*****
953 002050 000004      TST3:  SCOPE
954 002052 004567 012654      JSR    R5,NPREP          :PREPARE ARGUMENTS FOR INSTRUCTION TEST
955 002056 000001              1          :SOURCE1 LENGTH
956 002060 002314              S1T3          :SOURCE1 ADDRESS
957 002062 000001              1          :SOURCE2 LENGTH
958 002064 002315              S2T3          :SOURCE2 ADDRESS
959 002066 000000              0          :DESTINATION LENGTH
960 002070 004767 012730      JSR    PC,CLBUF          :CLEAR BUFFER AREA
961 002074 013767 000010 176610  MOV    @#10,TEMP1        :SAVE TIMEOUT TRAP VECTOR CONTENTS
962 002102 013767 000012 176604  MOV    @#12,TEMP2
963 002110 012737 002146 000010  MOV    #T3CONT,@#10     :POINT TIMEOUT VECTOR TO TEST CONTINUATION
964 002116 012737 000340 000012  MOV    #340,@#12
965 002124 004567 012714      JSR    R5,XPSW          :STORE EXPECTED PSW VALUE
966 002130 000217          .WORD  217
967 002132 004767 012622      JSR    PC,GENR          :SET UP GENERAL REGISTERS
968 002136 000277          SCC
969
970 002140 076050      T3:  ADDN          :EXECUTE 'ADDN' INSTRUCTION
971
972 002142 104001      ERROR  1          :*****TEST 3 - ERROR 1*****
973                                :DECIMAL INSTRUCTION DID NOT TRAP ON
974                                :A NEGATIVE LEADING SIGN SOURCE2
975 002144 000464      BR  ENDT3
976
977 002146      T3CONT:
978 002146 020067 176472      CMP    R0,S1LN          :CHECK R0 UNCHANGED
979 002152 001401          BEQ    64$
980 002154 104002          ERROR  2          :*****TEST 3 - ERROR 2*****
981                                :R0 CHANGED
982                                :R0 SHOULD STILL EQUAL CONTENTS OF 'S1LN'
983 002156 020167 176464 64$:  CMP    R1,S1ADR        :CHECK R1 UNCHANGED
984 002162 001401          BEQ    65$

```

```

985 002164 104003          ERROR 3          :*****TEST 3 - ERROR 3*****
986                                     :R1 CHANGED
987                                     :R1 SHOULD STILL EQUAL CONTENTS OF 'S1ADR'
988 002166 020267 176456 65$:  CMP      R2,S2LN          :CHECK R2 UNCHANGED
989 002172 001401          BEQ      66$
990 002174 104004          ERROR 4          :*****TEST 3 - ERROR 4*****
991                                     :R2 CHANGED
992                                     :R2 SHOULD STILL EQUAL THE CONTENTS OF 'S2LN'
993 002176 020367 176450 66$:  CMP      R3,S2ADR          :CHECK R3 UNCHANGED
994 002202 001401          BEQ      67$
995 002204 104005          ERROR 5          :*****TEST 3 - ERROR 5*****
996                                     :R3 CHANGED
997                                     :R3 SHOULD STILL EQUAL THE CONTENTS OF 'S2ADR'
998 002206 020467 176446 67$:  CMP      R4,FILL          :CHECK R4 UNCHANGED
999 002212 001401          BEQ      68$
1000 002214 104006         ERROR 6          :*****TEST 3 - ERROR 6*****
1001                                     :R4 CHANGED
1002                                     :R4 SHOULD STILL EQUAL THE CONTENTS OF 'FILL'
1003 002216 020567 176434 68$:  CMP      R5,DSTAD          :CHECK R5 UNCHANGED
1004 002222 001401          BEQ      69$
1005 002224 104007         ERROR 7          :*****TEST 3 - ERROR 7*****
1006                                     :R5 CHANGED
1007                                     :R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
1008 002226 000000 000002 69$:  MOV      2(SP),R0          :CHECK PSW AT TIME OF TRAP
1009 002226 016600 177400  BIC      #177400,R0
1010 002232 042700 176436  CMP      R0,EXPPSW
1011 002236 020067 176436  BEQ      1$
1012 002242 001401          ERROR 10         :*****TEST 3 - ERROR 10*****
1013 002244 104010         PSW ERROR
1014                                     :EXPECTED PSW IS STORED AT 'EXPPSW'
1015                                     :RESULTANT PSW IS IN R0
1016                                     :CHECK ADDRESS OF TRAP
1017 002246 021627 002142 1$:  CMP      (SP),#T3+2
1018 002252 001403          BEQ      2$
1019 002254 011637 000522  MOV      (SP),@#SBDADR  :STORE BAD ADDRESS
1020 002260 104011         ERROR 11         :*****TEST 3 - ERROR 11*****
1021                                     :TRAP ADDRESS ERROR
1022                                     :EXPECTED ADDRESS IS 'T3+2'
1023                                     :RESULTANT ADDRESS IS STORED AT 'SBDADR'
1024 002262 010600 000004 2$:  MOV      SP,R0
1025 002264 062700 000702  ADD      #4,R0
1026 002270 020037 000000  CMP      R0,@#SAVR6
1027 002274 001401          BEQ      3$
1028 002276 104012         ERROR 12         :*****TEST 3 - ERROR 12*****
1029                                     :STACK POINTER WAS NOT RESTORED
1030                                     :EXPECTED SP VALUE IS STORED AT 'SAVR6'
1031                                     :ERRONEOUS SO VALUE IS AT 'BADR6'
1032 002300 012716 002316 3$:  MOV      #ENDT3,(SP)
1033 002304 013766 000700 000002 MOV      @#EXPPSW,2(SP)
1034 002312 0C0002          RTI
1035
1036 002314 053             S1T3:  .BYTE 53          :SOURCE1 STRING
1037 002315 060             S2T3:  .BYTE 60          :MOST SIGNIFICANT DIGIT
1038                                     .EVEN
1039                                     .EVEN
1040

```



```

1041
1042 002316 016737 176370 000010 ENDT3: MOV TEMP1,@#10 ;RESTORE TIMEOUT VECTOR CONTENTS
1043 002324 016737 176364 000012 MOV TEMP2,@#12
1044
1045
1046
1047
1048
1049 002332 000004
1050 002334 004567 012372
1051 002340 000001
1052 002342 002576
1053 002344 000000
1054 002346 002577
1055 002350 000000
1056 002352 004767 012446 JSR PC,CLBUF ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
1057 002356 013767 000010 176326 MOV @#10,TEMP1 ;SOURCE1 LENGTH
1058 002364 013767 000012 176322 MOV @#12,TEMP2 ;SOURCE1 ADDRESS
1059 002372 012737 002430 000010 MOV #T4CONT,@#10 ;SOURCE2 LENGTH
1060 002400 012737 000340 000012 MOV #340,@#12 ;SOURCE2 ADDRESS
1061 002406 004567 012432 JSR R5,XPSW ;DESTINATION LENGTH
1062 002412 000217 .WORD 217 ;CLEAR BUFFER AREA
1063 002414 004767 012340 JSR PC,GENR ;SAVE TIMEOUT TRAP VECTOR CONTENTS
1064 002420 000277 SCC ;POINT TIMEOUT VECTOR TO TEST CONTINUATION
1065
1066 002422 076050 T4: ADDN ;STORE EXPECTED PSW VALUE
1067
1068 002424 104001 ERROR 1 ;SET UP GENERAL REGISTERS
1069
1070
1071 002426 000464 BR ENDT4 ;EXECUTE 'ADDN' INSTRUCTION
1072
1073 002430 T4CONT:
1074 002430 020067 176210 CMP R0,S1LN ;CHECK R0 UNCHANGED
1075 002434 001401 BEQ 64$
1076 002436 104002 ERROR 2 ;*****TEST 4 - ERROR 1*****
1077 ;RO CHANGED
1078 ;RO SHOULD STILL EQUAL CONTENTS OF 'S1LN'
1079 002440 020167 176202 64$: CMP R1,S1ADR ;CHECK R1 UNCHANGED
1080 002444 001401 BEQ 65$
1081 002446 104003 ERROR 3 ;*****TEST 4 - ERROR 2*****
1082 ;R1 CHANGED
1083 ;R1 SHOULD STILL EQUAL CONTENTS OF 'S1ADR'
1084 002450 020267 176174 65$: CMP R2,S2LN ;CHECK R2 UNCHANGED
1085 002454 001401 BEQ 66$
1086 002456 104004 ERROR 4 ;*****TEST 4 - ERROR 3*****
1087 ;R2 CHANGED
1088 ;R2 SHOULD STILL EQUAL THE CONTENTS OF 'S2LN'
1089 002460 020367 176166 66$: CMP R3,S2ADR ;CHECK R3 UNCHANGED
1090 002464 001401 BEQ 67$
1091 002466 104005 ERROR 5 ;*****TEST 4 - ERROR 4*****
1092 ;R3 CHANGED
1093 ;R3 SHOULD STILL EQUAL THE CONTENTS OF 'S2ADR'
1094 002470 020467 176164 67$: CMP R4,FILL ;CHECK R4 UNCHANGED
1095 002474 001401 BEQ 68$
1096 002476 104006 ERROR 6 ;*****TEST 4 - ERROR 5*****
;*****TEST 4 - ERROR 6*****

```

```

1097                                     :R4 CHANGED
1098                                     :R4 SHOULD STILL EQUAL THE CONTENTS OF 'FILL'
1099 002500 020567 176152 68$: CMP R5,DSTAD :CHECK R5 UNCHANGED
1100 002504 001401 BEQ 69$
1101 002506 104007 ERROR 7 :*****TEST 4 - ERROR 7*****
1102                                     :R5 CHANGED
1103                                     :R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
1104 002510 69$:
1105 002510 016600 000002 MOV 2(SP),R0 :CHECK PSW AT TIME OF TRAP
1106 002514 042700 177400 BIC #177400,R0
1107 002520 020067 176154 CMP R0,EXPPSW
1108 002524 001401 BEQ 1$
1109 002526 104010 ERROR 10 :*****TEST 4 - ERROR 10*****
1110                                     :PSW ERROR
1111                                     :EXPECTED PSW IS STORED AT 'EXPPSW'
1112                                     :RESULTANT PSW IS IN R0
1113 002530 021627 002424 1$: CMP (SP),#T4+2 :CHECK ADDRESS OF TRAP
1114 002534 001403 BEQ 2$
1115 002536 011637 000522 MOV (SP),@#$BADR :STORE BAD ADDRESS
1116 002542 104011 ERROR 11 :*****TEST 4 - ERROR 11*****
1117                                     :TRAP ADDRESS ERROR
1118                                     :EXPECTED ADDRESS IS 'T4+2'
1119                                     :RESULTANT ADDRESS IS STORED AT '$BADR'
1120 002544 010600 2$: MOV SP,R0 :CHECK SP RESTORATION
1121 002546 062700 000004 ADD #4,R0
1122 002552 020037 000702 CMP R0,@$SAVR6
1123 002556 001401 BEQ 3$
1124 002560 104012 ERROR 12 :*****TEST 4 - ERROR 12*****
1125                                     :STACK POINTER WAS NOT RESTORED
1126                                     :EXPECTED SP VALUE IS STORED AT '$SAVR6'
1127                                     :ERRONEOUS SO VALUE IS AT '$ADR6'
1128 002562 012716 002600 3$: MOV #ENDT4,(SP) :RESTORE SP & PSW
1129 002566 013766 000700 000002 MOV @#EXPPSW,2(SP)
1130 002574 000002 RTI
1131
1132 002576 S1T4: :SOURCE1 STRING
1133 002576 060 .BYTE 60 :MOST SIGNIFICANT DIGIT
1134 002577 S2T4: :SOURCE2 STRING
1135 002577 060 .BYTE 60 :MOST SIGNIFICANT DIGIT
1136 .EVEN
1137
1138 002600 016737 176106 000010 ENDT4: MOV TEMP1,@#10 :RESTORE TIMEOUT VECTOR CONTENTS
1139 002606 016737 176102 000012 MOV TEMP2,@#12
1140
1141
1142 :*****
1143 :*TEST 5 TEST 'ADDN' BY ADDING TWO ZEROES WITH ZERO DESTINATION LENGTH
1144 :*****
1145 002614 000004 TST5: SCOPE
1146 002616 004567 012110 JSR R5,NPREP :PREPARE ARGUMENTS FOR INSTRUCTION TEST
1147 002622 000001 1 :SOURCE1 LENGTH
1148 002624 002772 S1T5 :SOURCE1 ADDRESS
1149 002626 000001 1 :SOURCE2 LENGTH
1150 002630 002773 S2T5 :SOURCE2 ADDRESS
1151 002632 000000 0 :DESTINATION LENGTH
1152 002634 004767 012164 JSR PC,CLBUF :CLEAR BUFFER AREA

```


1153	002640	004567	012200		JSR	R5,XPSW	
1154	002644	000204			.WORD	204	
1155	002646	004767	012106		JSR	PC,GENR	:SET UP GENERAL REGISTERS
1156	002652	000277			SCC		:SET UP THE COMPLEMENT OF EXPECTED CC'S
1157	002654	000244			CLZ		
1158	002656	076050			ADDN		
1159							
1160	002660	106767	176012		MFPS	CCODES	:STORE RESULTANT PSW
1161	002664	042767	177400	176004	BIC	#177400,CCODES	:CLEAR UNUSED BITS
1162	002672	023767	000700	175776	CMP	@#EXPPSW,CCODES	:CHECK PSW AGAINST EXPECTED VALUE
1163	002700	001401			BEQ	64\$:BR, IF EQUAL
1164	002702	104001			ERROR	1	:*****TEST 5 - ERROR 1*****
1165							:PSW ERROR
1166							:EXPECTED PSW IS STORED AT 'SAVR6'
1167							:ERRONEOUS SP VALUE IS AT 'BADR6'
1168	002704			64\$:			
1169	002704	005700			TST	R0	:CHECK R0=0
1170	002706	001401			BEQ	65\$	
1171	002710	104002			ERROR	2	:*****TEST 5 - ERROR 2*****
1172							:R0 SHOULD BE ZERO
1173	002712	005701		65\$:	TST	R1	:CHECK R1=0
1174	002714	001401			BEQ	66\$	
1175	002716	104003			ERROR	3	:*****TEST 5 - ERROR 3*****
1176							:R1 SHOULD BE ZERO
1177	002720	005702		66\$:	TST	R2	:CHECK R2=0
1178	002722	001401			BEQ	67\$	
1179	002724	104004			ERROR	4	:*****TEST 5 - ERROR 4*****
1180							:R2 SHOULD BE ZERO
1181	002726	005703		67\$:	TST	R3	:CHECK R3=0
1182	002730	001401			BEQ	68\$	
1183	002732	104005			ERROR	5	:*****TEST 5 - ERROR 5*****
1184							:R3 SHOULD BE ZERO
1185	002734	020467	175714	68\$:	CMP	R4,DSTLN	:CHECK R4= DESTINATION LENGTH
1186	002740	001401			BEQ	69\$	
1187	002742	104006			ERROR	6	:*****TEST 5 - ERROR 6*****
1188							:R4 SHOULD STILL BE DESTINATION LENGTH
1189	002744	020567	175706	69\$:	CMP	R5,DSTAD	:CHECK R5 = DESTINATION ADDRESS
1190	002750	001401			BEQ	70\$	
1191	002752	104007			ERROR	7	:*****TEST 5 - ERROR 7*****
1192							:R5 SHOULD STILL BE DESTINATION ADDRESS
1193	002754	023706	000702	70\$:	CMP	@#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
1194	002760	001403			BEQ	71\$:BR IF OK
1195	002762	010637	000704		MOV	SP,@#BADR6	:COPY BAD SP VALUE
1196	002766	104010			ERROR	10	:*****TEST 5 - ERROR 10*****
1197							:STACK POINTER NOT RESTORED BY INSTRUCTION
1198							:EXPECTED SP IS STORED AT 'SAVR6'
1199							:ERRONEOUS SP VALUE IS AT 'BADR6'
1200	002770			71\$:			
1201	002770	000401			BR	TST6	:BR TO NEXT TEST
1202	002772			S1T5:			:SOURCE1 STRING
1203	002772	060			.BYTE	60	:MOST SIGNIFICANT DIGIT
1204	002773			S2T5:			:SOURCE2 STRING
1205	002773	060			.BYTE	60	:MOST SIGNIFICANT DIGIT
1206							
1207					.EVEN		
1208							

```
1209
1210
1211
1212
1213
1214 002774 000004
1215 002776 004567 011730
1216 003002 000004
1217 003004 003176
1218 003006 000002
1219 003010 003202
1220 003012 000002
1221 003014 004767 012004
1222 003020 004567 012020
1223 003024 000200
1224 003026 004767 011726
1225 003032 000277
1226 003034 076050
1227
1228 003036 106767 175634
1229 003042 042767 177400 175626
1230 003050 023767 000700 175620
1231 003056 001401
1232 003060 104001
1233
1234
1235
1236 003062
1237 003062 005700
1238 003064 001401
1239 003066 104002
1240
1241 003070 005701
1242 003072 001401
1243 003074 104003
1244
1245 003076 005702
1246 003100 001401
1247 003102 104004
1248
1249 003104 005703
1250 003106 001401
1251 003110 104005
1252
1253 003112 020467 175536
1254 003116 001401
1255 003120 104006
1256
1257 003122 020567 175530
1258 003126 001401
1259 003130 104007
1260
1261 003132 023706 000702
1262 003136 001403
1263 003140 010637 000704
1264 003144 104010
```

```
*****
*TEST 6 TEST 'ADDN' WITH POSITIVE OPERANDS, SRC1 LENGTH .GT. SRC2 LENGTH, DL = NO
*****
TST6: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4 ;SOURCE1 LENGTH
S1T6 ;SOURCE1 ADDRESS
2 ;SOURCE2 LENGTH
S2T6 ;SOURCE2 ADDRESS
2 ;DESTINATION LENGTH
;CLEAR BUFFER AREA
JSR PC,CLBUF
JSR R5,XPSW
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE CGPLEMENT OF EXPECTED CC'S
ADDN

MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP @WEXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 6 - 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 6 - ERROR 2*****
;R0 SHOULD BE ZERO
;CHECK R1=0

65$: TST R1
BEQ 66$
ERROR 3 ;*****TEST 6 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK R2=0

66$: TST R2
BEQ 67$
ERROR 4 ;*****TEST 6 - ERROR 4*****
;R2 SHOULD BE ZERO
;CHECK R3=0

67$: TST R3
BEQ 68$
ERROR 5 ;*****TEST 6 - ERROR 5*****
;R3 SHOULD BE ZERO
;CHECK R4= DESTINATION LENGTH

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

69$: CMP R5,DSTAD
BEQ 70$
ERROR 7 ;*****TEST 6 - ERROR 7*****
;R5 SHOULD STILL BE DESTINATION ADDRESS
;VERIFY STACK POINTER IS RESTORED
;BR IF OK
MOV SP,@WBADR6
ERROR 10 ;*****TEST 6 - ERROR 10*****
```


TEST 'ADDN' WITH POSITIVE OPERANDS, SRC1 LENGTH .GT. SRC2 LENGTH, DL = NO. SEQ 0026

```
1265                                     :STACK POINTER NOT RESTORED BY INSTRUCTION
1266                                     :EXPECTED SP IS STORED AT 'SAVR6'
1267                                     :ERRONEOUS SP VALUE IS AT 'BADR6'
1268 003146                               71$:
1269
1270 003146 012700 003204                 MOV #ANS6,R0           :CHECK ANSWER
1271 003152 016701 175500                 MOV DSTAD,R1          :POINT R0 TO EXPECTED ANSWER
1272 003156 016702 175472                 MOV DSTLN,R2          :POINT R1 TO RESULTANT ANSWER
1273 003162 122021                         :STORE ANSWER LENGTH IN R1
1274 003164 001401                         72$: CMPB (R0)+,(R1)+   :COMPARE EACH DIGIT
1275 003166 104011                         BEQ 73$               :BR IF EQUAL
1276                                     ERROR 11              :*****TEST 6 - ERROR 11*****
1277                                     :ERRONEOUS ANSWER
1278                                     :R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1279 003170 005302                         73$: DEC R2            :DECREMENT ANSWER LENGTH
1280 003172 001373                         BNE 72$              :BR IF NOT FINISHED
1281 003174 000404                         BR TST7              :BR TO NEXT TEST
1282 003176                               S1T6:                :SOURCE1 STRING
1283 003176 060                            .BYTE 60             :MOST SIGNIFICANT DIGIT
1284 003177 060                            .BYTE 60
1285 003200 062                            .BYTE 62
1286 003201 061                            .BYTE 61
1287 003202                               S2T6:                :SOURCE2 STRING
1288 003202 067                            .BYTE 67             :MOST SIGNIFICANT DIGIT
1289 003203 070                            .BYTE 70
1290 003204                               ANS6:                :EXPECTED ANSWER
1291 003204 071                            .BYTE 71
1292 003205 071                            .BYTE 71
1293                                     .EVEN
1294
1295
1296
1297
```

```
::*****
:*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
1315 003252 106767 175420                 MFPS CCODES          :STORE RESULTANT PSW
1316 003256 042767 177400 175412         BIC #177400,CCODES  :CLEAR UNUSED BITS
1317 003264 023767 000700 175404         CMP @EXPPSW,CCODES  :CHECK PSW AGAINST EXPECTED VALUE
1318 003272 001401                         BEQ 64$              :BR, IF EQUAL
1319 003274 104001                         ERROR 1              :*****TEST 7 - ERROR 1*****
1320                                     :PSW ERROR
```


1377 003420
 1378 003420 061
 1379 003421 060
 1380 003422 060
 1381 003423 160
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389 003424 000004
 1390 003426 004567 011300
 1391 003432 000004
 1392 003434 003630
 1393 003436 000002
 1394 003440 003634
 1395 003442 000002
 1396 003444 004767 011354
 1397 003450 004567 011370
 1398 003454 000212
 1399 003456 004767 011276
 1400 003462 000265
 1401 003464 000252
 1402 003466 076050
 1403
 1404 003470 106767 175202
 1405 003474 042767 177400 175174
 1406 003502 023767 000700 175166
 1407 003510 001401
 1408 003512 104001
 1409
 1410
 1411
 1412 003514
 1413 003514 005700
 1414 003516 001401
 1415 003520 104002
 1416
 1417 003522 005701
 1418 003524 001401
 1419 003526 104003
 1420
 1421 003530 005702
 1422 003532 001401
 1423 003534 104004
 1424
 1425 003536 005703
 1426 003540 001401
 1427 003542 104005
 1428
 1429 003544 020467 175104
 1430 003550 001401
 1431 003552 104006
 1432

ANS7: ;EXPECTED ANSWER
 ;MOST SIGNIFICANT DIGIT
 .BYTE 61
 .BYTE 60
 .BYTE 60
 .BYTE 160
 .EVEN

 : *TEST 10 TEST 'ADDN' WITH NEGATIVE OPERANDS, OVERFLOW
 : *****
 †ST10: SCOPE ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
 JSR R5,NPREP ;SOURCE1 LENGTH
 4 ;SOURCE1 ADDRESS
 S1T10 ;SOURCE2 LENGTH
 2 ;SOURCE2 ADDRESS
 S2T10 ;DESTINATION LENGTH
 2 ;CLEAR BUFFER AREA
 JSR PC,CLBUF
 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\$;BR, IF EQUAL
 ERROR 1 ;*****TEST 10 - ERROR 1*****
 ;PSW ERROR
 ;EXPECTED PSW IS STORED AT 'SAVR6'
 ;ERRONEOUS SP VALUE IS AT 'BADR6'
 64\$: ;CHECK R0=0
 TST R0
 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

```

1433 003554 020567 175076      69$:  CMP      R5,DSTAD      ;CHECK R5 = DESTINATION ADDRESS
1434 003560 001401              BEQ      70$
1435 003562 104007              ERROR    7      ;*****TEST 10 - ERROR 7*****
1436                                ;R5 SHOULD STILL BE DESTINATION ADDRESS
1437 003564 023706 000702      70$:  CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
1438 003570 001403              BEQ      71$      ;BR IF OK
1439 003572 010637 000704      MOV      SP,@#BADR6      ;COPY BAD SP VALUE
1440 003576 104010              ERROR    10      ;*****TEST 10 - ERROR 10*****
1441                                ;STACK POINTER NOT RESTORED BY INSTRUCTION
1442                                ;EXPECTED SP IS STORED AT 'SAVR6'
1443                                ;ERRONEOUS SP VALUE IS AT 'BADR6'
1444 003600                        71$:
1445                                ;CHECK ANSWER
1446 003600 012700 003636      MOV      #ANS10,R0      ;POINT R0 TO EXPECTED ANSWER
1447 003604 016701 175046      MOV      DSTAD,R1      ;POINT R1 TO RESULTANT ANSWER
1448 003610 016702 175040      MOV      DSTLN,R2      ;STORE ANSWER LENGTH IN R1
1449 003614 122021              CMPB     (R0)+,(R1)+    ;COMPARE EACH DIGIT
1450 003616 001401              BEQ      73$      ;BR IF EQUAL
1451 003620 104011              ERROR    11      ;*****TEST 10 - ERROR 11*****
1452                                ;ERRONEOUS ANSWER
1453                                ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1454                                ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1455 003622 005302      73$:  DEC      R2      ;DECREMENT ANSWER LENGTH
1456 003624 001373              BNE     72$      ;BR IF NOT FINISHED
1457 003626 000404              BR      TST11     ;BR TO NEXT TEST
1458 003630      S1T10:
1459 003630          060      .BYTE    60      ;SOURCE1 STRING
1460 003631          061      .BYTE    61      ;MOST SIGNIFICANT DIGIT
1461 003632          062      .BYTE    62
1462 003633          163      .BYTE    163
1463 003634      S2T10:
1464 003634          065      .BYTE    65      ;SOURCE2 STRING
1465 003635          164      .BYTE    164      ;MOST SIGNIFICANT DIGIT
1466 003636      ANS10:
1467 003636          067      .BYTE    67      ;EXPECTED ANSWER
1468 003637          167      .BYTE    167      ;MOST SIGNIFICANT DIGIT
1469
1470                                .EVEN
1471
1472
1473
1474
1475

```

```

::*****
:*TEST 11      TEST 'ADDN' WITH POSITIVE OPERANDS, S2L=S1L,OVERFLOW
:*****

```

```

1476 003640 000004              TST11: SCOPE
1477 003642 004567 011064      JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
1478 003646 000002              2      ;SOURCE1 LENGTH
1479 003650 004044              S1T11      ;SOURCE1 ADDRESS
1480 003652 000002              2      ;SOURCE2 LENGTH
1481 003654 004046              S2T11      ;SOURCE2 ADDRESS
1482 003656 000002              2      ;DESTINATION LENGTH
1483 003660 004767 011140      JSR      PC,CLBUF      ;CLEAR BUFFER AREA
1484 003664 004567 011154      JSR      R5,XPSW
1485 003670 000206              .WORD    206
1486 003672 004767 011062      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
1487 003676 000271              +SEN!SEC      ;SET UP THE COMPLEMENT OF EXPECTED CC'S
1488 003700 000246              +CLZ!CLV

```



```

1489 003702 076050          ADDN
1490
1491 003704 106767 174766    MFPS      CCODES      :STORE RESULTANT PSW
1492 003710 042767 177400 174760    BIC      #177400,CCODES :CLEAR UNUSED BITS
1493 003716 023767 000700 174752    CMP      @#EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
1494 003724 001401          BEQ      64$          :BR, IF EQUAL
1495 003726 104001          ERROR    1          :*****TEST 11 - ERROR 1*****
1496                                     :PSW ERROR
1497                                     :EXPECTED PSW IS STORED AT 'SAVR6'
1498                                     :ERRONEOUS SP VALUE IS AT 'BADR6'
1499                                     64$:
1500 003730 005700          TST      R0          :CHECK R0=0
1501 003732 001401          BEQ      65$
1502 003734 104002          ERROR    2          :*****TEST 11 - ERROR 2*****
1503                                     :R0 SHOULD BE ZERC
1504 003736 005701          TST      R1          :CHECK R1=0
1505 003740 001401          BEQ      66$
1506 003742 104003          ERROR    3          :*****TEST 11 - ERROR 3*****
1507                                     :R1 SHOULD BE ZERO
1508 003744 005702          TST      R2          :CHECK R2=0
1509 003746 001401          BEQ      67$
1510 003750 104004          ERROR    4          :*****TEST 11 - ERROR 4*****
1511                                     :R2 SHOULD BE ZERO
1512 003752 005703          TST      R3          :CHECK R3=0
1513 003754 001401          BEQ      68$
1514 003756 104005          ERROR    5          :*****TEST 11 - ERROR 5*****
1515                                     :R3 SHOULD BE ZERO
1516 003760 020467 174670    CMP      R4,DSTLN    :CHECK R4= DESTINATION LENGTH
1517 003764 001401          BEQ      69$
1518 003766 104006          ERROR    6          :*****TEST 11 - ERROR 6*****
1519                                     :R4 SHOULD STILL BE DESTINATION LENGTH
1520 003770 020567 174662    CMP      R5,DSTAD    :CHECK R5 = DESTINATION ADDRESS
1521 003774 001401          BEQ      70$
1522 003776 104007          ERROR    7          :*****TEST 11 - ERROR 7*****
1523                                     :R5 SHOULD STILL BE DESTINATION ADDRESS
1524 004000 023706 000702    CMP      @#SAVR6,SP  :VERIFY STACK POINTER IS RESTORED
1525 004004 001403          BEQ      71$          :BR IF OK
1526 004006 010637 000704    MOV      SP,@#BADR6 :COPY BAD SP VALUE
1527 004012 104010          ERROR    10         :*****TEST 11 - ERROR 10*****
1528                                     :STACK POINTER NOT RESTORED BY INSTRUCTION
1529                                     :EXPECTED SP IS STORED AT 'SAVR6'
1530                                     :ERRONEOUS SP VALUE IS AT 'BADR6'
1531 004014          71$:
1532                                     :CHECK ANSWER
1533 004014 012700 004050    MOV      #ANS11,R0   :POINT R0 TO EXPECTED ANSWER
1534 004020 016701 174632    MOV      DSTAD,R1    :POINT R1 TO RESULTANT ANSWER
1535 004024 016702 174624    MOV      DSTLN,R2    :STORE ANSWER LENGTH IN R1
1536 004030 122021          CMPB     (R0)+,(R1)+ :COMPARE EACH DIGIT
1537 004032 001401          BEQ      73$
1538 004034 104011          ERROR    11         :*****TEST 11 - ERROR 11*****
1539                                     :ERRONEOUS ANSWER
1540                                     :R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1541                                     :R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1542 004036 005302          DEC     R2          :DECREMENT ANSWER LENGTH
1543 004040 001373          BNE     72$
1544 004042 000403          BR      TST12       :BR TO NEXT TEST

```

1545 004044
1546 004044 063
1547 004045 062
1548 004046
1549 004046 066
1550 004047 070
1551 004050
1552 004050 060
1553 004051 060
1554
1555
1556
1557
1558
1559
1560

S1T11: :SOURCE1 STRING
:MOST SIGNIFICANT DIGIT
.BYTE 63
.BYTE 62
S2T11: :SOURCE2 STRING
:MOST SIGNIFICANT DIGIT
.BYTE 66
.BYTE 70
ANS11: :EXPECTED ANSWER
.BYTE 60
.BYTE 60
.EVEN

:TEST 12 TEST ADDN WITH NEGATIVE OPERANDS, S2L=S1L,OVERFLOW CARRY

1561 004052 000004
1562 004054 004567 010652
1563 004060 000002
1564 004062 004256
1565 004064 000002
1566 004066 004260
1567 004070 000001
1568 004072 004767 010726
1569 004076 004567 010742
1570 004102 000212
1571 004104 004767 010650
1572 004110 000265
1573 004112 000252
1574 004114 076050
1575

TST12: SCOPE :PREPARE ARGUMENTS FOR INSTRUCTION TEST
JSR R5,NPREP :SOURCE1 LENGTH
2 :SOURCE1 ADDRESS
S1T12 :SOURCE2 LENGTH
2 :SOURCE2 ADDRESS
S2T12 :DESTINATION LENGTH
1 :CLEAR BUFFER AREA
JSR PC,CLBUF
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

1576 004116 106767 174554
1577 004122 042767 177400 174546
1578 004130 023767 000700 174540
1579 004136 001401
1580 004140 104001
1581
1582
1583

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 12 - ERROR 1*****
:PSW ERROR
:EXPECTED PSW IS STORED AT 'SAVR6'
:ERRONEOUS SP VALUE IS AT 'BADR6'

1584 004142
1585 004142 005700
1586 004144 001401
1587 004146 104002
1588

64\$: TST R0 :CHECK R0=0
BEQ 65\$
ERROR 2 :*****TEST 12 - ERROR 2*****
:R0 SHOULD BE ZERO
:CHECK R1=0

1589 004150 005701
1590 004152 001401
1591 004154 104003
1592

65\$: TST R1 :CHECK R1=0
BEQ 66\$
ERROR 3 :*****TEST 12 - ERROR 3*****
:R1 SHOULD BE ZERO
:CHECK R2=0

1593 004156 005702
1594 004160 001401
1595 004162 104004
1596

66\$: TST R2 :CHECK R2=0
BEQ 67\$
ERROR 4 :*****TEST 12 - ERROR 4*****
:R2 SHOULD BE ZERO
:CHECK R3=0

1597 004164 005703
1598 004166 001401
1599 004170 104005
1600

67\$: TST R3 :CHECK R3=0
BEQ 68\$
ERROR 5 :*****TEST 12 - ERROR 5*****
:R3 SHOULD BE ZERO


```

1601 004172 020467 174456      68$:  CMP      R4,DSTLN      :CHECK R4= DESTINATION LENGTH
1602 004176 001401              BEQ      69$
1603 004200 104006              ERROR    6      :*****TEST 12 - ERROR 6*****
1604                                :R4 SHOULD STILL BE DESTINATION LENGTH
1605 004202 020567 174450      69$:  CMP      R5,DSTAD      :CHECK R5 = DESTINATION ADDRESS
1606 004206 001401              BEQ      70$
1607 004210 104007              ERROR    7      :*****TEST 12 - ERROR 7*****
1608                                :R5 SHOULD STILL BE DESTINATION ADDRESS
1609 004212 023706 000702      70$:  CMP      @#SAVR6,SP      :VERIFY STACK POINTER IS RESTORED
1610 004216 001403              BEQ      71$
1611 004220 010637 000704      MOV      SP,@#BADR6
1612 004224 104010              ERROR    10     :COPY BAD SP VALUE
1613                                :*****TEST 12 - ERROR 10*****
1614                                :STACK POINTER NOT RESTORED BY INSTRUCTION
1615                                :EXPECTED SP IS STORED AT 'SAVR6'
1616 004226      71$:
1617                                :ERRONEOUS SP VALUE IS AT 'BADR6'
1618 004226 012700 004262      MOV      #ANS12,R0      :CHECK ANSWER
1619 004232 016701 174420      MOV      DSTAD,R1      :POINT R0 TO EXPECTED ANSWER
1620 004236 016702 174412      MOV      DSTLN,R2      :POINT R1 TO RESULTANT ANSWER
1621 004242 122021              CMPB     (R0)+,(R1)+    :STORE ANSWER LENGTH IN R1
1622 004244 001401              BEQ      73$
1623 004246 104011              ERROR    11     :COMPARE EACH DIGIT
1624                                :BR IF EQUAL
1625                                :*****TEST 12 - ERROR 11*****
1626                                :ERRONEOUS ANSWER
1627 004250 005302      73$:  DEC      R2              :R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1628 004252 001373              BNE     72$            :R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1629 004254 000403              BR      TST13         :DECREMENT ANSWER LENGTH
1630 004256              S1T12:                :BR IF NOT FINISHED
1631 004256          066      .BYTE    66            :BR TO NEXT TEST
1632 004257          161      .BYTE    161         :SOURCE1 STRING
1633 004260              S2T12:                :MOST SIGNIFICANT DIGIT
1634 004260          064      .BYTE    64            :SOURCE2 STRING
1635 004261          165      .BYTE    165         :MOST SIGNIFICANT DIGIT
1636 004262      ANS12:
1637 004262          166      .BYTE    166         :EXPECTED ANSWER
1638                                .EVEN
1639                                004264
1640
1641
1642
1643
1644
1645 004264 000004
1646 004266 004567 010440      TST13: SCOPE
1647 004272 000001              JSR      R5,NPREP      :PREPARE ARGUMENTS FOR INSTRUCTION TEST
1648 004274 004442              1
1649 004276 000001              S1T13
1650 004300 004443              1
1651 004302 000000              S2T13
1652 004304 004767 010514      0
1653 004310 004567 010530      JSR      PC,CLBUF      :DESTINATION LENGTH
1654 004314 000204              JSR      R5,XPSW       :CLEAR BUFFER AREA
1655 004316 004767 010436      .WORD    204
1656 004322 000277              JSR      PC,GENR

```

```

:*****
:*TEST 13 TEST 'ADDN' WITH +SRC1 & -SRC2, ZERO RESULT
:*****

```

```

:SET UP GENERAL REGISTERS
:SET UP THE COMPLEMENT OF EXPECTED CC'S

```

```

1657 004324 000244          CLZ
1658 004326 076050          ADDN
1659
1660 004330 106767 174342    MFPS      CCODES      :STORE RESULTANT PSW
1661 004334 042767 177400 174334  BIC      #177400,CCODES  :CLEAR UNUSED BITS
1662 004342 023767 000700 174326  CMP      @#EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
1663 004350 001401          BEQ      64$          :BR, IF EQUAL
1664 004352 104001          ERROR   1          :*****TEST 13 - ERROR 1*****
1665
1666
1667
1668 004354          64$:          TST      R0          :CHECK R0=0
1669 004354 005700          BEQ      65$
1670 004356 001401          ERROR   2          :*****TEST 13 - ERROR 2*****
1671 004360 104002          :R0 SHOULD BE ZERO
1672
1673 004362 005701          65$:          TST      R1          :CHECK R1=0
1674 004364 001401          BEQ      66$
1675 004366 104003          ERROR   3          :*****TEST 13 - ERROR 3*****
1676
1677 004370 005702          66$:          TST      R2          :CHECK R2=0
1678 004372 001401          BEQ      67$
1679 004374 104004          ERROR   4          :*****TEST 13 - ERROR 4*****
1680
1681 004376 005703          67$:          TST      R3          :CHECK R3=0
1682 004400 001401          BEQ      68$
1683 004402 104005          ERROR   5          :*****TEST 13 - ERROR 5*****
1684
1685 004404 020467 174244          68$:          CMP      R4,DSTLN    :CHECK R4= DESTINATION LENGTH
1686 004410 001401          BEQ      69$
1687 004412 104006          ERROR   6          :*****TEST 13 - ERROR 6*****
1688
1689 004414 020567 174236          69$:          CMP      R5,DSTAD    :CHECK R5 = DESTINATION ADDRESS
1690 004420 001401          BEQ      70$
1691 004422 104007          ERROR   7          :*****TEST 13 - ERROR 7*****
1692
1693 004424 023706 000702          70$:          CMP      @#SAVR6,SP  :R5 SHOULD STILL BE DESTINATION ADDRESS
1694 004430 001403          BEQ      71$          :VERIFY STACK POINTER IS RESTORED
1695 004432 010637 000704          MOV      SP,@#BADR6  :BR IF OK
1696 004436 104010          ERROR   10         :COPY BAD SP VALUE
1697
1698
1699
1700 004440          71$:          TST14          :*****TEST 13 - ERROR 10*****
1701 004440 000401          BR      TST14          :STACK POINTER NOT RESTORED BY INSTRUCTION
1702 004442          S1T13:          .BYTE 63          :EXPECTED SP IS STORED AT "SAVR6"
1703 004442          S2T13:          .BYTE 163         :ERRONEOUS SP VALUE IS AT "BADR6"
1704 004443          :SOURCE1 STRING
1705 004443          :MOST SIGNIFICANT DIGIT
1706
1707
1708
1709
1710
1711
1712

```

```

:*****
:*TEST 14      TEST "ADDN" WITH -SRC1 & +SRC2, S1L .LT. S2L, /S2/ .GT. /S1/
:*****

```



```

1713 004444 000004          TST14: SCOPE
1714 004446 004567 010260 JSR      R5,NPREP      :PREPARE ARGUMENTS FOR INSTRUCTION TEST
1715 004452 000003          3          :SOURCE1 LENGTH
1716 004454 004650          S1T14      :SOURCE1 ADDRESS
1717 004456 000004          4          :SOURCE2 LENGTH
1718 004460 004653          S2T14      :SOURCE2 ADDRESS
1719 004462 000002          2          :DESTINATION LENGTH
1720 004464 004767 010334 JSR      PC,CLBUF      :CLEAR BUFFER AREA
1721 004470 004567 010350 JSR      R5,XPSW
1722 004474 000210          .WORD      210
1723 004476 004767 010256 JSR      PC,GENR      :SET UP GENERAL REGISTERS
1724 004502 000277          SCC
1725 004504 000250          CLN
1726 004506 076050          ADDN      :SET UP THE COMPLEMENT OF EXPECTED CC'S
1727
1728 004510 106767 174162 MFPS      CCODES      :STORE RESULTANT PSW
1729 004514 042767 177400 174154 BIC      #177400,CCODES :CLEAR UNUSED BITS
1730 004522 023767 000700 174146 CMP      @WEXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
1731 004530 001401          BEQ      64$
1732 004532 104001          ERROR    1          :*****TEST 14 - ERROR 1*****
1733          :PSW ERROR
1734          :EXPECTED PSW IS STORED AT 'SAVR6'
1735          :ERRONEOUS SP VALUE IS AT 'BADR6'
1736 004534          64$:
1737 004534 005700          TST      R0
1738 004536 001401          BEQ      65$
1739 004540 104002          ERROR    2          :*****TEST 14 - ERROR 2*****
1740          :R0 SHOULD BE ZERO
1741 004542 005701          65$: TST      R1
1742 004544 001401          BEQ      66$
1743 004546 104003          ERROR    3          :*****TEST 14 - ERROR 3*****
1744          :R1 SHOULD BE ZERO
1745 004550 005702          66$: TST      R2
1746 004552 001401          BEQ      67$
1747 004554 104004          ERROR    4          :*****TEST 14 - ERROR 4*****
1748          :R2 SHOULD BE ZERO
1749 004556 005703          67$: TST      R3
1750 004560 001401          BEQ      68$
1751 004562 104005          ERROR    5          :*****TEST 14 - ERROR 5*****
1752          :R3 SHOULD BE ZERO
1753 004564 020467 174064          68$: CMP      R4,DSTLN
1754 004570 001401          BEQ      69$
1755 004572 104006          ERROR    6          :*****TEST 14 - ERROR 6*****
1756          :R4 SHOULD STILL BE DESTINATION LENGTH
1757 004574 020567 174056          69$: CMP      R5,DSTAD
1758 004600 001401          BEQ      70$
1759 004602 104007          ERROR    7          :*****TEST 14 - ERROR 7*****
1760          :R5 SHOULD STILL BE DESTINATION ADDRESS
1761 004604 023706 000702          70$: CMP      @SAVR6,SP
1762 004610 001403          BEQ      71$
1763 004612 010637 000704          MOV      SP,@BADR6
1764 004616 104010          ERROR    10         :*****TEST 14 - ERROR 10*****
1765          :STACK POINTER NOT RESTORED BY INSTRUCTION
1766          :EXPECTED SP IS STORED AT 'SAVR6'
1767          :ERRONEOUS SP VALUE IS AT 'BADR6'
1768 004620          71$:

```

```

1769                                     :CHECK ANSWER
1770 004620 012700 004657                MOV #ANS14,R0      :POINT R0 TO EXPECTED ANSWER
1771 004624 016701 174026                MOV DSTAD,R1      :POINT R1 TO RESULTANT ANSWER
1772 004630 016702 174020                MOV DSTLN,R2      :STORE ANSWER LENGTH IN R1
1773 004634 122021 72$: CMPB (R0)+,(R1)+ :COMPARE EACH DIGIT
1774 004636 001401 BEQ 73$      :BR IF EQUAL
1775 004640 104011 ERROR 11              :*****TEST 14 - ERROR 11*****
1776                                     :ERRONEOUS ANSWER
1777                                     :R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1778                                     :R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1779 004642 005302 73$: DEC R2           :DECREMENT ANSWER LENGTH
1780 004644 001373 BNE 72$             :BR IF NOT FINISHED
1781 004646 000405 BR TST15          :BR TO NEXT TEST
1782 004650 S1T14:                       :SOURCE1 STRING
1783 004650 067 .BYTE 67                :MOST SIGNIFICANT DIGIT
1784 004651 066 .BYTE 66
1785 004652 064 .BYTE 64
1786 004653 S2T14:                       :SOURCE2 STRING
1787 004653 060 .BYTE 60                :MOST SIGNIFICANT DIGIT
1788 004654 070 .BYTE 70
1789 004655 063 .BYTE 63
1790 004656 171 .BYTE 171
1791 004657 ANS14:                       :EXPECTED ANSWER
1792 004657 067 .BYTE 67                :MOST SIGNIFICANT DIGIT
1793 004660 165 .BYTE 165
1794                                     :
1795 004662 .EVEN

```

```

1796
1797
1798 :*****
1799 :*TEST 15 TEST 'ADDN' WITH +SRC1 & -SRC2, S1L .GT. S2L, /S2 .GT. /S1/, OVERFLOW
1800 :*****
1801 004662 000004 1801: TST15: SCOPE
1802 004664 004567 010042 JSR R5,NPREP      :PREPARE ARGUMENTS FOR INSTRUCTION TEST
1803 004670 000004 4 :SOURCE1 LENGTH
1804 004672 005066 S1T15 :SOURCE1 ADDRESS
1805 004674 000003 3 :SOURCE2 LENGTH
1806 004676 005072 S2T15 :SOURCE2 ADDRESS
1807 004700 000002 2 :DESTINATION LENGTH
1808 004702 004767 010116 JSR PC,CLBUF     :CLEAR BUFFER AREA
1809 004706 004567 010132 JSR R5,XPSW
1810 004712 000206 .WORD 206
1811 004714 004767 010040 JSR PC,GENR      :SET UP GENERAL REGISTERS
1812 004720 000271 +SEN!SEC :SET UP THE COMPLEMENT OF EXPECTED CC'S
1813 004722 000246 +CLZ!CLV
1814 004724 076050 ADDN
1815
1816 004726 106767 173744 MFPS CCODES      :STORE RESULTANT PSW
1817 004732 042767 177400 173736 BIC #177400,CCODES :CLEAR UNUSED BITS
1818 004740 023767 000700 173730 CMP @#EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
1819 004746 001401 BEQ 64$      :BR, IF EQUAL
1820 004750 104001 ERROR 1              :*****TEST 15 - ERROR 1*****
1821                                     :PSW ERROR
1822                                     :EXPECTED PSW IS STORED AT 'SAVR6'
1823                                     :ERRONEOUS SP VALUE IS AT 'BADR6'
1824 004752 64$:

```


1825	004752	005700		TST	R0		:CHECK R0=0
1826	004754	001401		BEQ	65\$		
1827	004756	104002		ERROR	2		:*****TEST 15 - ERROR 2*****
1828							:R0 SHOULD BE ZERO
1829	004760	005701		65\$: TST	R1		:CHECK R1=0
1830	004762	001401		BEQ	66\$		
1831	004764	104003		ERROR	3		:*****TEST 15 - ERROR 3*****
1832							:R1 SHOULD BE ZERO
1833	004766	005702		66\$: TST	R2		:CHECK R2=0
1834	004770	001401		BEQ	67\$		
1835	004772	104004		ERROR	4		:*****TEST 15 - ERROR 4*****
1836							:R2 SHOULD BE ZERO
1837	004774	005703		67\$: TST	R3		:CHECK R3=0
1838	004776	001401		BEQ	68\$		
1839	005000	104005		ERROR	5		:*****TEST 15 - ERROR 5*****
1840							:R3 SHOULD BE ZERO
1841	005002	020467	173646	68\$: CMP	R4,DSTLN		:CHECK R4= DESTINATION LENGTH
1842	005006	001401		BEQ	69\$		
1843	005010	104006		ERROR	6		:*****TEST 15 - ERROR 6*****
1844							:R4 SHOULD STILL BE DESTINATION LENGTH
1845	005012	020567	173640	69\$: CMP	R5,DSTAD		:CHECK R5 = DESTINATION ADDRESS
1846	005016	001401		BEQ	70\$		
1847	005020	104007		ERROR	7		:*****TEST 15 - ERROR 7*****
1848							:R5 SHOULD STILL BE DESTINATION ADDRESS
1849	005022	023706	000702	70\$: CMP	@#SAVR6,SP		:VERIFY STACK POINTER IS RESTORED
1850	005026	001403		BEQ	71\$:BR IF OK
1851	005030	010637	000704	MOV	SP,@#BADR6		:COPY BAD SP VALUE
1852	005034	104010		ERROR	10		:*****TEST 15 - ERROR 10*****
1853							:STACK POINTER NOT RESTORED BY INSTRUCTION
1854							:EXPECTED SP IS STORED AT 'SAVR6'
1855							:ERRONEOUS SP VALUE IS AT 'BADR6'
1856	005036			71\$:			
1857							:CHECK ANSWER
1858	005036	012700	005075	MOV	#ANS15,R0		:POINT R0 TO EXPECTED ANSWER
1859	005042	016701	173610	MOV	DSTAD,R1		:POINT R1 TO RESULTANT ANSWER
1860	005046	016702	173602	MOV	DSTLN,R2		:STORE ANSWER LENGTH IN R1
1861	005052	122021		72\$: CMPB	(R0)+,(R1)+		:COMPARE EACH DIGIT
1862	005054	001401		BEQ	73\$:BR IF EQUAL
1863	005056	104011		ERROR	11		:*****TEST 15 - ERROR 11*****
1864							:ERRONEOUS ANSWER
1865							:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1866							:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1867	005060	005302		73\$: DEC	R2		:DECREMENT ANSWER LENGTH
1868	005062	001373		BNE	72\$:BR IF NOT FINISHED
1869	005064	000405		BR	TST16		:BR TO NEXT TEST
1870	005066			S1T15:			:SOURCE1 STRING
1871	005066	060		.BYTE	60		:MOST SIGNIFICANT DIGIT
1872	005067	060		.BYTE	60		
1873	005070	067		.BYTE	67		
1874	005071	163		.BYTE	163		
1875	005072			S2T15:			:SOURCE2 STRING
1876	005072	061		.BYTE	61		:MOST SIGNIFICANT DIGIT
1877	005073	067		.BYTE	67		
1878	005074	063		.BYTE	63		
1879	005075			ANS15:			:EXPECTED ANSWER
1880	005075	060		.BYTE	60		:MOST SIGNIFICANT DIGIT

```

1881 005076 060
1882
1883 005100
1884
1885
1886
1887
1888
1889
1890 005100 000004
1891 005102 004567 007624
1892 005106 000003
1893 005110 005304
1894 005112 000004
1895 005114 005307
1896 005116 000002
1897 005120 004767 007700
1898 005124 004567 007714
1899 005130 000206
1900 005132 004767 007622
1901 005136 000271
1902 005140 000246
1903 005142 076050
1904
1905 005144 106767 173526
1906 005150 042767 177400 173520
1907 005156 023767 000700 173512
1908 005164 001401
1909 005166 104001
1910
1911
1912
1913 005170
1914 005170 005700
1915 005172 001401
1916 005174 104002
1917
1918 005176 005701
1919 005200 001401
1920 005202 104003
1921
1922 005204 005702
1923 005206 001401
1924 005210 104004
1925
1926 005212 005703
1927 005214 001401
1928 005216 104005
1929
1930 005220 020467 173430
1931 005224 001401
1932 005226 104006
1933
1934 005230 020567 173422
1935 005234 001401
1936 005236 104007

```

```

:*****
:*TEST 16 TEST ADDN WITH -SRC1 & +SRC2,S1L .LT. S2L, /S1/ .GT. /S2/,OVERFLOW
:*****
TST16: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3 ;SOURCE1 LENGTH
S1T16 ;SOURCE1 ADDRESS
4 ;SOURCE2 LENGTH
S2T16 ;SOURCE2 ADDRESS
2 ;DESTINATION LENGTH
JSR PC,CLBUF ;CLEAR BUFFER AREA
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$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 16 - 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 16 - ERROR 2*****
;R0 SHOULD BE ZERO

65$: TST R1 ;CHECK R1=0
BEQ 66$
ERROR 3 ;*****TEST 16 - ERROR 3*****
;R1 SHOULD BE ZERO

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

67$: TST R3 ;CHECK R3=0
BEQ 68$
ERROR 5 ;*****TEST 16 - ERROR 5*****
;R3 SHOULD BE ZERO

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

69$: CMP R5,DSTAD ;CHECK R5 = DESTINATION ADDRESS
BEQ 70$
ERROR 7 ;*****TEST 16 - ERROR 7*****

```



```

1937
1938 005240 023706 000702      70$:  CMP      @#SAVR6,SP      ;R5 SHOULD STILL BE DESTINATION ADDRESS
1939 005244 001403              BEQ      71$              ;VERIFY STACK POINTER IS RESTORED
1940 005246 010637 000704      MOV      SP,@#BADR6      ;BR IF OK
1941 005252 104010              ERROR    10              ;COPY BAD SP VALUE
1942
1943
1944
1945 005254              71$:
1946
1947 005254 012700 005313      MOV      #ANS16,R0      ;CHECK ANSWER
1948 005260 016701 173372      MOV      DSTAD,R1      ;POINT R0 TO EXPECTED ANSWER
1949 005261 016702 173364      MOV      DSTLN,R2      ;POINT R1 TO RESULTANT ANSWER
1950 005270 122021              72$:  CMPB     (R0)+,(R1)+    ;STORE ANSWER LENGTH IN R1
1951 005272 001401              BEQ      73$              ;COMPARE EACH DIGIT
1952 005274 104011              ERROR    11              ;BR IF EQUAL
1953
1954
1955
1956 005276 005302              73$:  DEC      R2              ;*****TEST 16 - ERROR 11*****
1957 005300 001373              BNE     72$              ;ERRONEOUS ANSWER
1958 005302 000405              BR      TST17            ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
1959 005304              S1T16: .BYTE    61            ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
1960 005304 061              .BYTE    67            ;DECREMENT ANSWER LENGTH
1961 005305 067              .BYTE    162           ;BR IF NOT FINISHED
1962 005306 162              S2T16: .BYTE    60            ;BR TO NEXT TEST
1963 005307              .BYTE    60            ;SOURCE1 STRING
1964 005307 060              .BYTE    60            ;MOST SIGNIFICANT DIGIT
1965 005310 060              .BYTE    67            ;SOURCE2 STRING
1966 005311 067              .BYTE    67            ;MOST SIGNIFICANT DIGIT
1967 005312 062              .BYTE    62            ;EXPECTED ANSWER
1968 005313              ANS16: .BYTE    60            ;MOST SIGNIFICANT DIGIT
1969 005313 060              .BYTE    60
1970 005314 060              .BYTE    60
1971
1972              .EVEN
1973
1974
1975
1976
1977
1978 005316 000004              ;*****
1979 005320 004567 007406      TST17: SCOPE              ;*****TEST 17 TEST ADDN WITH +SRC1 & -SRC2, S1L=S2L, /S1/ .GT./S2/,OVERFLOW
1980 005324 000003              JSR      R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
1981 005326 005522              3              ;SOURCE1 LENGTH
1982 005330 000003              S1T17          ;SOURCE1 ADDRESS
1983 005332 005525              3              ;SOURCE2 LENGTH
1984 005334 000002              S2T17          ;SOURCE2 ADDRESS
1985 005336 004767 007462              2              ;DESTINATION LENGTH
1986 005342 004567 007476      JSR      PC,CLBUF      ;CLEAR BUFFER AREA
1987 005346 000202              JSR      R5,XPSW
1988 005350 004767 007404      .WORD    202
1989 005354 000277              JSR      PC,GENR      ;SET UP GENERAL REGISTERS
1990 005356 000242              SCC
1991 005360 076050              CLV
1992              ADDN              ;SET UP THE COMPLEMENT OF EXPECTED CC'S

```

```

1993 005362 106767 173310 MFPS CCODES ;STORE RESULTANT PSW
1994 005366 042767 177400 173302 BIC #177400,CCODES ;CLEAR UNUSED BITS
1995 005374 023767 000700 173274 CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
1996 005402 001401 BEQ 64$ ;BR, IF EQUAL
1997 005404 104001 ERROR 1 ;*****TEST 17 - ERROR 1*****
1998 ;PSW ERROR
1999 ;EXPECTED PSW IS STORED AT 'SAVR6'
2000 ;ERRONEOUS SP VALUE IS AT 'BADR6'
2001 005406 64$: TST R0 ;CHECK R0=0
2002 005406 005700 BEQ 65$
2003 005410 001401 ERROR 2 ;*****TEST 17 - ERROR 2*****
2004 005412 104002 ;R0 SHOULD BE ZERO
2005 ;CHECK R1=0
2006 005414 005701 65$: TST R1
2007 005416 001401 BEQ 66$
2008 005420 104003 ERROR 3 ;*****TEST 17 - ERROR 3*****
2009 ;R1 SHOULD BE ZERO
2010 005422 005702 66$: TST R2
2011 005424 001401 BEQ 67$
2012 005426 104004 ERROR 4 ;*****TEST 17 - ERROR 4*****
2013 ;R2 SHOULD BE ZERO
2014 005430 005703 67$: TST R3
2015 005432 001401 BEQ 68$
2016 005434 104005 ERROR 5 ;*****TEST 17 - ERROR 5*****
2017 ;R3 SHOULD BE ZERO
2018 005436 020467 173212 68$: CMP R4,DSTLN ;CHECK R4= DESTINATION LENGTH
2019 005442 001401 BEQ 69$
2020 005444 104006 ERROR 6 ;*****TEST 17 - ERROR 6*****
2021 ;R4 SHOULD STILL BE DESTINATION LENGTH
2022 005446 020567 173204 69$: CMP R5,DSTAD ;CHECK R5 = DESTINATION ADDRESS
2023 005452 001401 BEQ 70$
2024 005454 104007 ERROR 7 ;*****TEST 17 - ERROR 7*****
2025 ;R5 SHOULD STILL BE DESTINATION ADDRESS
2026 005456 023706 000702 70$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2027 005462 001403 BEQ 71$ ;BR IF OK
2028 005464 010637 000704 MOV SP,@#BADR6 ;COPY BAD SP VALUE
2029 005470 104010 ERROR 10 ;*****TEST 17 - ERROR 10*****
2030 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2031 ;EXPECTED SP IS STORED AT 'SAVR6'
2032 ;ERRONEOUS SP VALUE IS AT 'BADR6'
2033 005472 71$:
2034 ;CHECK ANSWER
2035 005472 012700 005530 MOV #ANS17,R0 ;POINT R0 TO EXPECTED ANSWER
2036 005476 016701 173154 MOV DSTAD,R1 ;POINT R1 TO RESULTANT ANSWER
2037 005502 016702 173146 MOV DSTLN,R2 ;STORE ANSWER LENGTH IN R1
2038 005506 122021 72$: CMPB (R0)+,(R1)+ ;COMPARE EACH DIGIT
2039 005510 001401 BEQ 73$ ;BR IF EQUAL
2040 005512 104011 ERROR 11 ;*****TEST 17 - ERROR 11*****
2041 ;ERRONEOUS ANSWER
2042 ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2043 ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2044 005514 005302 73$: DEC R2 ;DECREMENT ANSWER LENGTH
2045 005516 001373 BNE 72$ ;BR IF NOT FINISHED
2046 005520 000404 BR TST20 ;BR TO NEXT TEST
2047 005522 S1T17: ;SOURCE1 STRING
2048 005522 063 .BYTE 63 ;MOST SIGNIFICANT DIGIT

```


2049 005523 065
2050 005524 067
2051 005525
2052 005525 060
2053 005526 063
2054 005527 165
2055 005530
2056 005530 062
2057 005531 062
2058
2059
2060
2061
2062

S2T17:

.BYTE 65
.BYTE 67
.BYTE 60
.BYTE 63
.BYTE 165

:SOURCE2 STRING
:MOST SIGNIFICANT DIGIT

ANS17:

.BYTE 62
.BYTE 62

:EXPECTED ANSWER
:MOST SIGNIFICANT DIGIT

.EVEN

::*****
:*TEST 20 TEST ADDN WITH POSITIVE OPERANDS, OVERFLOW, NO CARRY OUT OF OVERFLOW
:*****

2064
2065 005532 000004
2066 005534 004567 007172
2067 005540 000003
2068 005542 005736
2069 005544 000003
2070 005546 005741
2071 005550 000002
2072 005552 004767 007246
2073 005556 004567 007262
2074 005562 000202
2075 005564 004767 007170
2076 005570 000277
2077 005572 000242
2078 005574 076050
2079

TST20:

SCOPE
JSR R5,NPREP
3
S1T20
3
S2T20
2
JSR PC,CLBUF
JSR R5,XPSW
.WORD 202
JSR PC,GENR
SCC
CLV
ADDN
MFPS
BIC #177400,CCODES
CMP @#EXPPSW,CCODES
BEQ 64\$
ERROR 1

:PREPARE ARGUMENTS FOR INSTRUCTION TEST
:SOURCE1 LENGTH
:SOURCE1 ADDRESS
:SOURCE2 LENGTH
:SOURCE2 ADDRESS
:DESTINATION LENGTH
:CLEAR BUFFER AREA

:SET UP GENERAL REGISTERS
:SET UP THE COMPLEMENT OF EXPECTED CC'S

2080 005576 106767 173074
2081 005602 042767 177400 173066
2082 005610 023767 000700 173060
2083 005616 001401
2084 005620 104001
2085
2086
2087

CCODES
#177400,CCODES
@#EXPPSW,CCODES
64\$
1

:STORE RESULTANT PSW
:CLEAR UNUSED BITS
:CHECK PSW AGAINST EXPECTED VALUE
:BR, IF EQUAL
:*****TEST 20 - ERROR 1*****
:PSW ERROR
:EXPECTED PSW IS STORED AT 'SAVR6'
:ERRONEOUS SP VALUE IS AT 'BADR6'

2088 005622
2089 005622 005700
2090 005624 001401
2091 005626 104002
2092

64\$:

TST R0
BEQ 65\$
ERROR 2

:CHECK R0=0
:*****TEST 20 - ERROR 2*****
:R0 SHOULD BE ZERO
:CHECK R1=0

2093 005630 005701
2094 005632 001401
2095 005634 104003
2096

65\$:

TST R1
BEQ 66\$
ERROR 3

:*****TEST 20 - ERROR 3*****
:R1 SHOULD BE ZERO
:CHECK R2=0

2097 005636 005702
2098 005640 001401
2099 005642 104004
2100

66\$:

TST R2
BEQ 67\$
ERROR 4

:*****TEST 20 - ERROR 4*****
:R2 SHOULD BE ZERO
:CHECK R3=0

2101 005644 005703
2102 005646 001401
2103 005650 104005
2104

67\$:

TST R3
BEQ 68\$
ERROR 5

:*****TEST 20 - ERROR 5*****
:R3 SHOULD BE ZERO

```

2105 005652 020467 172776      68$:  CMP      R4,DSTLN      ;CHECK R4= DESTINATION LENGTH
2106 005656 001401              BEQ      69$
2107 005660 104006              ERROR    6      ;*****TEST 20 - ERROR 6*****
2108                                ;R4 SHOULD STILL BE DESTINATION LENGTH
2109 005662 020567 172770      69$:  CMP      R5,DSTAD      ;CHECK R5 = DESTINATION ADDRESS
2110 005666 001401              BEQ      70$
2111 005670 104007              ERROR    7      ;*****TEST 20 - ERROR 7*****
2112                                ;R5 SHOULD STILL BE DESTINATION ADDRESS
2113 005672 023706 000702      70$:  CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
2114 005676 001403              BEQ      71$      ;BR IF OK
2115 005700 010637 000704      MOV      SP,@#BADR6      ;COPY BAD SP VALUE
2116 005704 104010              ERROR    10     ;*****TEST 20 - ERROR 10*****
2117                                ;STACK POINTER NOT RESTORED BY INSTRUCTION
2118                                ;EXPECTED SP IS STORED AT 'SAVR6'
2119                                ;ERRONEOUS SP VALUE IS AT 'BADR6'
2120 005706      71$:
2121                                ;CHECK ANSWER
2122 005706 012700 005744      MOV      #ANS20,R0      ;POINT R0 TO EXPECTED ANSWER
2123 005712 016701 172740      MOV      DSTAD,R1      ;POINT R1 TO RESULTANT ANSWER
2124 005716 016702 172732      MOV      DSTLN,R2      ;STORE ANSWER LENGTH IN R1
2125 005722 122021      72$:  CMPB     (R0)+,(R1)+    ;COMPARE EACH DIGIT
2126 005724 001401              BEQ      73$      ;BR IF EQUAL
2127 005726 104011              ERROR    11     ;*****TEST 20 - ERROR 11*****
2128                                ;ERRONEOUS ANSWER
2129                                ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2130                                ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2131 005730 005302      73$:  DEC      R2      ;DECREMENT ANSWER LENGTH
2132 005732 001373              BNE     72$      ;BR IF NOT FINISHED
2133 005734 000404              BR      TST21     ;BR TO NEXT TEST
2134 005736      S1T20:
2135 005736          070      .BYTE    70      ;SOURCE1 STRING
2136 005737          062      .BYTE    62      ;MOST SIGNIFICANT DIGIT
2137 005740          064      .BYTE    64
2138 005741      S2T20:
2139 005741          061      .BYTE    61      ;SOURCE2 STRING
2140 005742          070      .BYTE    70      ;MOST SIGNIFICANT DIGIT
2141 005743          065      .BYTE    65
2142 005744      ANS20:
2143 005744          060      .BYTE    60      ;EXPECTED ANSWER
2144 005745          071      .BYTE    71      ;MOST SIGNIFICANT DIGIT
2145
2146          .EVEN
2147
2148
2149
2150      ;*****
2151      ;*TEST 21      TEST INTERRUPTABILITY OF 'ADDN'
2152      ;*****
2153 005746 000004      TST21:  SCOPE
2154 005750 105777 172564      TSTB     @SWR      ;TEST BIT 7 OF SWR
2155 005754 100555      BMI     TST22     ;SKIP TO NEXT TEST IF SET
2156 005756 026767 172566 172700  CMP      $TPS,TCSR   ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
2157 005764 001007      BNE     T21CONT    ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
2158 005766 032767 000001 172612  BIT      #BIT0,$ENV  ;IF YES, IS PROGRAM UNDER APT?
2159 005774 001403      BEQ     T21CONT    ;BR, IF NOT
2160 005776 005767 172572      TST     $PASS      ;IF YES,CHECK IF NOT ON FIRST PASS
2160 006002 001142      BNE     TST22     ;IF NOT, BR & SKIP TEST

```



```

2161 006004          T21CONT:
2162 006004 004567 006722      JSR   R5,NPREP      ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2163 006010 000004          4      ;SOURCE1 LENGTH
2164 006012 006260          S1T21      ;SOURCE1 ADDRESS
2165 006014 000004          4      ;SOURCE2 LENGTH
2166 006016 006264          S2T21      ;SOURCE2 ADDRESS
2167 006020 000005          5      ;DESTINATION LENGTH
2168 006022 004767 006776      JSR   PC,CLBUF     ;CLEAR BUFFER AREA
2169 006026 012767 006106 172640  MOV   #ADDNPC,PCI  ;STORE PC OF TEST INSTRUCTION
2170 006034 012777 015114 172626  MOV   #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
2171 006042 005077 172624      CLR   @TPSW        ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
2172 006046 004767 007016      JSR   PC,TDONE     ;WAIT FOR SLU READY
2173 006052 013777 000554 172606  MOV   @#NULL,@TBUF ;SEND NULL CHARACTER
2174 006060 004567 006760      JSR   R5,XPSW     ;STORE EXPECTED PSW
2175 006064 000000          .WORD 0
2176 006066 106427 000000      MTPS #0           ;ALLOW INTERRUPTS
2177 006072 052777 000100 172564  BIS   #100,@TCSR  ;ENABLE TTY INTERRUPTS
2178 006100 004767 006654      READDN: JSR  PC,GENR ;SET UP GENERAL REGISTERS
2179 006104 000277          SCC
2180 006106 076050          ADDNPC: ADDN      ;SET UP THE COMPLEMENT OF EXPECTED CC'S
2181
2182 006110 106767 172562      MFPS  CCODES      ;STORE RESULTANT PSW
2183 006114 032777 000100 172542  BIT   #100,@TCSR  ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
2184 006122 001366          BNE   READDN      ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
2185 006124 042767 177400 172544  BIC   #177400,CCODES ;CLEAR UNUSED BITS
2186 006132 023767 000700 172536  CMP   @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2187 006140 001401          BEQ   64$        ;BR, IF EQUAL
2188 006142 104001          ERROR 1        ;*****TEST 21 - ERROR 1*****
2189
2190
2191
2192
2192 006144          64$:
2193 006144 005700          TST   R0          ;CHECK R0=0
2194 006146 001401          BEQ   65$
2195 006150 104002          ERROR 2        ;*****TEST 21 - ERROR 2*****
2196
2197 006152 005701          65$: TST   R1          ;R0 SHOULD BE ZERO
2198 006154 001401          BEQ   66$        ;CHECK R1=0
2199 006156 104003          ERROR 3        ;*****TEST 21 - ERROR 3*****
2200
2201 006160 005702          66$: TST   R2          ;R1 SHOULD BE ZERO
2202 006162 001401          BEQ   67$        ;CHECK R2=0
2203 006164 104004          ERROR 4        ;*****TEST 21 - ERROR 4*****
2204
2205 006166 005703          67$: TST   R3          ;R2 SHOULD BE ZERO
2206 006170 001401          BEQ   68$        ;CHECK R3=0
2207 006172 104005          ERROR 5        ;*****TEST 21 - ERROR 5*****
2208
2209 006174 020467 172454          68$: CMP   R4,DSTLN   ;R3 SHOULD BE ZERO
2210 006200 001401          BEQ   69$        ;CHECK R4= DESTINATION LENGTH
2211 006202 104006          ERROR 6        ;*****TEST 21 - ERROR 6*****
2212
2213 006204 020567 172446          69$: CMP   R5,DSTAD   ;R4 SHOULD STILL BE DESTINATION LENGTH
2214 006210 001401          BEQ   70$        ;CHECK R5 = DESTINATION ADDRESS
2215 006212 104007          ERROR 7        ;*****TEST 21 - ERROR 7*****
2216

```

```

2217 006214 023706 000702 70$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2218 006220 001403 BEQ 71$ ;BR IF OK
2219 006222 010637 000704 MOV SP,@#BADR6 ;COPY BAD SP VALUE
2220 006226 104010 ERROR 10 ;*****TEST 21 - ERROR 10*****
2221 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2222 ;EXPECTED SP IS STORED AT "SAVR6"
2223 ;ERRONEOUS SP VALUE IS AT "BADR6"
2224 006230 71$:
2225 ;CHECK ANSWER
2226 006230 012700 006270 MOV #ANS21,R0 ;POINT R0 TO EXPECTED ANSWER
2227 006234 016701 172416 MOV DSTAD,R1 ;POINT R1 TO RESULTANT ANSWER
2228 006240 016702 172410 MOV DSTLN,R2 ;STORE ANSWER LENGTH IN R1
2229 006244 122021 72$: CMPB (R0)+,(R1)+ ;COMPARE EACH DIGIT
2230 006246 001401 BEQ 73$ ;BR IF EQUAL
2231 006250 104011 ERROR 11 ;*****TEST 21 - ERROR 11*****
2232 ;ERRONEOUS ANSWER
2233 ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2234 ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2235 006252 005302 73$: DEC R2 ;DECREMENT ANSWER LENGTH
2236 006254 001373 BNE 72$ ;BR IF NOT FINISHED
2237 006256 000407 BR ENDT21 ;BR TO END OF THIS TEST
2238 006260 S1T21: ;SOURCE1 STRING
2239 006260 061 .BYTE 61 ;MOST SIGNIFICANT DIGIT
2240 006261 062 .BYTE 62
2241 006262 063 .BYTE 63
2242 006263 064 .BYTE 64
2243 006264 S2T21: ;SOURCE2 STRING
2244 006264 065 .BYTE 65 ;MOST SIGNIFICANT DIGIT
2245 006265 066 .BYTE 66
2246 006266 067 .BYTE 67
2247 006267 070 .BYTE 70
2248 006270 ANS21: ;EXPECTED ANSWER
2249 006270 060 .BYTE 60 ;MOST SIGNIFICANT DIGIT
2250 006271 066 .BYTE 66
2251 006272 071 .BYTE 71
2252 006273 061 .BYTE 61
2253 006274 062 .BYTE 62
2254
2255 006276 .EVEN
2256 006276 016777 172370 172364 ENDT21: MOV TPSW,@TVECT
2257 006304 106427 000200 MTPS #200
2258
2259
2260
2261 ;*****
2262 ;*TEST 22 TEST "SUBN" WITH POSITIVE OPERANDS, SRC1 .GT. SRC2
2263 ;*****
2263 006310 000004 TST22: SCOPE
2264 006312 004567 006414 JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2265 006316 000004 4 ;SOURCE1 LENGTH
2266 006320 006514 S1T22 ;SOURCE1 ADDRESS
2267 006322 000003 3 ;SOURCE2 LENGTH
2268 006324 006520 S2T22 ;SOURCE2 ADDRESS
2269 006326 000003 3 ;DESTINATION LENGTH
2270 006330 004767 006470 JSR PC,CLBUF ;CLEAR BUFFER AREA
2271 006334 004567 006504 JSR R5,XPSW
2272 006340 000210 .WORD 210

```



```

2273 006342 004767 006412 JSR PC,GENR ;SET UP GENERAL REGISTERS
2274 006346 000277 SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
2275 006350 000250 CLN
2276 006352 076051 SUBN
2277
2278 006354 106767 172316 MFPS CCODES ;STORE RESULTANT PSW
2279 006360 042767 177400 172310 BIC #177400,CCODES ;CLEAR UNUSED BITS
2280 006366 023767 000700 172302 CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2281 006374 001401 BEQ 64$ ;BR, IF EQUAL
2282 006376 104001 ERROR 1 ;*****TEST 22 - ERROR 1*****
2283 ;PSW ERROR
2284 ;EXPECTED PSW IS STORED AT 'SAVR6'
2285 ;ERRONEOUS SP VALUE IS AT 'BADR6'
2286 006400 64$: TST R0 ;CHECK R0=0
2287 006400 005700 BEQ 65$
2288 006402 001401 ERROR 2 ;*****TEST 22 - ERROR 2*****
2289 006404 104002 ;R0 SHOULD BE ZERO
2290 ;CHECK R1=0
2291 006406 005701 65$: TST R1
2292 006410 001401 BEQ 66$
2293 006412 104003 ERROR 3 ;*****TEST 22 - ERROR 3*****
2294 ;R1 SHOULD BE ZERO
2295 006414 005702 66$: TST R2
2296 006416 001401 BEQ 67$
2297 006420 104004 ERROR 4 ;*****TEST 22 - ERROR 4*****
2298 ;R2 SHOULD BE ZERO
2299 006422 005703 67$: TST R3
2300 006424 001401 BEQ 68$
2301 006426 104005 ERROR 5 ;*****TEST 22 - ERROR 5*****
2302 ;R3 SHOULD BE ZERO
2303 006430 020467 172220 68$: CMP R4,DSTLN
2304 006434 001401 BEQ 69$
2305 006436 104006 ERROR 6 ;*****TEST 22 - ERROR 6*****
2306 ;R4 SHOULD STILL BE DESTINATION LENGTH
2307 006440 020567 172212 69$: CMP R5,DSTAD
2308 006444 001401 BEQ 70$
2309 006446 104007 ERROR 7 ;*****TEST 22 - ERROR 7*****
2310 ;R5 SHOULD STILL BE DESTINATION ADDRESS
2311 006450 023706 000702 70$: CMP @#SAVR6,SP
2312 006454 001403 BEQ 71$
2313 006456 010637 000704 MOV SP,@#BADR6
2314 006462 104010 ERROR 10 ;*****TEST 22 - ERROR 10*****
2315 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2316 ;EXPECTED SP IS STORED AT 'SAVR6'
2317 ;ERRONEOUS SP VALUE IS AT 'BADR6'
2318 006464 71$:
2319 ;CHECK ANSWER
2320 006464 012700 006523 MOV #ANS22,R0 ;POINT R0 TO EXPECTED ANSWER
2321 006470 016701 172162 MOV DSTAD,R1 ;POINT R1 TO RESULTANT ANSWER
2322 006474 016702 172154 MOV DSTLN,R2 ;STORE ANSWER LENGTH IN R1
2323 006500 122021 72$: CMPB (R0)+,(R1)+ ;COMPARE EACH DIGIT
2324 006502 001401 BEQ 73$
2325 006504 104011 ERROR 11 ;*****TEST 22 - ERROR 11*****
2326 ;ERRONEOUS ANSWER
2327 ;R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2328 ;R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT

```

```

2329 006506 005302      73$:  DEC      R2      ;DECREMENT ANSWER LENGTH
2330 006510 001373      BNE      72$      ;BR IF NOT FINISHED
2331 006512 000405      BR       TST23    ;BR TO NEXT TEST
2332 006514           S1T22:           ;SOURCE1 STRING
2333 006514      060    .BYTE      60      ;MOST SIGNIFICANT DIGIT
2334 006515      071    .BYTE      71
2335 006516      066    .BYTE      66
2336 006517      062    .BYTE      62
2337 006520           S2T22:           ;SOURCE2 STRING
2338 006520      067    .BYTE      67      ;MOST SIGNIFICANT DIGIT
2339 006521      065    .BYTE      65
2340 006522      064    .BYTE      64
2341 006523           ANS22:           ;EXPECTED ANSWER
2342 006523      062    .BYTE      62      ;MOST SIGNIFICANT DIGIT
2343 006524      060    .BYTE      60
2344 006525      170    .BYTE     170
2345           .EVEN
2346
2347
2348
2349
2350
2351 006526 000004           ::*****
2352 006530 004567 006176  *TEST 23      TEST 'SUBN' WITH NEGATIVE OPERANDS, SRC1 .GTT. SRC2
2353 006534 000003           ::*****
2354 006536 006730      TST23:  SCOPE
2355 006540 000004      JSR      R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2356 006542 006733      3        ;SOURCE1 LENGTH
2357 006544 000003      S1T23    ;SOURCE1 ADDRESS
2358 006546 004767 006252  4        ;SOURCE2 LENGTH
2359 006552 004567 006266  S2T23    ;SOURCE2 ADDRESS
2360 006556 000200      3        ;DESTINATION LENGTH
2361 006560 004767 006174  JSR      PC,CLBUF ;CLEAR BUFFER AREA
2362 006564 000277      JSR      R5,XPSW
2363 006566 076051      .WORD    200
2364           JSR      PC,GENR ;SET UP GENERAL REGISTERS
2365 006570 106767 172102  SCC      ;SET UP THE COMPLEMENT OF EXPECTED CC'S
2366 006574 042767 177400 172074  SUBN
2367 006602 023767 000700 172066  MFPS     CCODES   ;STORE RESULTANT PSW
2368 006610 001401      BIC     #177400,CCODES ;CLEAR UNUSED BITS
2369 006612 104001      CMP     @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2370           BEQ     64$   ;BR, IF EQUAL
2371           ERROR    1    ;*****TEST 23 - ERROR 1*****
2372           ;PSW ERROR
2373           ;EXPECTED PSW IS STORED AT 'SAVR6'
2374           ;ERRONEOUS SP VALUE IS AT 'BADR6'
2375 006614           64$:  TST     R0      ;CHECK R0=0
2376 006616 001401      BEQ     65$
2377 006620 104002      ERROR   2        ;*****TEST 23 - ERROR 2*****
2378 006622 005701      ;R0 SHOULD BE ZERO
2379 006624 001401      65$:  TST     R1      ;CHECK R1=0
2380 006626 104003      BEQ     66$
2381           ERROR   3        ;*****TEST 23 - ERROR 3*****
2382 006630 005702      ;R1 SHOULD BE ZERO
2383 006632 001401      66$:  TST     R2      ;CHECK R2=0
2384 006634 104004      BEQ     67$
                ERROR   4        ;*****TEST 23 - ERROR 4*****

```



```

2385                                     :R2 SHOULD BE ZERO
2386 006636 005703                       67$: TST      R3                      :CHECK R3=0
2387 006640 001401                       BEQ      68$
2388 006642 104005                       ERROR    5
2389                                     :*****TEST 23 - ERROR 5*****
2390 006644 020467 172004                 68$: CMP      R4,DSTLN                :R3 SHOULD BE ZERO
2391 006650 001401                       BEQ      69$                      :CHECK R4= DESTINATION LENGTH
2392 006652 104006                       ERROR    6
2393                                     :*****TEST 23 - ERROR 6*****
2394 006654 020567 171776                 69$: CMP      R5,DSTAD                :R4 SHOULD STILL BE DESTINATION LENGTH
2395 006660 001401                       BEQ      70$                      :CHECK R5 = DESTINATION ADDRESS
2396 006662 104007                       ERROR    7
2397                                     :*****TEST 23 - ERROR 7*****
2398 006664 023706 000702                 70$: CMP      @SAVR6,SP                :R5 SHOULD STILL BE DESTINATION ADDRESS
2399 006670 001403                       BEQ      71$                      :VERIFY STACK POINTER IS RESTORED
2400 006672 010637 000704                 MOV      SP,@BADR6                :BR IF OK
2401 006676 104010                       ERROR    10                       :COPY BAD SP VALUE
2402                                     :*****TEST 23 - ERROR 10*****
2403                                     :STACK POINTER NOT RESTORED BY INSTRUCTION
2404                                     :EXPECTED SP IS STORED AT "SAVR6"
2405 006700                               71$:                                     :ERRONEOUS SP VALUE IS AT "BADR6"
2406                                     :CHECK ANSWER
2407 006700 012700 006737                 MOV      #ANS23,R0                :POINT R0 TO EXPECTED ANSWER
2408 006704 016701 171746                 MOV      DSTAD,R1                 :POINT R1 TO RESULTANT ANSWER
2409 006710 016702 171740                 MOV      DSTLN,R2                 :STORE ANSWER LENGTH IN R1
2410 006714 122021                       CMPB     (R0)+,(R1)+              :COMPARE EACH DIGIT
2411 006716 001401                       BEQ      73$                      :BR IF EQUAL
2412 006720 104011                       ERROR    11                       :*****TEST 23 - ERROR 11*****
2413                                     :ERRONEOUS ANSWER
2414                                     :R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
2415                                     :R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
2416 006722 005302                       73$: DEC      R2                      :DECREMENT ANSWER LENGTH
2417 006724 001373                       BNE     72$                      :BR IF NOT FINISHED
2418 006726 000405                       BR      TST24                    :BR TO NEXT TEST
2419 006730                               S1T23:                             :SOURCE1 STRING
2420 006730 071                            .BYTE   71                       :MOST SIGNIFICANT DIGIT
2421 006731 066                            .BYTE   66
2422 006732 162                            .BYTE   162
2423 006733                               S2T23:                             :SOURCE2 STRING
2424 006733 060                            .BYTE   60                       :MOST SIGNIFICANT DIGIT
2425 006734 070                            .BYTE   70
2426 006735 063                            .BYTE   63
2427 006736 161                            .BYTE   161
2428 006737                               ANS23:                             :EXPECTED ANSWER
2429 006737 061                            .BYTE   61                       :MOST SIGNIFICANT DIGIT
2430 006740 063                            .BYTE   63
2431 006741 061                            .BYTE   61
2432                                     .EVEN
2433
2434
2435 :*****
2436 :*TEST 24 TEST INTERRUPTABILITY OF "SUBN"
2437 :*****
2438 006742 000004                               TST24: SCOPE
2439 006744 105777 171570                 TSTB    @SWR                      :TEST BIT 7 OF SWR
2440 006750 100556                         BMI     TST25                    :SKIP TO NEXT TEST IF SET

```

```

2441 006752 026767 171572 171704      CMP      STPS,TCSR      ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
2442 006760 001007                    BNE      T24CONT      ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
2443 006762 032767 000001 171616      BIT      #BIT0,$ENV   ;IF YES, IS PROGRAM UNDER APT?
2444 006770 001403                    BEQ      T24CONT      ;BR, IF NOT
2445 006772 005767 171576                    TST      $PASS       ;IF YES,CHECK IF NOT ON FIRST PASS
2446 006776 001143                    BNE      TST25       ;IF NOT, BR & SKIP TEST
2447 007000                    T24CONT:
2448 007000 004567 005726                    JSR      R5,NPREP    ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2449 007004 000004                    4        ;SOURCE1 LENGTH
2450 007006 007256                    S1T24     ;SOURCE1 ADDRESS
2451 007010 000004                    4        ;SOURCE2 LENGTH
2452 007012 007262                    S2T24     ;SOURCE2 ADDRESS
2453 007014 000005                    5        ;DESTINATION LENGTH
2454 007016 004767 006002                    JSR      PC,CLBUF    ;CLEAR BUFFER AREA
2455 007022 012767 007104 171644                    MOV      #SUBNPC,PCI ;STORE PC OF TEST INSTRUCTION
2456 007030 012777 015114 171632                    MOV      #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
2457 007036 005077 171630                    CLR      @TPSW       ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
2458 007042 004767 006022                    JSR      PC,TDONE    ;WAIT FOR SLU READY
2459 007046 013777 000554 171612                    MOV      @#$NULL,@TBUF ;SEND NULL CHARACTER
2460 007054 004567 005764                    JSR      R5,XPSW     ;STORE EXPECTED PSW
2461 007060 000010                    .WORD    10
2462 007062 106427 000000                    MTPS     #0          ;ALLOW INTERRUPTS
2463 007066 052777 000100 171570                    BIS      #100,@TCSR  ;ENABLE TTY INTERRUPTS
2464 007074 004767 005660                    RESUBN: JSR      PC,GENR ;SET UP GENERAL REGISTERS
2465 007100 000277                    SCC
2466 007102 000250                    CLN
2467 007104 076051                    SUBNPC: SUBN
2468
2469 007106 106767 171564                    MFPS     CCODES     ;STORE RESULTANT PSW
2470 007112 032777 000100 171544                    BIT      #100,@TCSR  ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
2471 007120 001365                    BNE      RESUBN     ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
2472 007122 042767 177400 171546                    BIC      #177400,CCODES ;CLEAR UNUSED BITS
2473 007130 023767 000700 171540                    CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2474 007136 001401                    BEQ      64$        ;BR, IF EQUAL
2475 007140 104001                    ERROR    1          ;*****TEST 24 - ERROR 1*****
2476
2477
2478
2479
2480 007142 005700                    64$:    TST      R0        ;CHECK R0=0
2481 007144 001401                    BEQ      65$
2482 007146 104002                    ERROR    2          ;*****TEST 24 - ERROR 2*****
2483
2484 007150 005701                    65$:    TST      R1        ;CHECK R1=0
2485 007152 001401                    BEQ      66$
2486 007154 104003                    ERROR    3          ;*****TEST 24 - ERROR 3*****
2487
2488 007156 005702                    66$:    TST      R2        ;CHECK R2=0
2489 007160 001401                    BEQ      67$
2490 007162 104004                    ERROR    4          ;*****TEST 24 - ERROR 4*****
2491
2492 007164 005703                    67$:    TST      R3        ;CHECK R3=0
2493 007166 001401                    BEQ      68$
2494 007170 104005                    ERROR    5          ;*****TEST 24 - ERROR 5*****
2495
2496 007172 020467 171456                    68$:    CMP      R4,DSTLN ;CHECK R4= DESTINATION LENGTH

```



```

2497 007176 001401 BEQ 69$
2498 007200 104006 ERROR 6
2499
2500 007202 020567 171450 69$: CMP R5,DSTAD
2501 007206 001401 BEQ 70$
2502 007210 104007 ERROR 7
2503
2504 007212 023706 000702 70$: CMP @#SAVR6,SP
2505 007216 001403 BEQ 71$
2506 007220 010637 000704 MOV SP,@#BADR6
2507 007224 104010 ERROR 10
2508
2509
2510
2511 007226 71$:
2512
2513 007226 012700 007266 MOV #ANS24,R0
2514 007232 016701 171420 MOV DSTAD,R1
2515 007236 016702 171412 MOV DSTLN,R2
2516 007242 122021 72$: CMPB (R0)+,(R1)+
2517 007244 001401 BEQ 73$
2518 007246 104011 ERROR 11
2519
2520
2521
2522 007250 005302 73$: DEC R2
2523 007252 001373 BNE 72$
2524 007254 000407 BR ENDT24
2525 007256 S1T24:
2526 007256 061 .BYTE 61
2527 007257 062 .BYTE 62
2528 007260 063 .BYTE 63
2529 007261 064 .BYTE 64
2530 007262 S2T24:
2531 007262 065 .BYTE 65
2532 007263 066 .BYTE 66
2533 007264 067 .BYTE 67
2534 007265 170 .BYTE 170
2535 007266 ANS24:
2536 007266 060 .BYTE 60
2537 007267 066 .BYTE 66
2538 007270 071 .BYTE 71
2539 007271 061 .BYTE 61
2540 007272 162 .BYTE 162
2541
2542
2543 007274 007274 .EVEN
2544 007302 016777 171372 171366 ENDT24: MOV TPSW,@TVECT
2545 106427 000200 MTPS #200
2546
2547
2548
2549
2550 007306 000004
2551 007310 004567 005416
2552 007314 000003

```

*****TEST 24 - ERROR 6*****
:R4 SHOULD STILL BE DESTINATION LENGTH
:CHECK R5 = DESTINATION ADDRESS

*****TEST 24 - ERROR 7*****
:R5 SHOULD STILL BE DESTINATION ADDRESS
:VERIFY STACK POINTER IS RESTORED
:BR IF OK
:COPY BAD SP VALUE

*****TEST 24 - ERROR 10*****
:STACK POINTER NOT RESTORED BY INSTRUCTION
:EXPECTED SP IS STORED AT "SAVR6"
:ERRONEOUS SP VALUE IS AT "BADR6"

:CHECK ANSWER
:POINT R0 TO EXPECTED ANSWER
:POINT R1 TO RESULTANT ANSWER
:STORE ANSWER LENGTH IN R1
:COMPARE EACH DIGIT
:BR IF EQUAL

*****TEST 24 - ERROR 11*****
:ERRONEOUS ANSWER
:R0 CONTAINS THE PC+1 OF THE EXPECTED DIGIT
:R1 CONTAINS THE PC+1 OF THE RESULTANT DIGIT
:DECREMENT ANSWER LENGTH
:BR IF NOT FINISHED
:BR TO END OF THIS TEST

:SOURCE1 STRING
:MOST SIGNIFICANT DIGIT

:SOURCE2 STRING
:MOST SIGNIFICANT DIGIT

:EXPECTED ANSWER
:MOST SIGNIFICANT DIGIT

:TEST 25 TEST "CMPN" WITH ALL ZEROES IN BOTH SOURCE STRINGS

TST25: SCOPE JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3 ;SOURCE1 LENGTH

```

2553 007316 007460          S1T25 ;SOURCE1 ADDRESS
2554 007320 000003          3      ;SOURCE2 LENGTH
2555 007322 007463          S2T25 ;SOURCE2 ADDRESS
2556 007324 000377          377   ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
2557 007326 004567 005512  JSR    R5,XPSW
2558 007332 000204          .WORD 204
2559 007334 004767 005420  JSR    PC,GENR ;SET UP GENERAL REGISTERS
2560 007340 000277          SCC    ;SET UP THE COMPLEMENT OF EXPECTED CC'S
2561 007342 000244          CLZ
2562
2563 007344 076052          CMPN
2564
2565 007346 106767 171324  MFPS  CCODES ;STORE RESULTANT PSW
2566 007352 042767 177400 171316 BIC    #177400,CCODES ;CLEAR UNUSED BITS
2567 007360 023767 000700 171310 CMP    @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2568 007366 001401          BEQ   64$ ;BR, IF EQUAL
2569 007370 104001          ERROR 1 ;*****TEST 25 - ERROR 1*****
2570 ;PSW ERROR
2571 ;EXPECTED PSW IS STORED AT "SAVR6"
2572 ;ERRONEOUS SP VALUE IS AT "BADR6"
2573          64$:
2574 007372 005700          TST   R0 ;CHECK R0=0
2575 007374 001401          BEQ   65$
2576 007376 104002          ERROR 2 ;*****TEST 25 - ERROR 2*****
2577 ;R0 SHOULD BE ZERO
2578 007400 005701          65$: TST   R1 ;CHECK R1=0
2579 007402 001401          BEQ   66$
2580 007404 104003          ERROR 3 ;*****TEST 25 - ERROR 3*****
2581 ;R1 SHOULD BE ZERO
2582 007406 005702          66$: TST   R2 ;CHECK R2=0
2583 007410 001401          BEQ   67$
2584 007412 104004          ERROR 4 ;*****TEST 25 - ERROR 4*****
2585 ;R2 SHOULD BE ZERO
2586 007414 005703          67$: TST   R3 ;CHECK R3=0
2587 007416 001401          BEQ   68$
2588 007420 104005          ERROR 5 ;*****TEST 25 - ERROR 5*****
2589 ;R3 SHOULD BE ZERO
2590 007422 020467 171226          68$: CMP    R4,DSTLN ;CHECK R4 UNCHANGED
2591 007426 001401          BEQ   69$ ;BR IF OK
2592 007430 104006          ERROR 6 ;*****TEST 25 - ERROR 6*****
2593 ;R4 CHANGED
2594 ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
2595 007432 020567 171220          69$: CMP    R5,DSTAD ;CHECK R5 UNCHANGED
2596 007436 001401          BEQ   70$
2597 007440 104007          ERROR 7 ;*****TEST 25 - ERROR 7*****
2598 ;R5 CHANGED
2599 ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
2600 007442 023706 000702          70$: CMP    @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2601 007446 001403          BEQ   71$ ;BR IF OK
2602 007450 010637 000704          MOV   SP,@#BADR6 ;COPY BAD SP VALUE
2603 007454 104010          ERROR 10 ;*****TEST 25 - ERROR 10*****
2604 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2605 ;EXPECTED SP IS STORED AT "SAVR6"
2606 ;ERRONEOUS SP VALUE IS AT "BADR6"
2607 007456 000403          71$: BR    TST26 ;BR TO NEXT TEST
2608

```


2609 007460
 2610 007460 060
 2611 007461 060
 2612 007462 060
 2613 007463
 2614 007463 060
 2615 007464 060
 2616 007465 060
 2617
 2618
 2619
 2620
 2621
 2622
 2623
 2624 007466 000004
 2625 007470 004567 005236
 2626 007474 000004
 2627 007476 007640
 2628 007500 000004
 2629 007502 007644
 2630 007504 000377
 2631 007506 004567 005332
 2632 007512 000204
 2633 007514 004767 005240
 2634 007520 000277
 2635 007522 000244
 2636
 2637 007524 076052
 2638
 2639 007526 106767 171144
 2640 007532 042767 177400 171136
 2641 007540 023767 000700 171130
 2642 007546 001401
 2643 007550 104001
 2644
 2645
 2646
 2647 007552
 2648 007552 005700
 2649 007554 001401
 2650 007556 104002
 2651
 2652 007560 005701
 2653 007562 001401
 2654 007564 104003
 2655
 2656 007566 005702
 2657 007570 001401
 2658 007572 104004
 2659
 2660 007574 005703
 2661 007576 001401
 2662 007600 104005
 2663
 2664 007602 020467 171046

S1T25: :SOURCE1 STRING
 :MOST SIGNIFICANT DIGIT
 .BYTE 60
 .BYTE 60
 .BYTE 60
 S2T25: :SOURCE2 STRING
 :MOST SIGNIFICANT DIGIT
 .BYTE 60
 .BYTE 60
 .BYTE 60
 .EVEN

 :*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,GENR :SET UP GENERAL REGISTERS
 SCC :SET UP THE COMPLEMENT OF EXPECTED CC'S
 CLZ
 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 'BADR6'
 64\$: TST R0 :CHECK R0=0
 BEQ 65\$
 ERROR 2 :*****TEST 26 - ERROR 2*****
 :R0 SHOULD BE ZERO
 65\$: TST R1 :CHECK R1=0
 BEQ 66\$
 ERROR 3 :*****TEST 26 - ERROR 3*****
 :R1 SHOULD BE ZERO
 66\$: TST R2 :CHECK R2=0
 BEQ 67\$
 ERROR 4 :*****TEST 26 - ERROR 4*****
 :R2 SHOULD BE ZERO
 67\$: TST R3 :CHECK R3=0
 BEQ 68\$
 ERROR 5 :*****TEST 26 - ERROR 5*****
 :R3 SHOULD BE ZERO
 68\$: CMP R4,DSTLN :CHECK R4 UNCHANGED


```

2721                                     :ERRONEOUS SP VALUE IS AT 'BADR6'
2722 007732                               64$: TST R0                               :CHECK R0=0
2723 007732 005700                       BEQ 65$                               :*****TEST 27 - ERROR 2*****
2724 007734 001401                       ERROR 2                               :R0 SHOULD BE ZERO
2725 007736 104002                                     :CHECK R1=0
2726                                     :*****TEST 27 - ERROR 3*****
2727 007740 005701                               65$: TST R1                               :R1 SHOULD BE ZERO
2728 007742 001401                       BEQ 66$                               :CHECK R2=0
2729 007744 104003                       ERROR 3                               :*****TEST 27 - ERROR 4*****
2730                                     :R2 SHOULD BE ZERO
2731 007746 005702                               66$: TST R2                               :CHECK R3=0
2732 007750 001401                       BEQ 67$                               :*****TEST 27 - ERROR 5*****
2733 007752 104004                       ERROR 4                               :R3 SHOULD BE ZERO
2734                                     :CHECK R4 UNCHANGED
2735 007754 005703                               67$: TST R3                               :BR IF OK
2736 007756 001401                       BEQ 68$                               :*****TEST 27 - ERROR 6*****
2737 007760 104005                       ERROR 5                               :R4 CHANGED
2738                                     :R4 SHOULD STILL EQUAL CONTENTS OF 'FILL'
2739 007762 020467 170666                 68$: CMP R4,DSTLN                       :CHECK R5 UNCHANGED
2740 007766 001401                       BEQ 69$                               :*****TEST 27 - ERROR 7*****
2741 007770 104006                       ERROR 6                               :R5 CHANGED
2742                                     :R5 SHOULD STILL EQUAL CONTENTS OF 'DSTAD'
2743                                     :VERIFY STACK POINTER IS RESTORED
2744 007772 020567 170660                 69$: CMP R5,DSTAD                       :BR IF OK
2745 007776 001401                       BEQ 70$                               :COPY BACK SP VALUE
2746 010000 104007                       ERROR 7                               :*****TEST 27 - ERROR 10*****
2747                                     :STACK POINTER NOT RESTORED BY INSTRUCTION
2748                                     :EXPECTED SP IS STORED AT 'SAVR6'
2749 010002 023706 000702                 70$: CMP @SAVR6,SP                       :ERRONEOUS SP VALUE IS AT 'BADR6'
2750 010006 001403                       BEQ 71$
2751 010010 010637 000704                 MOV SP,@BADR6
2752 010014 104010                       ERROR 10
2753
2754
2755
2756 010016                               71$: BR TST30                               :BR TO NEXT TEST
2757 010016 000404                               S1T27: .BYTE 71                               :SOURCE1 STRING
2758 010020                               .BYTE 67                               :MOST SIGNIFICANT DIGIT
2759 010020 071                               .BYTE 65
2760 010021 067                               .BYTE 63
2761 010022 065                               S2T27: .BYTE 71                               :SOURCE2 STRING
2762 010023 063                               .BYTE 67                               :MOST SIGNIFICANT DIGIT
2763 010024                               .BYTE 65
2764 010024 071                               .BYTE 163
2765 010025 067
2766 010026 065
2767 010027 163
2768
2769 .EVEN
2770
2771
2772
2773 :*****
2774 :*TEST 30 TEST "CMPN" WITH IDENTICAL NON-ZERO MAGNITUDES, POSITIVE SOURCE2, NEGATI
2775 :*****
2776 010030 000004                               TST30: SCOPE
2776 010032 004567 004674                               JSR R5,NPREP                               :PREPARE ARGUMENTS FOR INSTRUCTION TEST

```

```

2777 010036 000004          4          :SOURCE1 LENGTH
2778 010040 010202        S1T30    :SOURCE1 ADDRESS
2779 010042 000004          4          :SOURCE2 LENGTH
2780 010044 010206        S2T30    :SOURCE2 ADDRESS
2781 010046 000377          377       :STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
2782 010050 004567 004770  JSR        R5,XPSW
2783 010054 000210          .WORD    210
2784 010056 004767 004676  JSR        PC,GENR          :SET UP GENERAL REGISTERS
2785 010062 000277          SCC
2786 010064 000250          CLN          :SET UP THE COMPLEMENT OF EXPECTED CC'S
2787
2788 010066 076052          CMPN
2789
2790 010070 106767 170602  MFPS      CCODES          :STORE RESULTANT PSW
2791 010074 042767 177400 2170574  BIC      #177400,CCODES  :CLEAR UNUSED BITS
2792 010102 023767 000700 2170566  CMP      @#EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
2793 010110 001401          BEQ      64$             :BR, IF EQUAL
2794 010112 104001          ERROR    1             :*****TEST 30 - ERROR 1*****
2795
2796
2797
2798          010114          64$:          TST      R0
2799 010114 005700          BEQ      R0             :CHECK R0=0
2800 010116 001401          ERROR    2             :*****TEST 30 - ERROR 2*****
2801 010120 104002          :R0 SHOULD BE ZERO
2802
2803 010122 005701          65$:          TST      R1
2804 010124 001401          BEQ      R1             :CHECK R1=0
2805 010126 104003          ERROR    3             :*****TEST 30 - ERROR 3*****
2806
2807 010130 005702          66$:          TST      R2
2808 010132 001401          BEQ      R2             :CHECK R2=0
2809 010134 104004          ERROR    4             :*****TEST 30 - ERROR 4*****
2810
2811 010136 005703          67$:          TST      R3
2812 010140 001401          BEQ      R3             :CHECK R3=0
2813 010142 104005          ERROR    5             :*****TEST 30 - ERROR 5*****
2814
2815 010144 020467 170504          68$:          CMP      R4,DSTLN
2816 010150 001401          BEQ      R4             :CHECK R4 UNCHANGED
2817 010152 104006          ERROR    6             :BR IF OK
2818
2819
2820 010154 020567 170476          69$:          CMP      R5,DSTAD
2821 010160 001401          BEQ      R5             :CHECK R5 UNCHANGED
2822 010162 104007          ERROR    7             :*****TEST 30 - ERROR 7*****
2823
2824
2825 010164 023706 000702          70$:          CMP      @#SAVR6,SP
2826 010170 001403          BEQ      R5             :R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
2827 010172 010637 000704          MOV      SP,@#BADR6
2828 010176 104010          ERROR    10            :VERIFY STACK POINTER IS RESTORED
2829
2830
2831
2832 010200          71$:          :BR IF OK
                :COPY BAD SP VALUE
                :*****TEST 30 - ERROR 10*****
                :STACK POINTER NOT RESTORED BY INSTRUCTION
                :EXPECTED SP IS STORED AT "SAVR6"
                :ERRONEOUS SP VALUE IS AT "BADR6"

```


2833 010200 000404
2834 010202
2835 010202 070
2836 010203 066
2837 010204 064
2838 010205 162
2839 010206
2840 010206 070
2841 010207 066
2842 010210 064
2843 010211 062
2844
2845
2846
2847
2848
2849
2850
2851
2852 010212 000004
2853 010214 004567 004512
2854 010220 000002
2855 010222 010364
2856 010224 000002
2857 010226 010366
2858 010230 000377
2859 010232 004567 004606
2860 010236 000200
2861 010240 004767 004514
2862 010244 000277
2863 010246 000250
2864
2865 010250 076052
2866
2867 010252 106767 170420
2868 010256 042767 177400 170412
2869 010264 023767 000700 170404
2870 010272 001401
2871 010274 104001
2872
2873
2874
2875 010276
2876 010276 005700
2877 010300 001401
2878 010302 104002
2879
2880 010304 005701
2881 010306 001401
2882 010310 104003
2883
2884 010312 005702
2885 010314 001401
2886 010316 104004
2887
2888 010320 005703

BR TST31 ;BR TO NEXT TEST
S1T30: ;SOURCE1 STRING
;MOST SIGNIFICANT DIGIT
.BYTE 70
.BYTE 66
.BYTE 64
.BYTE 162
S2T30: ;SOURCE2 STRING
;MOST SIGNIFICANT DIGIT
.BYTE 70
.BYTE 66
.BYTE 64
.BYTE 62
.EVEN
:*****
:*TEST 31 TEST 'CMPN' WITH +SRC1 & -SRC2, S1L = S2L, /S1/ .GT. /S2/
:*****
TST31: SCOPE
JSR R5,NPREP ;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 ;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 31 - 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 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

```

2889 010322 001401      BEQ      68$
2890 010324 104005      ERROR    5      ;*****TEST 31 - ERROR 5*****
2891                                     ;R3 SHOULD BE ZERO
2892 010326 020467 170322 68$:    CMP      R4,DSTLN ;CHECK R4 UNCHANGED
2893 010332 001401      BEQ      69$      ;BR IF OK
2894 010334 104006      ERROR    6      ;*****TEST 31 - ERROR 6*****
2895                                     ;R4 CHANGED
2896                                     ;R4 SHOULD STILL EQUAL CONTENTS OF 'FILL'
2897 010336 020567 170314 69$:    CMP      R5,DSTAD ;CHECK R5 UNCHANGED
2898 010342 001401      BEQ      70$
2899 010344 104007      ERROR    7      ;*****TEST 31 - ERROR 7*****
2900                                     ;R5 CHANGED
2901                                     ;R5 SHOULD STILL EQUAL CONTENTS OF 'DSTAD'
2902 010346 023706 000702 70$:    CMP      @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2903 010352 001403      BEQ      71$      ;BR IF OK
2904 010354 010637 000704      MOV      SP,@#BADR6 ;COPY BAD SP VALUE
2905 010360 104010      ERROR    10     ;*****TEST 31 - ERROR 10*****
2906                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
2907                                     ;EXPECTED SP IS STORED AT 'SAVR6'
2908                                     ;ERRONEOUS SP VALUE IS AT 'BADR6'
2909 010362                                     71$:
2910 010362 000402      BR        TST32      ;BR TO NEXT TEST
2911 010364      S1T31:
2912 010364          071      .BYTE    71      ;SOURCE1 STRING
2913 010365          066      .BYTE    66      ;MOST SIGNIFICANT DIGIT
2914 010366      S2T31:
2915 010366          066      .BYTE    66      ;SOURCE2 STRING
2916 010367          171      .BYTE    171     ;MOST SIGNIFICANT DIGIT
2917
2918          .EVEN
2919
2920
2921
2922
2923
2924 010370 000004      ;*****
2925 010372 004567 004334 ;*TEST 32      TEST "CMPN" WITH -SRC1 & +SRC2, S1L = S2L, /S1/ .GT. /S2/
2926 010376 000003      ;*****
2927 010400 010542      TST32:  SCOPE
2928 010402 000003      JSR      R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
2929 010404 010545      3      ;SOURCE1 LENGTH
2930 010406 000377      S1T32 ;SOURCE1 ADDRESS
2931 010410 004567 004430 JSR      R5,XPSW ;SOURCE2 LENGTH
2932 010414 000210      3      ;SOURCE2 ADDRESS
2933 010416 004767 004336 JSR      PC,GENR ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY 'CMPN' IN
2934 010422 000267      +SEV!SEZ!SEC
2935 010424 000250      CLN      ;SET UP GENERAL REGISTERS
2936                                     ;SET UP THE COMPLEMENT OF EXPECTED CC'S
2937 010426 076052      CMPN
2938
2939 010430 106767 170242 MFPS     CCODES ;STORE RESULTANT PSW
2940 010434 042767 177400 170234 BIC      #177400,CCODES ;CLEAR UNUSED BITS
2941 010442 023767 000700 170226 CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2942 010450 001401      BEQ      64$      ;BR, IF EQUAL
2943 010452 104001      ERROR    1      ;*****TEST 32 - ERROR 1*****
2944

```



```

2945                                     :EXPECTED PSW IS STORED AT "SAVR6"
2946                                     :ERRONEOUS SP VALUE IS AT "BADR6"
2947 010454                               64$: TST      R0
2948 010454 005700                       BEQ      65$
2949 010456 001401                       ERROR    2
2950 010460 104002                       :*****TEST 32 - ERROR 2*****
2951                                     :R0 SHOULD BE ZERO
2952 010462 005701                       65$: TST      R1
2953 010464 001401                       BEQ      66$
2954 010466 104003                       ERROR    3
2955                                     :*****TEST 32 - ERROR 3*****
2956 010470 005702                       66$: TST      R2
2957 010472 001401                       BEQ      67$
2958 010474 104004                       ERROR    4
2959                                     :*****TEST 32 - ERROR 4*****
2960 010476 005703                       67$: TST      R3
2961 010500 001401                       BEQ      68$
2962 010502 104005                       ERROR    5
2963                                     :*****TEST 32 - ERROR 5*****
2964 010504 020467 170144                68$: CMP      R4,DSTLN
2965 010510 001401                       BEQ      69$
2966 010512 104006                       ERROR    6
2967                                     :*****TEST 32 - ERROR 6*****
2968                                     :R4 CHANGED
2969 010514 020567 170136                69$: CMP      R5,DSTAD
2970 010520 001401                       BEQ      70$
2971 010522 104007                       ERROR    7
2972                                     :*****TEST 32 - ERROR 7*****
2973                                     :R5 CHANGED
2974 010524 023706 000702                70$: CMP      @#SAVR6,SP
2975 010530 001403                       BEQ      71$
2976 010532 010637 000704                MOV      SP,@#BADR6
2977 010536 104010                       ERROR    10
2978                                     :*****TEST 32 - ERROR 10*****
2979                                     :STACK POINTER NOT RESTORED BY INSTRUCTION
2980                                     :EXPECTED SP IS STORED AT "SAVR6"
2981                                     :ERRONEOUS SP VALUE IS AT "BADR6"
2981 010540                               71$: BR      TST33
2982 010540 000403                       S1T32:
2983 010542                               .BYTE   70
2984 010542 070                            .BYTE   63
2985 010543 063                            .BYTE   161
2986 010544 161                            S2T32:
2987 010545                               .BYTE   61
2988 010545 061                            .BYTE   63
2989 010546 063                            .BYTE   70
2990 010547 070
2991                                     .EVEN
2992
2993
2994
2995                                     :*****
2996                                     :*TEST 3* TEST "CMPN" WITH S1L = S2L, POSITIVE SOURCE2, SOURCE2 MAGNITUDE IS GREAT
2997                                     :*****
2998 010550 000004                               TST33: SCOPE
2999 010552 004567 004154                   JSR      R5,NPREP
3000 010556 000004                               4
                                     :PREPARE ARGUMENTS FOR INSTRUCTION TEST
                                     :SOURCE1 LENGTH

```

```

3001 010560 010722 S1T33 ;SOURCE1 ADDRESS
3002 010562 000004 4 ;SOURCE2 LENGTH
3003 010564 010726 S2T33 ;SOURCE2 ADDRESS
3004 010566 000377 377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3005 010570 004567 004250 JSR R5,XPSW
3006 010574 000210 .WORD 210
3007 010576 004767 004156 JSR PC,GENR ;SET UP GENERAL REGISTERS
3008 010602 000277 SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
3009 010604 000250 CLN
3010
3011 010606 076052 CMPN
3012
3013 010610 106767 170062 MFPS CCODES ;STORE RESULTANT PSW
3014 010614 042767 177400 170054 BIC #177400,CCODES ;CLEAR UNUSED BITS
3015 010622 023767 000700 170046 CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
3016 010630 001401 BEQ 64$ ;BR, IF EQUAL
3017 010632 104001 ERROR 1 ;*****TEST 33 - ERROR 1*****
3018 ;PSW ERROR
3019 ;EXPECTED PSW IS STORED AT "SAVR6"
3020 ;ERRONEOUS SP VALUE IS AT "BADR6"
3021 010634 64$: TST R0 ;CHECK R0=0
3022 010634 005700 BEQ 65$
3023 010636 001401 ERROR 2 ;*****TEST 33 - ERROR 2*****
3024 010640 104002 ;R0 SHOULD BE ZERO
3025 ;CHECK R1=0
3026 010642 005701 65$: TST R1
3027 010644 001401 BEQ 66$
3028 010646 104003 ERROR 3 ;*****TEST 33 - ERROR 3*****
3029 ;R1 SHOULD BE ZERO
3030 010650 005702 66$: TST R2
3031 010652 001401 BEQ 67$
3032 010654 104004 ERROR 4 ;*****TEST 33 - ERROR 4*****
3033 ;R2 SHOULD BE ZERO
3034 010656 005703 67$: TST R3
3035 010660 001401 BEQ 68$
3036 010662 104005 ERROR 5 ;*****TEST 33 - ERROR 5*****
3037 ;R3 SHOULD BE ZERO
3038 010664 020467 167764 68$: CMP R4,DSTLN ;CHECK R4 UNCHANGED
3039 010670 001401 BEQ 69$ ;BR IF OK
3040 010672 104006 ERROR 6 ;*****TEST 33 - ERROR 6*****
3041 ;R4 CHANGED
3042 ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
3043 010674 020567 167756 69$: CMP R5,DSTAD ;CHECK R5 UNCHANGED
3044 010700 001401 BEQ 70$
3045 010702 104007 ERROR 7 ;*****TEST 33 - ERROR 7*****
3046 ;R5 CHANGED
3047 ;R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
3048 010704 023706 000702 70$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
3049 010710 001403 BEQ 71$ ;BR IF OK
3050 010712 010637 000704 MOV SP,@#BADR6 ;COPY BAD SP VALUE
3051 010716 104010 ERROR 10 ;*****TEST 33 - ERROR 10*****
3052 ;STACK POINTER NOT RESTORED BY INSTRUCTION
3053 ;EXPECTED SP IS STORED AT "SAVR6"
3054 ;ERRONEOUS SP VALUE IS AT "BADR6"
3055 010720 71$: BR TST34 ;BR TO NEXT TEST
3056 010720 000404

```


3057 010722
3058 010722 067
3059 010723 066
3060 010724 065
3061 010725 064
3062 010726
3063 010726 067
3064 010727 066
3065 010730 065
3066 010731 065
3067
3068
3069
3070
3071
3072
3073
3074 010732 000004
3075 010734 004567 003772
3076 010740 000002
3077 010742 011104
3078 010744 000004
3079 010746 011106
3080 010750 000377
3081 010752 004567 004066
3082 010756 000204
3083 010760 004767 003774
3084 010764 000277
3085 010766 000244
3086
3087 010770 076052
3088
3089 010772 106767 167700
3090 010776 042767 177400 167672
3091 011004 023767 000700 167664
3092 011012 001401
3093 011014 104001
3094
3095
3096
3097 011016
3098 011016 005700
3099 011020 001401
3100 011022 104002
3101
3102 011024 005701
3103 011026 001401
3104 011030 104003
3105
3106 011032 005702
3107 011034 001401
3108 011036 104004
3109
3110 011040 005703
3111 011042 001401
3112 011044 104005

S1T33: :SOURCE1 STRING
:MOST SIGNIFICANT DIGIT
.BYTE 67
.BYTE 66
.BYTE 65
.BYTE 64
S2T33: :SOURCE2 STRING
:MOST SIGNIFICANT DIGIT
.BYTE 67
.BYTE 66
.BYTE 65
.BYTE 65
.EVEN

:TEST 34 TEST "CMPN" WITH S2L .GT. S1L, SUCCESSFUL COMPARE

TST34: SCOPE JSR R5,NPREP :PREPARE ARGUMENTS FOR INSTRUCTION TEST
JSR 2 :SOURCE1 LENGTH
S1T34 :SOURCE1 ADDRESS
4 :SOURCE2 LENGTH
S2T34 :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 :SET UP THE COMPLEMENT OF EXPECTED CC'S
CLZ
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 34 - 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 34 - ERROR 2*****
:R0 SHOULD BE ZERO
:CHECK R1=0
65\$: TST R1
BEQ 66\$:CHECK R1=0
ERROR 3 :*****TEST 34 - ERROR 3*****
:R1 SHOULD BE ZERO
:CHECK R2=0
66\$: TST R2
BEQ 67\$:CHECK R2=0
ERROR 4 :*****TEST 34 - ERROR 4*****
:R2 SHOULD BE ZERO
:CHECK R3=0
67\$: TST R3
BEQ 68\$:CHECK R3=0
ERROR 5 :*****TEST 34 - ERROR 5*****


```

3225 011302 000004          4          :SOURCE2 LENGTH
3226 011304 011445          S2T36     :SOURCE2 ADDRESS
3227 011306 000377          377       :STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3228 011310 004567 003530  JSR        R5,XPSW
3229 011314 000210          .WORD     210
3230 011316 004767 003436  JSR        PC,GENR          :SET UP GENERAL REGISTERS
3231 011322 000277          SCC
3232 011324 000250          CLN          :SET UP THE COMPLEMENT OF EXPECTED CC'S
3233
3234 011326 076052          CMPN
3235
3236 011330 106767 167342  MFPS      CCODES          :STORE RESULTANT PSW
3237 011334 042767 177400 167334  BIC      #177400,CCODES    :CLEAR UNUSED BITS
3238 011342 023767 000700 167326  CMP      @#EXPPSW,CCODES  :CHECK PSW AGAINST EXPECTED VALUE
3239 011350 001401          BEQ      64$
3240 011352 104001          ERROR    1          :BR, IF EQUAL
3241
3242
3243
3244
3244 011354          64$:
3245 011354 005700          TST      R0
3246 011356 001401          BEQ      65$
3247 011360 104002          ERROR    2          :CHECK R0=0
3248
3249 011362 005701          65$:
3250 011364 001401          TST      R1
3251 011366 104003          BEQ      66$
3252
3253 011370 005702          66$:
3254 011372 001401          TST      R2
3255 011374 104004          BEQ      67$
3256
3257 011376 005703          67$:
3258 011400 001401          TST      R3
3259 011402 104005          BEQ      68$
3260
3261 011404 020467 167244          68$:
3262 011410 001401          CMP      R4,DSTLN
3263 011412 104006          BEQ      69$
3264
3265 011414 020567 167236          69$:
3266 011420 001401          CMP      R5,DSTAD
3267 011422 104007          BEQ      70$
3268
3269 011424 023706 000702          70$:
3270 011430 001403          CMP      @#SAVR6,SP
3271 011432 010637 000704          BEQ      71$
3272 011436 104010          MOV      SP,@#BADR6
3273
3274
3275
3276
3277
3278 011440          71$:
3279 011440 000403          BR
3280 011442          S1T36:

```

```

:SOURCE2 LENGTH
:SOURCE2 ADDRESS
:STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
:SET UP GENERAL REGISTERS
:SET UP THE COMPLEMENT OF EXPECTED CC'S
:STORE RESULTANT PSW
:CLEAR UNUSED BITS
:CHECK PSW AGAINST EXPECTED VALUE
:BR, IF EQUAL
:*****TEST 36 - ERROR 1*****
:PSW ERROR
:EXPECTED PSW IS STORED AT "SAVR6"
:ERRONEOUS SP VALUE IS AT "BADR6"
:CHECK R0=0
:*****TEST 36 - ERROR 2*****
:R0 SHOULD BE ZERO
:CHECK R1=0
:*****TEST 36 - ERROR 3*****
:R1 SHOULD BE ZERO
:CHECK R2=0
:*****TEST 36 - ERROR 4*****
:R2 SHOULD BE ZERO
:CHECK R3=0
:*****TEST 36 - ERROR 5*****
:R3 SHOULD BE ZERO
:CHECK R4 UNCHANGED
:BR IF OK
:*****TEST 36 - ERROR 6*****
:R4 CHANGED
:R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
:CHECK R5 UNCHANGED
:*****TEST 36 - ERROR 7*****
:R5 CHANGED
:R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
:VERIFY STACK POINTER IS RESTORED
:BR IF OK
:COPY BAD SP VALUE
:*****TEST 36 - 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

```



```

3281 011442 070 .BYTE 70 ;MOST SIGNIFICANT DIGIT
3282 011443 061 .BYTE 61
3283 011444 061 .BYTE 61
3284 011445 S2T36: ;SOURCE2 STRING
3285 011445 060 .BYTE 60 ;MOST SIGNIFICANT DIGIT
3286 011446 070 .BYTE 70
3287 011447 061 .BYTE 61
3288
3289 .EVEN
3290
3291
3292
3293
3294
3295 011450 000004
3296 011452 004567 003254 JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3297 011456 000004 4 ;SOURCE1 LENGTH
3298 011460 011622 S1T37 ;SOURCE1 ADDRESS
3299 011462 000002 2 ;SOURCE2 LENGTH
3300 011464 011626 S2T37 ;SOURCE2 ADDRESS
3301 011466 000377 377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3302 011470 004567 003350 JSR R5,XPSW
3303 011474 000204 .WORD 204
3304 011476 004767 003256 JSR PC,GENR ;SET UP GENERAL REGISTERS
3305 011502 000277 SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
3306 011504 000244 CLZ
3307
3308 011506 076052 CMPN
3309
3310 011510 106767 167162 MFPS CCODES ;STORE RESULTANT PSW
3311 011514 042767 177400 167154 BIC #177400,CCODES ;CLEAR UNUSED BITS
3312 011522 023767 000700 167146 CMP @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
3313 011530 001401 BEQ 64$ ;BR, IF EQUAL
3314 011532 104001 ERROR 1 ;*****TEST 37 - ERROR 1*****
3315 ;PSW ERROR
3316 ;EXPECTED PSW IS STORED AT "SAVR6"
3317 ;ERRONEOUS SP VALUE IS AT "BADR6"
3318 011534 64$:
3319 011534 005700 TST R0 ;CHECK R0=0
3320 011536 001401 BEQ 65$
3321 011540 104002 ERROR 2 ;*****TEST 37 - ERROR 2*****
3322 ;R0 SHOULD BE ZERO
3323 011542 005701 65$: TST R1 ;CHECK R1=0
3324 011544 001401 BEQ 66$
3325 011546 104003 ERROR 3 ;*****TEST 37 - ERROR 3*****
3326 ;R1 SHOULD BE ZERO
3327 011550 005702 66$: TST R2 ;CHECK R2=0
3328 011552 001401 BEQ 67$
3329 011554 104004 ERROR 4 ;*****TEST 37 - ERROR 4*****
3330 ;R2 SHOULD BE ZERO
3331 011556 005703 67$: TST R3 ;CHECK R3=0
3332 011560 001401 BEQ 68$
3333 011562 104005 ERROR 5 ;*****TEST 37 - ERROR 5*****
3334 ;R3 SHOULD BE ZERO
3335 011564 020467 167064 68$: CMP R4,DSTLN ;CHECK R4 UNCHANGED
3336 011570 001401 BEQ 69$ ;BR IF OK

```



```

3393 011714 005700      TST      R0      :CHECK R0=0
3394 011716 001401      BEQ      65$
3395 011720 104002      ERROR    2      :*****TEST 40 - ERROR 2*****
3396                                :R0 SHOULD BE ZERO
3397 011722 005701      65$: TST      R1      :CHECK R1=0
3398 011724 001401      BEQ      66$
3399 011726 104003      ERROR    3      :*****TEST 40 - ERROR 3*****
3400                                :R1 SHOULD BE ZERO
3401 011730 005702      66$: TST      R2      :CHECK R2=0
3402 011732 001401      BEQ      67$
3403 011734 104004      ERROR    4      :*****TEST 40 - ERROR 4*****
3404                                :R2 SHOULD BE ZERO
3405 011736 005703      67$: TST      R3      :CHECK R3=0
3406 011740 001401      BEQ      68$
3407 011742 104005      ERROR    5      :*****TEST 40 - ERROR 5*****
3408                                :R3 SHOULD BE ZERO
3409 011744 020467 166704 68$: CMP      R4,DSTLN :CHECK R4 UNCHANGED
3410 011750 001401      BEQ      69$
3411 011752 104006      ERROR    6      :BR IF OK
3412                                :*****TEST 40 - ERROR 6*****
3413                                :R4 CHANGED
3414 011754 020567 166676 69$: CMP      R5,DSTAD :CHECK R5 UNCHANGED
3415 011760 001401      BEQ      70$
3416 011762 104007      ERROR    7      :*****TEST 40 - ERROR 7*****
3417                                :R5 CHANGED
3418                                :R5 SHOULD STILL EQUAL CONTENTS OF 'DSTAD'
3419 011764 023706 000702 70$: CMP      @#SAVR6,SP :VERIFY STACK POINTER IS RESTORED
3420 011770 001403      BEQ      71$
3421 011772 010637 000704 MOV      SP,@#BADR6 :BR IF OK
3422 011776 104010      ERROR    10     :COPY BAD SP VALUE
3423                                :*****TEST 40 - ERROR 10*****
3424                                :STACK POINTER NOT RESTORED BY INSTRUCTION
3425                                :EXPECTED SP IS STORED AT "SAVR6"
3426                                :ERRONEOUS SP VALUE IS AT "BADR6"
3426 012000      71$:
3427 012000 000403      BR      TST41    :BR TO NEXT TEST
3428 012002      S1T40:
3429 012002      .BYTE    61    :SOURCE1 STRING
3430 012003      .BYTE    60    :MOST SIGNIFICANT DIGIT
3431 012004      .BYTE    63
3432 012005      .BYTE    167
3433 012006      S2T40:
3434 012006      .BYTE    63    :SOURCE2 STRING
3435 012007      .BYTE    167    :MOST SIGNIFICANT DIGIT
3436
3437      .EVEN
3438
3439
3440
3441
3442

```

```

:*****
:*TEST 41      TEST "CMPN" WITH S1L .GT. S2L, POSITIVE SOURCE1, NON-ZERO IN EXCESS
:*****

```

```

TST41: SCOPE
      JSR      R5,NPREP :PREPARE ARGUMENTS FOR INSTRUCTION TEST
3443 012010 000004      4      :SOURCE1 LENGTH
3444 012012 004567 002714 S1T41 :SOURCE1 ADDRESS
3445 012016 000004      2      :SOURCE2 LENGTH
3446 012020 012160      S2T41 :SOURCE2 ADDRESS
3447 012022 000002
3448 012024 012164

```

```

3449 012026 000377          377          :STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3450 012030 004567 003010   JSR      R5,XPSW
3451 012034 000200          .WORD   200
3452 012036 004767 002716   JSR      PC,GENR          :SET UP GENERAL REGISTERS
3453 012042 000277          SCC              :SET UP THE COMPLEMENT OF EXPECTED CC'S
3454
3455 012044 076052          CMPN
3456
3457 012046 106767 166624   MFPS      CCODES          :STORE RESULTANT PSW
3458 012052 042767 177400 166616   BIC      #177400,CCODES   :CLEAR UNUSED BITS
3459 012060 023767 000700 166610   CMP      @#EXPPSW,CCODES  :CHECK PSW AGAINST EXPECTED VALUE
3460 012066 001401          BEQ      64$             :BR, IF EQUAL
3461 012070 104001          ERROR    1              :*****TEST 41 - ERROR 1*****
3462
3463
3464
3465 012072          64$:          TST      R0              :CHECK R0=0
3466 012072 005700          BEQ      65$
3467 012074 001401          ERROR    2              :*****TEST 41 - ERROR 2*****
3468 012076 104002          :R0 SHOULD BE ZERO
3469
3470 012100 005701          65$:          TST      R1              :CHECK R1=0
3471 012102 001401          BEQ      66$
3472 012104 104003          ERROR    3              :*****TEST 41 - ERROR 3*****
3473
3474 012106 005702          66$:          TST      R2              :CHECK R2=0
3475 012110 001401          BEQ      67$
3476 012112 104004          ERROR    4              :*****TEST 41 - ERROR 4*****
3477
3478 012114 005703          67$:          TST      R3              :CHECK R3=0
3479 012116 001401          BEQ      68$
3480 012120 104005          ERROR    5              :*****TEST 41 - ERROR 5*****
3481
3482 012122 020467 166526          68$:          CMP      R4,DSTLN        :R3 SHOULD BE ZERO
3483 012126 001401          BEQ      69$            :CHECK R4 UNCHANGED
3484 012130 104006          ERROR    6              :BR IF OK
3485
3486
3487 012132 020567 166520          69$:          CMP      R5,DSTAD        :*****TEST 41 - ERROR 6*****
3488 012136 001401          BEQ      70$            :R4 CHANGED
3489 012140 104007          ERROR    7              :R4 SHOULD STILL EQUAL CONTENTS OF "FILL"
3490
3491
3492 012142 023706 000702          70$:          CMP      @#SAVR6,SP      :CHECK R5 UNCHANGED
3493 012146 001403          BEQ      71$            :*****TEST 41 - ERROR 7*****
3494 012150 010637 000704          MOV      SP,@#BADR6      :R5 CHANGED
3495 012154 104010          ERROR    10             :R5 SHOULD STILL EQUAL CONTENTS OF "DSTAD"
3496
3497
3498
3499 012156          71$:          BR      TST42           :VERIFY STACK POINTER IS RESTORED
3500 012156 000403          S1T41:        :BR IF OK
3501 012160          :COPY BAD SP VALUE
3502 012160 061          :*****TEST 41 - ERROR 10*****
3503 012161 060          :STACK POINTER NOT RESTORED BY INSTRUCTION
3504 012162 063          :EXPECTED SP IS STORED AT "SAVR6"
          :ERRONEOUS SP VALUE IS AT "BADR6"
          :BR TO NEXT TEST
          :SOURCE1 STRING
          :MOST SIGNIFICANT DIGIT

```


3505 012163 067
3506 012164
3507 012164 063
3508 012165 067
3509
3510
3511
3512
3513
3514
3515
3516
3517 012166 000004
3518 012170 004567 002536
3519 012174 000003
3520 012176 012336
3521 012200 000003
3522 012202 012341
3523 012204 000377
3524 012206 004567 002632
3525 012212 000200
3526 012214 004767 002540
3527 012220 000277
3528
3529 012222 076052
3530
3531 012224 106767 166446
3532 012230 042767 177400 166440
3533 012236 023767 000700 166432
3534 012244 001401
3535 012246 104001
3536
3537
3538
3539 012250
3540 012250 005700
3541 012252 001401
3542 012254 104002
3543
3544 012256 005701
3545 012260 001401
3546 012262 104003
3547
3548 012264 005702
3549 012266 001401
3550 012270 104004
3551
3552 012272 005703
3553 012274 001401
3554 012276 104005
3555
3556 012300 020467 166350
3557 012304 001401
3558 012306 104006
3559
3560

S2T41: .BYTE 67 ;SOURCE2 STRING
.BYTE 63 ;MOST SIGNIFICANT DIGIT
.BYTE 67
.EVEN
:*****
:*TEST 42 TEST "CMPN" WITH POSITIVE OPERANDS, /S1/ .GT. /S2/
:*****
TST42: SCOPE R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
JSR 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 "SAVR6"
;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 ;*****TEST 42 - ERROR 6*****
BEQ 69\$;R4 CHANGED
ERROR 6 ;R4 SHOULD STILL EQUAL CONTENTS OF "FILL"

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

```
3561 012310 020567 166342 69$: CMP R5,DSTAD ;CHECK R5 UNCHANGED
3562 012314 001401 BEQ 70$
3563 012316 104007 ERROR 7 ;*****TEST 42 - ERROR 7*****
3564 ;R5 CHANGED
3565 ;R5 SHOULD STILL EQUAL CONTENTS OF 'DSTAD'
3566 012320 023706 000702 70$: CMP @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
3567 012324 001403 BEQ 71$ ;BR IF OK
3568 012326 010637 000704 MOV SP,@BADR6 ;COPY BAD SP VALUE
3569 012332 104010 ERROR 10 ;*****TEST 42 - ERROR 10*****
3570 ;STACK POINTER NOT RESTORED BY INSTRUCTION
3571 ;EXPECTED SP IS STORED AT 'SAVR6'
3572 ;ERRONEOUS SP VALUE IS AT 'BADR6'
3573 012334 71$: BR TST43 ;BR TO NEXT TEST
3574 012334 000403 S1T42: .BYTE 67 ;SOURCE1 STRING
3575 012336 .BYTE 65 ;MOST SIGNIFICANT DIGIT
3576 012336 067 .BYTE 64
3577 012337 065
3578 012340 064
3579 012341 S2T42: .BYTE 67 ;SOURCE2 STRING
3580 012341 067 .BYTE 65 ;MOST SIGNIFICANT DIGIT
3581 012342 065 .BYTE 63
3582 012343 063
3583
3584 .EVEN
3585
3586
3587
```

*TEST 43 TEST "CMPN" WITH NEGATIVE OPERANDS, /S1/ .GT. /S2/

```
3588
3589
3590 012344 000004 TST43: SCOPE
3591 012346 004567 002360 JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
3592 012352 000003 3 ;SOURCE1 LENGTH
3593 012354 012514 S1T43 ;SOURCE1 ADDRESS
3594 012356 000003 3 ;SOURCE2 LENGTH
3595 012360 012517 S2T43 ;SOURCE2 ADDRESS
3596 012362 000377 377 ;STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3597 012364 004567 002454 JSR R5,XPSW
3598 012370 000210 .WORD 210
3599 012372 004767 002362 JSR PC,GENR ;SET UP GENERAL REGISTERS
3600 012376 000277 SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
3601
3602 012400 076052 CMPN
3603
3604 012402 106767 166270 MFPS CCODES ;STORE RESULTANT PSW
3605 012406 042767 177400 166262 BIC #177400,CCODES ;CLEAR UNUSED BITS
3606 012414 023767 000700 166254 CMP @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
3607 012422 001401 BEQ 64$ ;BR, IF EQUAL
3608 012424 104001 ERROR 1 ;*****TEST 43 - ERROR 1*****
3609 ;PSW ERROR
3610 ;EXPECTED PSW IS STORED AT 'SAVR6'
3611 ;ERRONEOUS SP VALUE IS AT 'BADR6'
3612 012426 64$: TST R0 ;CHECK R0=0
3613 012426 005700 BEQ 65$
3614 012430 001401 ERROR 2 ;*****TEST 43 - ERROR 2*****
3615 012432 104002 ;R0 SHOULD BE ZERO
3616
```


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

```
3617 012434 005701 65$: TST R1 :CHECK R1=0
3618 012436 001401 BEQ 66$
3619 012440 104003 ERROR 3 :*****TEST 43 - ERROR 3*****
3620 :R1 SHOULD BE ZERO
3621 012442 005702 66$: TST R2 :CHECK R2=0
3622 012444 001401 BEQ 67$
3623 012446 104004 ERROR 4 :*****TEST 43 - ERROR 4*****
3624 :R2 SHOULD BE ZERO
3625 012450 005703 67$: TST R3 :CHECK R3=0
3626 012452 001401 BEQ 68$
3627 012454 104005 ERROR 5 :*****TEST 43 - ERROR 5*****
3628 :R3 SHOULD BE ZERO
3629 012456 020467 166172 68$: CMP R4,DSTLN :CHECK R4 UNCHANGED
3630 012462 001401 BEQ 69$ :BR IF OK
3631 012464 104006 ERROR 6 :*****TEST 43 - ERROR 6*****
3632 :R4 CHANGED
3633 :R4 SHOULD STILL EQUAL CONTENTS OF 'FILL'
3634 012466 020567 166164 69$: CMP R5,DSTAD :CHECK R5 UNCHANGED
3635 012472 001401 BEQ 70$
3636 012474 104007 ERROR 7 :*****TEST 43 - ERROR 7*****
3637 :R5 CHANGED
3638 :R5 SHOULD STILL EQUAL CONTENTS OF 'DSTAD'
3639 012476 023706 000702 70$: CMP @#SAVR6,SP :VERIFY STACK POINTER IS RESTORED
3640 012502 001403 BEQ 71$ :BR IF OK
3641 012504 010637 000704 MOV SP,@#BADR6 :COPY BAD SP VALUE
3642 012510 104010 ERROR 10 :*****TEST 43 - ERROR 10*****
3643 :STACK POINTER NOT RESTORED BY INSTRUCTION
3644 :EXPECTED SP IS STORED AT 'SAVR6'
3645 :ERRONEOUS SP VALUE IS AT 'BADR6'
3646 012512 71$: BR TST44 :BR TO NEXT TEST
3647 012512 000403 S1T43: :SOURCE1 STRING
3648 012514 :.BYTE 67 :MOST SIGNIFICANT DIGIT
3649 012514 067 :.BYTE 65
3650 012515 065 :.BYTE 164
3651 012516 164 S2T43: :SOURCE2 STRING
3652 012517 :.BYTE 67 :MOST SIGNIFICANT DIGIT
3653 012517 067 :.BYTE 65
3654 012520 065 :.BYTE 163
3655 012521 163 :EVEN
3656
3657
3658
3659
3660
```

*TEST 44 TEST INTERRUPTABILITY OF 'CMPN'

```
3663 012522 000004 TST44: SCOPE
3664 012524 105777 166010 TSTB @SWR :TEST BIT 7 OF SWR
3665 012530 100535 BMI TST45 :SKIP TO NEXT TEST IF SET
3666 012532 026767 166012 166124 CMP $TPS,TCSR :IS SLU USED FOR INTERRUPTS THE CONSOLE?
3667 012540 001007 BNE T44CONT :BR, IF NOT & PERFORM INTERRUPTABILITY TEST
3668 012542 032767 000001 166036 BIT #BIT0,$ENV :IF YES, IS PROGRAM UNDER APT?
3669 012550 001403 BEQ T44CONT :BR, IF NOT
3670 012552 005767 166016 TST $PASS :IF YES,CHECK IF NOT ON FIRST PASS
3671 012556 001122 BNE TST45 :IF NOT, BR & SKIP TEST
3672 012560 T44CONT:
```

TEST INTERRUPTABILITY OF "CMPN"

3673	012560	004567	002146		JSR	R5,NPREP	:PREPARE ARGUMENTS FOR INSTRUCTION TEST
3674	012564	000004			4		:SOURCE1 LENGTH
3675	012566	013002			S1T44	:SOURCE1	ADDRESS
3676	012570	000004			4		:SOURCE2 LENGTH
3677	012572	013006			S2T44		:SOURCE2 ADDRESS
3678	012574	000377			377		:STORE NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CMPN" IN
3679	012576	012767	012656	166070	MOV	#CMPNPC,PCI	:STORE PC OF TEST INSTRUCTION
3680	012604	012777	015114	166056	MOV	#INTR,@TVECT	:POINT TTY VECTOR TO INTERRUPT ROUTINE
3681	012612	005077	166054		CLR	@TPSW	:ALLOW INTERRUPTS AFTER TTY INTERRUPT
3682	012616	004767	002246		JSR	PC,TDONE	:WAIT FOR SLU READY
3683	012622	013777	000554	166036	MOV	@\$NULL,@TBUF	:SEND NULL CHARACTER
3684	012630	004567	002210		JSR	R5,XPSW	:STORE EXPECTED PSW
3685	012634	000000			.WORD	0	
3686	012636	106427	000000		MTPS	#0	:ALLOW INTERRUPTS
3687	012642	052777	000100	166014	BIS	#100,@TCSR	:ENABLE TTY INTERRUPTS
3688	012650	004767	002104		RECMPN: JSR	PC,GENR	:SET UP GENERAL REGISTERS
3689	012654	000277			SCC		:SET UP THE COMPLEMENT OF EXPECTED CC'S
3690							
3691	012656	076052			CMPNPC: CMPN		
3692							
3693	012660	106767	166012		MFPS	CCODES	:STORE RESULTANT PSW
3694	012664	032777	000100	165772	BIT	#100,@TCSR	:IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
3695	012672	001366			BNE	RECMPN	:BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
3696	012674	042767	177400	165774	BIC	#177400,CCODES	:CLEAR UNUSED BITS
3697	012702	023767	000700	165766	CMP	@WEXPPSW,CCODES	:CHECK PSW AGAINST EXPECTED VALUE
3698	012710	001401			BEQ	64\$:BR, IF EQUAL
3699	012712	104001			ERROR	1	:*****TEST 44 - ERROR 1*****
3700							:PSW ERROR
3701							:EXPECTED PSW IS STORED AT "SAVR6"
3702							:ERRONEOUS SP VALUE IS AT "BADR6"
3703	012714				64\$:		
3704	012714	005700			TST	R0	:CHECK R0=0
3705	012716	001401			BEQ	65\$	
3706	012720	104002			ERROR	2	:*****TEST 44 - ERROR 2*****
3707							:RC SHOULD BE ZERO
3708	012722	005701			65\$:	TST	R1
3709	012724	001401			BEQ	66\$:CHECK R1=0
3710	012726	104003			ERROR	3	:*****TEST 44 - ERROR 3*****
3711							:R1 SHOULD BE ZERO
3712	012730	005702			66\$:	TST	R2
3713	012732	001401			BEQ	67\$:CHECK R2=0
3714	012734	104004			ERROR	4	:*****TEST 44 - ERROR 4*****
3715							:R2 SHOULD BE ZERO
3716	012736	005703			67\$:	TST	R3
3717	012740	001401			BEQ	68\$:CHECK R3=0
3718	012742	104005			ERROR	5	:*****TEST 44 - ERROR 5*****
3719							:R3 SHOULD BE ZERO
3720	012744	020467	165704		68\$:	CMP	R4,DSTLN
3721	012750	001401			BEQ	69\$:CHECK R4 UNCHANGED
3722	012752	104006			ERROR	6	:BR IF OK
3723							:*****TEST 44 - ERROR 6*****
3724							:R4 CHANGED
3725	012754	020567	165676		69\$:	CMP	R5,DSTAD
3726	012760	001401			BEQ	70\$:CHECK R5 UNCHANGED
3727	012762	104007			ERROR	7	:*****TEST 44 - ERROR 7*****
3728							:R5 CHANGED


```
3729                                     :R5 SHOULD STILL EQUAL CONTENTS OF 'DSTAD'  
3730 012764 023706 000702      70$:  CMP      @#SAVR6,SP      :VERIFY STACK POINTER IS RESTORED  
3731 012770 001403                                     :BR IF OK  
3732 012772 010637 000704      MOV      SP,@#BADR6      :COPY BAD SP VALUE  
3733 012776 104010      ERROR      10      :*****TEST 44 - ERROR 10*****  
3734                                     :STACK POINTER NOT RESTORED BY INSTRUCTION  
3735                                     :EXPECTED SP IS STORED AT 'SAVR6'  
3736                                     :ERRONEOUS SP VALUE IS AT 'BADR6'  
3737 013000                                     71$:  
3738 013000 000404      BR      ENDT44      :BR TO END OF THIS TEST  
3739 013002      S1T44:                                     :SOURCE1 STRING  
3740 013002      .BYTE      71      :MOST SIGNIFICANT DIGIT  
3741 013003      .BYTE      70  
3742 013004      .BYTE      67  
3743 013005      .BYTE      66  
3744 013006      S2T44:                                     :SOURCE2 STRING  
3745 013006      .BYTE      71      :MOST SIGNIFICANT DIGIT  
3746 013007      .BYTE      70  
3747 013010      .BYTE      67  
3748 013011      .BYTE      65  
3749  
3750                                     .EVEN  
3751 013012 016777 165654 165650 ENDT44: MOV      TPSW,@TVECT  
3752 013020 106427 000200      MTPS      #200  
3753  
3754  
3755  
3756 :*****  
3757 :*TEST 45      TEST 'CVTNL' WITH SRC = +2,147,483,647  
3758 :*****  
3759 013024 000004      TST45: SCOPE  
3760 013026 004567 001700      JSR      R5,NPREP      :PREPARE ARGUMENTS FOR INSTRUCTION TEST  
3761 013032 000012      12      :SOURCE LENGTH  
3762 013034 013164      ST45      :SOURCE ADDRESS  
3763 013036 177777      177777      :STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS  
3764 013040 177777      177777  
3765 013042 000377      377      :STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY 'CVTZL'  
3766 013044 004567 001774      JSR      R5,XPSW  
3767 013050 000200      .WORD      200  
3768 013052 004767 001702      JSR      PC,GENR      :SET UP GENERAL REGISTERS  
3769 013056 000277      SCC      :SET UP THE COMPLEMENT OF EXPECTED CC'S  
3770  
3771 013060 076053      CVTNL  
3772  
3773 013062 106767 165610      MFPS      CCODES      :STORE RESULTANT PSW  
3774 013066 042767 177400 165602      BIC      #177400,CCODES :CLEAR UNUSED BITS  
3775 013074 023767 000700 165574      CMP      @#EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE  
3776 013102 001401      BEQ      64$  
3777 013104 104001      ERROR      1      :BR, IF EQUAL  
3778                                     :*****TEST 45 - ERROR 1*****  
3779                                     :PSW ERROR  
3780                                     :EXPECTED PSW IS STORED AT 'SAVR6'  
3781                                     :ERRONEOUS SP VALUE IS AT 'BADR6'  
3781 013106      64$:  
3782 013106 005700      TST      R0      :CHECK R0=0  
3783 013110 001401      BEQ      65$  
3784 013112 104002      ERROR      2      :*****TEST 45 - ERROR 2*****
```

```
3785                                     :R0 SHOULD BE ZERO
3786 013114 005701                       65$: TST R1                               :CHECK R1=0
3787 013116 001401                       BEQ 66$
3788 013120 104003                       ERROR 3                               :*****TEST 45 - ERROR 3*****
3789                                     :R1 SHOULD BE ZERO
3790 013122 020227 077777               66$: CMP R2,#077777                       :CHECK UPPER WORD OF ANSWER
3791 013126 001401                       BEQ 67$
3792 013130 104004                       ERROR 4                               :BR IF OK
3793                                     :*****TEST 45 - ERROR 4*****
3794                                     :UPPER WORD OF ANSWER IS IN ERROR
3795                                     :EXPECTED VALUE IS 077777
3796 013132 020327 177777               67$: CMP R3,#177777                       :ERRONEOUS ANSWER VALUE IS IN R2
3797 013136 001401                       BEQ 68$
3798 013140 104005                       ERROR 5                               :CHECK LOWER WORD OF ANSWER
3799                                     :BR IF OK
3800                                     :*****TEST 45 - ERROR 5*****
3801                                     :LOWER WORD OF ANSWER IS IN ERROR
3802 013142 020467 165506               68$: CMP R4,DSTLN                          :EXPECTED VALUE IS 177777
3803 013146 001401                       BEQ 69$
3804 013150 104006                       ERROR 6                               :ERRONEOUS ANSWER IS IN R3
3805                                     :CHECK R4 UNCHANGED
3806                                     :*****TEST 45 - ERROR 6*****
3807 013152 020567 165500               69$: CMP R5,DSTAD                          :R4 CHANGED
3808 013156 001401                       BEQ 70$
3809 013160 104007                       ERROR 7                               :R4 SHOULD STILL EQUAL TH CONTENTS OF 'FILL'
3810                                     :CHECK R5 UNCHANGED
3811                                     :BR IF OK
3812 013162                               70$: BR TST46                               :*****TEST 45 - ERROR 7*****
3813 013162 000405                       ST45:                                     :R5 CHANGED
3814 013164                               :R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
3815 013164 062                                     :BR TO NEXT TEST
3816 013165 061                                     :SOURCE STRING
3817 013166 064                                     :MOST SIGNIFICANT DIGIT
3818 013167 067
3819 013170 064
3820 013171 070
3821 013172 063
3822 013173 066
3823 013174 064
3824 013175 067
3825
3826
3827
3828
3829
3830
3831
3832 013176 000004 001526               :*****
3833 013200 004567                               :*TEST 46 TEST "CVTNL" WITH SRC= +2,687,483,648, OVERFLOW WITH CORRECT SIGN
3834 013204 000012                               :*****
3835 013206 013340
3836 013210 177777
3837 013212 177777
3838 013214 000377
3839 013216 004567 001622               TST46: SCOPE
3840 013222 000202                       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
                                           :STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
                                           377
                                           JSR R5,XPSW
                                           .WORD 202
```



```

3841 013224 004767 001530 JSR PC,GENR :SET UP GENERAL REGISTERS
3842 013230 000277 SCC :SET UP THE COMPLEMENT OF EXPECTED CC'S
3843 013232 000242 CLV
3844
3845 013234 076053 CVTNL
3846
3847 013236 106767 165434 MFPS CCODES :STORE RESULTANT PSW
3848 013242 042767 177400 165426 BIC #177400,CCODES :CLEAR UNUSED BITS
3849 013250 023767 000700 165420 CMP @#EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
3850 013256 001401 BEQ 64$ :BR, IF EQUAL
3851 013260 104001 ERROR 1 :*****TEST 46 - ERROR 1*****
3852 :PSW ERROR
3853 :EXPECTED PSW IS STORED AT "SAVR6"
3854 :ERRONEOUS SP VALUE IS AT "BADR6"
3855 013262 64$: TST R0 :CHECK R0=0
3856 013262 005700 BEQ 65$
3857 013264 001401 ERROR 2 :*****TEST 46 - ERROR 2*****
3858 013266 104002 :R0 SHOULD BE ZERO
3859 :CHECK R1=0
3860 013270 005701 65$: TST R1
3861 013272 001401 BEQ 66$
3862 013274 104003 ERROR 3 :*****TEST 46 - ERROR 3*****
3863 :R1 SHOULD BE ZERO
3864 013276 020227 014631 66$: CMP R2,#014631 :CHECK UPPER WORD OF ANSWER
3865 013302 001401 BEQ 67$ :BR IF OK
3866 013304 104004 ERROR 4 :*****TEST 46 - ERROR 4*****
3867 :UPPER WORD OF ANSWER IS IN ERROR
3868 :EXPECTED VALUE IS 014631
3869 :ERRONEOUS ANSWER VALUE IS IN R2
3870 013306 020327 031462 67$: CMP R3,#031462 :CHECK LOWER WORD OF ANSWER
3871 013312 001401 BEQ 68$ :BR IF OK
3872 013314 104005 ERROR 5 :*****TEST 46 - ERROR 5*****
3873 :LOWER WORD OF ANSWER IS IN ERROR
3874 :EXPECTED VALUE IS 031462
3875 :ERRONEOUS ANSWER IS IN R3
3876 013316 020467 165332 68$: CMP R4,DSTLN :CHECK R4 UNCHANGED
3877 013322 001401 BEQ 69$
3878 013324 104006 ERROR 6 :*****TEST 46 - ERROR 6*****
3879 :R4 CHANGED
3880 :R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"
3881 013326 020567 165324 69$: CMP R5,DSTAD :CHECK R5 UNCHANGED
3882 013332 001401 BEQ 70$ :BR IF OK
3883 013334 104007 ERROR 7 :*****TEST 46 - ERROR 7*****
3884 :R5 CHANGED
3885 :R5 SHOULD STILL EQUAL THE CONTENTS OF "DSTAD"
3886 013336 70$: BR TST47 :BR TO NEXT TEST
3887 013336 000405 ST46: :SOURCE STRING
3888 013340 :MOST SIGNIFICANT DIGIT
3889 013340 064 .BYTE 64
3890 013341 062 .BYTE 62
3891 013342 071 .BYTE 71
3892 013343 064 .BYTE 64
3893 013344 071 .BYTE 71
3894 013345 066 .BYTE 66
3895 013346 067 .BYTE 67
3896 013347 062 .BYTE 62

```

3897 013350 071
3898 013351 066
3899
3900
3901
3902
3903
3904
3905 013352 000004
3906 013354 004567 001352
3907 013360 000012
3908 013362 013514
3909 013364 177777
3910 013366 177777
3911 013370 000377
3912 013372 004567 001446
3913 013376 000212
3914 013400 004767 001354
3915 013404 000277
3916 013406 000242
3917
3918 013410 076053
3919
3920 013412 106767 165260
3921 013416 042767 177400 165252
3922 013424 023767 000700 165244
3923 013432 001401
3924 013434 104001
3925
3926
3927
3928 013436
3929 013436 005700
3930 013440 001401
3931 013442 104002
3932
3933 013444 005701
3934 013446 001401
3935 013450 104003
3936
3937 013452 020227 100000
3938 013456 001401
3939 013460 104004
3940
3941
3942
3943 013462 020327 000000
3944 013466 001401
3945 013470 104005
3946
3947
3948
3949 013472 020467 165156
3950 013476 001401
3951 013500 104006
3952

.BYTE 71
.BYTE 66

*TEST 47 TEST "CVTNL" WITH SRC = +2,147,483,648, OVERFLOW

TST47: SCOPE
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
12 ;SOURCE LENGTH
ST47 ;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 212
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
CLV
CVTNL
MFPS CCODES ;STORE RESULTANT PSW
BIC #177400,CCODES ;CLEAR UNUSED BITS
CMP @WEXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64\$;BR, IF EQUAL
ERROR 1 ;*****TEST 47 - 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 47 - ERROR 2*****
;R0 SHOULD BE ZERO
65\$: TST R1 ;CHECK R1=0
BEQ 66\$
ERROR 3 ;*****TEST 47 - ERROR 3*****
;R1 SHOULD BE ZERO
;CHECK UPPER WORD OF ANSWER
66\$: CMP R2,#100000 ;CHECK UPPER WORD OF ANSWER
BEQ 67\$;BR IF OK
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\$;BR IF OK
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 ;CHECK R4 UNCHANGED
BEQ 69\$
ERROR 6 ;*****TEST 47 - ERROR 6*****
;R4 CHANGED


```
3953  
3954 013502 020567 165150 69$: CMP R5,DSTAD :R4 SHOULD STILL EQUAL TH CONTENTS OF 'FILL'  
3955 013506 001401 BEQ 70$ :CHECK R5 UNCHANGED  
3956 013510 104007 ERROR 7 :BR IF OK  
3957 :*****TEST 47 - ERROR 7*****  
3958 :R5 CHANGED  
3959 :R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'  
3960 013512 000405 70$: BR TST50 :BR TO NEXT TEST  
3961 013514 ST47: :SOURCE STRING  
3962 013514 062 .BYTE 62 :MOST SIGNIFICANT DIGIT  
3963 013515 061 .BYTE 61  
3964 013516 064 .BYTE 64  
3965 013517 067 .BYTE 67  
3966 013520 064 .BYTE 64  
3967 013521 070 .BYTE 70  
3968 013522 063 .BYTE 63  
3969 013523 066 .BYTE 66  
3970 013524 064 .BYTE 64  
3971 013525 070 .BYTE 70  
3972  
3973  
3974  
3975  
3976  
3977  
3978 013526 000004 TST50: SCOPE :*****  
3979 013530 004567 001176 JSR R5,NPREP :*TEST 50 TEST "CVTNL" WITH SRC = -2,147,483,648  
3980 013534 000012 JSR 12 :*****  
3981 013536 013670 ST50 :PREPARE ARGUMENTS FOR INSTRUCTION TEST  
3982 013540 177777 177777 :SOURCE LENGTH  
3983 013542 177777 177777 :SOURCE ADDRESS  
3984 013544 000377 377 :STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS  
3985 013546 004567 001272 JSR R5,XPSW :STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"  
3986 013552 000210 .WORD 210  
3987 013554 004767 001200 JSR PC,GENR :SET UP GENERAL REGISTERS  
3988 013560 000277 SCC :SET UP THE COMPLEMENT OF EXPECTED CC'S  
3989 013562 000250 CLN  
3990  
3991 013564 076053 CVTNL  
3992  
3993 013566 106767 165104 MFPS CCODES :STORE RESULTANT PSW  
3994 013572 042767 177400 165076 BIC #177400,CCODES :CLEAR UNUSED BITS  
3995 013600 023767 000700 165070 CMP @EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE  
3996 013606 001401 BEQ 64$ :BR, IF EQUAL  
3997 013610 104001 ERROR 1 :*****TEST 50 - ERROR 1*****  
3998 :PSW ERROR  
3999 :EXPECTED PSW IS STORED AT 'SAVR6'  
4000 :ERRONEOUS SP VALUE IS AT 'BADR6'  
4001 013612 64$: TST R0 :CHECK R0=0  
4002 013612 005700 BEQ 65$  
4003 013614 001401 ERROR 2 :*****TEST 50 - ERROR 2*****  
4004 013616 104002 :R0 SHOULD BE ZERO  
4005 :CHECK R1=0  
4006 013620 005701 65$: TST R1  
4007 013622 001401 BEQ 66$  
4008 013624 104003 ERROR 3 :*****TEST 50 - ERROR 3*****
```

```
4009                                     ;R1 SHOULD BE ZERO
4010 013626 020227 100000 66$: CMP R2,#100000 ;CHECK UPPER WORD OF ANSWER
4011 013632 001401 BEQ 67$ ;BR IF OK
4012 013634 104004 ERROR 4 ;*****TEST 50 - ERROR 4*****
4013                                     ;UPPER WORD OF ANSWER IS IN ERROR
4014                                     ;EXPECTED VALUE IS 100000
4015                                     ;ERRONEOUS ANSWER VALUE IS IN R2
4016 013636 020327 000000 67$: CMP R3,#0 ;CHECK LOWER WORD OF ANSWER
4017 013642 001401 BEQ 68$ ;BR IF OK
4018 013644 104005 ERROR 5 ;*****TEST 50 - ERROR 5*****
4019                                     ;LOWER WORD OF ANSWER IS IN ERROR
4020                                     ;EXPECTED VALUE IS 0
4021                                     ;ERRONEOUS ANSWER IS IN R3
4022 013646 020467 165002 68$: CMP R4,DSTLN ;CHECK R4 UNCHANGED
4023 013652 001401 BEQ 69$
4024 013654 104006 ERROR 6 ;*****TEST 50 - ERROR 6*****
4025                                     ;R4 CHANGED
4026                                     ;R4 SHOULD STILL EQUAL TH CONTENTS OF 'FILL'
4027 013656 020567 164774 69$: CMP R5,DSTAD ;CHECK R5 UNCHANGED
4028 013662 001401 BEQ 70$ ;BR IF OK
4029 013664 104007 ERROR 7 ;*****TEST 50 - ERROR 7*****
4030                                     ;R5 CHANGED
4031                                     ;R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
4032 013666 70$: BR TST51 ;BR TO NEXT TEST
4033 013666 000405 ST50: ;SOURCE STRING
4034 013670 ;MOST SIGNIFICANT DIGIT
4035 013670 062 .BYTE 62
4036 013671 061 .BYTE 61
4037 013672 064 .BYTE 64
4038 013673 067 .BYTE 67
4039 013674 064 .BYTE 64
4040 013675 070 .BYTE 70
4041 013676 063 .BYTE 63
4042 013677 066 .BYTE 66
4043 013700 064 .BYTE 64
4044 013701 170 .BYTE 170
4045
4046
4047
4048
4049 ;*****
4050 ;*TEST 51 TEST "CVTNL" WITH SRC = -2,147,483,649, OVERFLOW
4051 013702 000004 TST51: SCOPE ;*****
4052 013704 004567 001022 JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4053 013710 000012 12 ;SOURCE LENGTH
4054 013712 014044 ST51 ;SOURCE ADDRESS
4055 013714 177777 177777 ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
4056 013716 177777 177777
4057 013720 000377 377 ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
4058 013722 004567 001116 JSR R5,XPSW
4059 013726 000202 .WORD 202
4060 013730 004767 001024 JSR PC,GENR ;SET UP GENERAL REGISTERS
4061 013734 000265 +SEZ!SEC ;SET UP THE COMPLEMENT OF EXPECTED CC'S
4062 013736 000252 +CLN!CLV
4063
4064 013740 076053 CVTNL
```



```
4121 .....  
4122 .....  
4123 .....  
4124 014056 000004 .....  
4125 014060 004567 000646 .....  
4126 014064 000002 .....  
4127 014066 014220 .....  
4128 014070 177777 .....  
4129 014072 177777 .....  
4130 014074 000377 .....  
4131 014076 004567 000742 .....  
4132 014102 000204 .....  
4133 014104 004767 000650 .....  
4134 014110 000277 .....  
4135 014112 000244 .....  
4136 .....  
4137 014114 076053 .....  
4138 .....  
4139 014116 106767 164554 .....  
4140 014122 042767 177400 164546 .....  
4141 014130 023767 000700 164540 .....  
4142 014136 001401 .....  
4143 014140 104001 .....  
4144 .....  
4145 .....  
4146 .....  
4147 014142 ..... 64$: .....  
4148 014142 005700 .....  
4149 014144 001401 .....  
4150 014146 104002 .....  
4151 .....  
4152 014150 005701 ..... 65$: .....  
4153 014152 001401 .....  
4154 014154 104003 .....  
4155 .....  
4156 014156 020227 000000 ..... 66$: .....  
4157 014162 001401 .....  
4158 014164 104004 .....  
4159 .....  
4160 .....  
4161 .....  
4162 014166 020327 000000 ..... 67$: .....  
4163 014172 001401 .....  
4164 014174 104005 .....  
4165 .....  
4166 .....  
4167 .....  
4168 014176 020467 164452 ..... 68$: .....  
4169 014202 001401 .....  
4170 014204 104006 .....  
4171 .....  
4172 .....  
4173 014206 020567 164444 ..... 69$: .....  
4174 014212 001401 .....  
4175 014214 104007 .....  
4176 .....  
:*****  
:*TEST 52 TEST "CVTNL" WITH SRC LENGTH = 1, SOURCE= 60,60  
:*****  
†ST52: SCOPE  
JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST  
2 ;SOURCE LENGTH  
ST52 ;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 204  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
SCC ;SET UP THE COMPLEMENT OF EXPECTED CC'S  
CLZ  
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 52 - 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 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  
66$: CMP R2,#0 ;CHECK UPPER WORD OF ANSWER  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 52 - ERROR 4*****  
;UPPER WORD OF ANSWER IS IN ERROR  
;EXPECTED VALUE IS 0  
;ERRONEOUS ANSWER VALUE IS IN R2  
67$: CMP R3,#0 ;CHECK LOWER WORD OF ANSWER  
BEQ 68$ ;BR IF OK  
ERROR 5 ;*****TEST 52 - 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 52 - ERROR 6*****  
;R4 CHANGED  
;R4 SHOULD STILL EQUAL TH CONTENTS OF "FILL"  
69$: CMP R5,DSTAD ;CHECK R5 UNCHANGED  
BEQ 70$ ;BR IF OK  
ERROR 7 ;*****TEST 52 - ERROR 7*****  
;R5 CHANGED
```



```
4177                                     ;R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
4178 014216                               70$:
4179 014216 000401                         BR      TST53      ;BR TO NEXT TEST
4180 014220                               ST52:          ;SOURCE STRING
4181 014220      060                       .BYTE 60      ;MOST SIGNIFICANT DIGIT
4182 014221      060                       .BYTE 60
4183
4184
4185
4186
4187
4188 014222 000004                               ;*****
4189 014224 105777 164310                       ;*TEST 53      TEST INTERRUPTABILITY OF "CVTNL"
4190 014230 100532                               ;*****
4191 014232 026767 164312 164424           TST53: SCOPE
4192 014240 001007                               TSTB @SWR      ;TEST BIT 7 OF SWR
4193 014242 032767 000001 164336           BMI SEOP      ;SKIP TO NEXT TEST IF SET
4194 014250 001403                               CMP STPS,TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
4195 014252 005767 164316           BNE CVTCONT  ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
4196 014256 001117                               BIT #BIT0,$ENV ;IF YES, IS PROGRAM UNDER APT?
4197 014260                               BEQ CVTCONT  ;BR, IF NOT
4198 014260 004567 000446           TST SPASS    ;IF YES,CHECK IF NOT ON FIRST PASS
4199 014264 000012                               BNE SEOP     ;IF NOT, BR & SKIP TEST
4200 014266 014472                               CVTCONT:
4201 014270 177777                               JSR R5,NPREP ;PREPARE ARGUMENTS FOR INSTRUCTION TEST
4202 014272 177777                               12          ;SOURCE LENGTH
4203 014274 000377                               ST53        ;SOURCE ADDRESS
4204 014276 012767 014356 164370           177777     ;STORE ALL ONES TO LOAD INTO R2 & R3 REGISTERS
4205 014304 012777 015114 164356           377        ;STORE A NON-ZERO VALUE TO TEST R4 UNAFFECTED BY "CVTZL"
4206 014312 005077 164354                               MOV #CVTPC,PC ;STORE PC OF TEST INSTRUCTION
4207 014316 004767 000546                               MOV #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
4208 014322 013777 000554 164336           CLR @TPSW    ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
4209 014330 004567 000510           JSR PC,TDONE ;WAIT FOR SLU READY
4210 014334 000000                               JSR @#NULL,@TBUF ;SEND NULL CHARACTER
4211 014336 106427 000000                               JSR R5,XPSW   ;STORE EXPECTED PSW
4212 014342 052777 000100 164314           .WORD 0
4213 014350 004767 000404           MTPS #0      ;ALLOW INTERRUPTS
4214 014354 000277                               BIS #100,@TCSR ;ENABLE TTY INTERRUPTS
4215
4216 014356 076053                               RECVTN: JSR PC,GENR ;SET UP GENERAL REGISTERS
4217
4218 014360 106767 164312                               SCC        ;SET UP THE COMPLEMENT OF EXPECTED CC'S
4219 014364 032777 000100 164272           CVTPC: CVTNL
4220 014372 001366                               MFPS CCODES  ;STORE RESULTANT PSW
4221 014374 042767 177400 164274           BIT #100,@TCSR ;IF INTERRUPTS ARE DISABLED, INSTRUCTION WAS INTERRUPTED
4222 014402 023767 000700 164266           BNE RECVTN  ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
4223 014410 001401                               BIC #177400,CCODES ;CLEAR UNUSED BITS
4224 014412 104001                               CMP @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
4225
4226
4227
4228 014414                               BEQ 64$     ;BR, IF EQUAL
4229 014414 005700                               ERROR 1     ;*****TEST 53 - ERROR 1*****
4230 014416 001401                               ;PSW ERROR
4231 014420 104002                               ;EXPECTED PSW IS STORED AT "SAVR6"
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
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
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
```

```

4233 014422 005701      65$:  TST      R1      ;CHECK R1=0
4234 014424 001401      BEQ      66$
4235 014426 104003      ERROR    3      ;*****TEST 53 - ERROR 3*****
4236                                ;R1 SHOULD BE ZERO
4237 014430 020227 052525  66$:  CMP      R2,#052525 ;CHECK UPPER WORD OF ANSWER
4238 014434 001401      BEQ      67$
4239 014436 104004      ERROR    4      ;BR IF OK
4240                                ;*****TEST 53 - ERROR 4*****
4241                                ;UPPER WORD OF ANSWER IS IN ERROR
4242                                ;EXPECTED VALUE IS 052525
4243 014440 020327 125252  67$:  CMP      R3,#125252 ;ERRONEOUS ANSWER VALUE IS IN R2
4244 014444 001401      BEQ      68$
4245 014446 104005      ERROR    5      ;CHECK LOWER WORD OF ANSWER
4246                                ;BR IF OK
4247                                ;*****TEST 53 - ERROR 5*****
4248                                ;LOWER WORD OF ANSWER IS IN ERROR
4249 014450 020467 164200  68$:  CMP      R4,DSTLN  ;EXPECTED VALUE IS 125252
4250 014454 001401      BEQ      69$
4251 014456 104006      ERROR    6      ;ERRONEOUS ANSWER IS IN R3
4252                                ;CHECK R4 UNCHANGED
4253                                ;*****TEST 53 - ERROR 6*****
4254 014460 020567 164172  69$:  CMP      R5,DSTAD  ;R4 CHANGED
4255 014464 001401      BEQ      70$
4256 014466 104007      ERROR    7      ;R4 SHOULD STILL EQUAL TH CONTENTS OF 'FILL'
4257                                ;CHECK R5 UNCHANGED
4258                                ;BR IF OK
4259 014470                                ;*****TEST 53 - ERROR 7*****
4260 014470 000405      70$:  BR      ENDT53    ;R5 CHANGED
4261 014472                                ;R5 SHOULD STILL EQUAL THE CONTENTS OF 'DSTAD'
4262 014472      061      ST53:  .BYTE    61      ;BR TO END OF THIS TEST
4263 014473      064      .BYTE    64      ;SOURCE STRING
4264 014474      063      .BYTE    63      ;MOST SIGNIFICANT DIGIT
4265 014475      061      .BYTE    61
4266 014476      066      .BYTE    66
4267 014477      067      .BYTE    67
4268 014500      067      .BYTE    67
4269 014501      066      .BYTE    66
4270 014502      061      .BYTE    61
4271 014503      060      .BYTE    60
4272
4273 014504 016777 164162 164156 ENDT53: MOV     TPSW,@TVECT
4274 014512 106427 000200      MTPS     #200
4275      .SBTTL  END OF PASS ROUTINE
4276
4277      ;*****
4278      ;*INCREMENT THE PASS NUMBER ($PASS)
4279      ;*IF SW12=1 INHIBIT TRACE TRAP
4280      ;*IF THERES A MONITOR GO TO IT
4281      ;*IF THERE ISN'T JUMP TO BEGIN
4282
4283      $EOP:
4284 014516 000004      SCOPE
4285 014520 005067 163756      CLR     $TSTNM  ;: ZERO THE TEST NUMBER
4286 014524 005267 164044      INC     $PASS   ;: INCREMENT THE PASS NUMBER
4287 014530 042767 100000 164036 BIC     #100000,$PASS ;: DON'T ALLOW A NEG. NUMBER
4288 014536 005327      DEC     (PC)+  ;: LOOP?

```



```

4345                                     : 'OLDPC' CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4346 014720 000000                       HALT                               : PROGRAM MUST BE RESTARTED AT THIS POINT
4347
4348
4349                                     : ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 10
4350 014722 011637 000706               ILLTRP: MOV      (SP),@#OLDPC      : GET PC+2 WHERE UNEXPECTED ILLEGAL INSTRUCTION TRAP OCCU
4351 014726 104210                       ERROR 210                        : *****ERROR 210*****
4352                                     : UNEXPECTED TRAP TO LOCATION 10
4353                                     : 'OLDPC' CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4354 014730 000000                       HALT                               : PROGRAM MUST BE RESTARTED AT THIS POINT
4355
4356
4357
4358
4359                                     : SUBROUTINE TO SET OPERAND VALUES
4360
4361 014732 012567 163706                 NPREP: MOV      (R5)+,S1LN        : STORE INSTRUCTION TEST ARGUMENTS
4362 014736 012567 163704                 MOV      (R5)+,S1ADR
4363 014742 012567 163702                 MOV      (R5)+,S2LN
4364 014746 012567 163700                 MOV      (R5)+,S2ADR
4365 014752 012567 163676                 MOV      (R5)+,DSTLN
4366 014756 000205                       RTS      R5
4367
4368
4369                                     : SUBROUTINE TO SET UP GENERAL REGISTERS
4370
4371 014760 016700 163660                 GENR:  MOV      S1LN,R0           : TRANSFER INSTRUCTION TEST ARGUMENTS TO
4372 014764 016701 163656                 MOV      S1ADR,R1              : THE GENERAL REGISTERS
4373 014770 016702 163654                 MOV      S2LN,R2
4374 014774 016703 163652                 MOV      S2ADR,R3
4375 015000 016704 163650                 MOV      DSTLN,R4
4376 015004 016705 163646                 MOV      DSTAD,R5
4377 015010 010637 000702                 MOV      SP,@#SAVR6           : COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
4378 015014 062737 000002 000702         ADD      #2,@#SAVR6           : ADJUST SAVED SP BECAUSE OF JSR TO THIS ROUTINE
4379 015022 000207                       RTS      PC
4380
4381
4382
4383                                     : SUBROUTINE TO CLEAR BUFFER AREA
4384 015024 012700 017044                 CLBUF: MOV      #BUF,R0        : POINT R0 TO BUFFER AREA
4385 015030 012701 000020                 MOV      #20,R1              : STORE BUFFER LENGTH IN R1
4386 015034 005020                       1$: CLR      (R0)+            : CLEAR BUFFER
4387 015036 005301                       DEC      R1                  : DECREMENT BUFFER LENGTH
4388 015040 001375                       BNE     1$                  : BR IF NOT FINISHED
4389 015042 000207                       RTS      PC                  : RETURN
4390
4391                                     : SUBROUTINE TO RECORD EXPECTED PSW
4392 015044 012537 000700                 XPSW: MOV      (R5)+,@#EXPPSW  : STORE EXPECTED PSW VALUE IN 'EXPPSW'
4393 015050 106700                       MFPS     R0                  : GET PRESENT PSW
4394 015052 032700 000020                 BIT      #TBIT,R0            : IS T-BIT SET?
4395 015056 001403                       BEQ     1$                  : BR IF NOT
4396 015060 052737 000020 000700         BIS      #TBIT,@#EXPPSW     : OTHERWISE SET T-BIT IN EXPECTED PSW VALUE
4397 015066 000205                       1$: RTS      R5              : RETURN
4398
4399
4400

```



```

4401
4402      :SUBROUTINE TO TEST FOR TRANSMIT DONE FLAG
4403 015070 005037 000710 TDONE: CLR @TEMP ;CLEAR TIMER
4404 015074 105777 163564 1$: TSTB @TCSR ;IS SLU READY?
4405 015100 100404 BMI RETN ;BR IF READY
4406 015102 005237 000710 INC @TEMP ;OTHERWISE INCREMENT TIMER
4407 015106 001372 BNE 1$ ;BR IF NOT TIMED OUT
4408 015110 104300 ERROR 300 ;*****ERROR 300*****
4409 RETN: RTS PC ;NEVER GOT TRANSMIT DONE FLAG
4410 ;RETURN
4411
4412      :SUBROUTINE TO HANDLE TTY INTERRUPTS IN INSTRUCTION
4413      :INTERRUPTABILITY TESTS
4414
4415
4416 015114 INTR: CMP (SP),PCI ;WAS PC AT INSTRUCTION UNDER TEST?
4417 015114 021667 163554 BNE SEND ;BR, IF NO
4418 015120 001003 CKR4: BIT #177400,R4 ;IF YES, CHECK UPPER BYTE OF R4
4419 015122 032704 177400 BNE CLRINT ;IF ZERO, INSTRUCTION WAS NOT INTERRUPTED-TRY AGAIN
4420 015126 001004 SEND: MOV @NULL,@TBUF ;SEND ANOTHER CHARACTER
4421 015130 013777 000554 163530 RTI ;RETURN
4422 015136 000002 CLRINT: BIC #100,@TCSR ;IF NON-ZERO, CLEAR INTERRUPT ENABLE
4423 015140 042777 000100 163516 RTI ;CONTINUE INSTRUCTION
4424 015146 000002
4425
4426
4427
4428      .SBTTL SCOPE HANDLER ROUTINE
4429
4430      ;*****
4431      ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4432      ;*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4433      ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4434      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4435      ;*SW14=1 LOOP ON TEST
4436      ;*SW09=1 LOOP ON ERROR
4437      ;*SW08=1 LOOP ON TEST IN SWR<5:0>
4438      ;*CALL
4439      ;* SCOPE ;:SCOPE=IOT
4440
4441 015150 $SCOPE:
4442 015150 032777 040000 163362 1$: BIT #BIT14,@SWR ;:LOOP ON PRESENT TEST?
4443 015156 001065 BNE $OVER ;:YES IF SW14=1
4444 ;*****START OF CODE FOR THE XOR TESTER*****
4445 015160 000416 $XTSTR: BR 6$ ;:IF RUNNING ON THE 'XOR' TESTER CHANGE
4446 ;THIS INSTRUCTION TO A 'NOP' (NOP=240)
4447 015162 013746 000004 MOV @ERRVEC,-(SP) ;:SAVE THE CONTENTS OF THE ERROR VECTOR
4448 015166 012737 015206 000004 MOV #5,@ERRVEC ;:SET FOR TIMEOUT
4449 015174 005737 177060 TST @177060 ;:TIME OUT ON XOR?
4450 015200 012637 000004 MOV (SP)+,@ERRVEC ;:RESTORE THE ERROR VECTOR
4451 015204 000434 BR $SVLAD ;:GO TO THE NEXT TEST
4452 015206 022626 5$: CMP (SP)+,(SP)+ ;:CLEAR THE STACK AFTER A TIME OUT
4453 015210 012637 000004 MOV (SP)+,@ERRVEC ;:RESTORE THE ERROR VECTOR
4454 015214 000422 BR 7$ ;:LOOP ON THE PRESENT TEST
4455 015216 6$::*****END OF CODE FOR THE XOR TESTER*****
4456 015216 032777 000400 163314 BIT #BIT08,@SWR ;:LOOP ON SPEC. TEST?

```

```

4457 015224 001407 BEQ 2$ ::BR IF NO
4458 015226 017746 163306 MOV @SWR,-(SP) ::SET DESIRED TEST NUM. FROM SWR
4459 015232 042716 000300 BIC #SSWRMK,(SP) ::STRIP AWAY UNDESIRED BITS
4460 015236 122667 163240 CMPB (SP)+,$TSTNM ::ON THE RIGHT TEST?
4461 015242 001433 BEQ $OVER ::BR IF YES
4462 015244 105767 163233 2$: TSTB $ERFLG ::HAS AN ERROR OCCURRED?
4463 015250 001412 BEQ $$VLAD ::BR IF NO
4464 015252 032777 001000 163260 BIT #BIT09,@SWR ::LOOP ON ERROR?
4465 015260 001404 BEQ 4$ ::BR IF NO
4466 015262 016767 163222 163216 7$: MOV $LPERR,$LPADR ::SET LOOP ADDRESS TO LAST SCOPE
4467 015270 000420 BR $OVER
4468 015272 105067 163205 4$: CLRB $ERFLG ::ZERO THE ERROR FLAG
4469 015276 105267 163200 $$VLAD: INCB $TSTNM ::COUNT TEST NUMBERS
4470 015302 116767 163174 163262 MOVB $TSTNM,$TESTN ::SET TEST NUMBER IN APT MAILBOX
4471 015310 011667 163172 MOV (SP),$LPADR ::SAVE SCOPE LOOP ADDRESS
4472 015314 011667 163170 MOV (SP),$LPERR ::SAVE ERROR LOOP ADDRESS
4473 015320 005067 163234 CLR $ESCAPE ::CLEAR THE ESCAPE FROM ERROR ADDRESS
4474 015324 112767 000001 163163 MOVB #1,$ERMAX ::ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4475 015332 016777 163144 163202 $OVER: MOV $TSTNM,@DISPLAY ::DISPLAY TEST NUMBER
4476 015340 016716 163142 MOV $LPADR,(SP) ::FUDGE RETURN ADDRESS
4477 015344 000002 RTI ::FIXES PS

```

.SBTTL POWER DOWN AND UP ROUTINES

```

4478
4479
4480
4481
4482 ::*****
4483 015346 012737 015530 000024 $PWRDN: MOV # $ILLUP,@#PWRVEC ::SET FOR FAST UP
4484 015354 012737 000340 000026 MOV #340,@#PWRVEC+2 ::PRIO:7
4485 015362 010046 MOV R0,-(SP) ::PUSH R0 ON STACK
4486 015364 010146 MOV R1,-(SP) ::PUSH R1 ON STACK
4487 015366 010246 MOV R2,-(SP) ::PUSH R2 ON STACK
4488 015370 010345 MOV R3,-(SP) ::PUSH R3 ON STACK
4489 015372 010446 MOV R4,-(SP) ::PUSH R4 ON STACK
4490 015374 010546 MOV R5,-(SP) ::PUSH R5 ON STACK
4491 015376 017746 163136 MOV @SWR,-(SP) ::PUSH @SWR ON STACK
4492 015402 010667 000126 MOV SP,$SAVR6 ::SAVE SP
4493 015406 012737 015420 000024 MOV # $PWRUP,@#PWRVEC ::SET UP VECTOR
4494 015414 000000 HALT
4495 015416 000776 BR -2 ::HANG UP
4496
4497
4498 ::*****

```

POWER UP ROUTINE

```

4499 015420 012737 015530 000024 $PWRUP: MOV # $ILLUP,@#PWRVEC ::SET FOR FAST DOWN
4500 015426 016706 000102 MOV $SAVR6,SP ::GET SP
4501 015432 005067 000076 CLR $SAVR6 ::WAIT LOOP FOR THE TTY
4502 015436 005267 000072 1$: INC $SAVR6 ::WAIT FOR THE INC
4503 015442 001375 BNE 1$ ::OF WORD
4504 015444 005067 163032 CLR $TSTNM
4505 015450 012677 163064 MOV (SP)+,@SWR ::POP STACK INTO @SWR
4506 015454 012605 MOV (SP)+,R5 ::POP STACK INTO R5
4507 015456 012604 MOV (SP)+,R4 ::POP STACK INTO R4
4508 015460 012603 MOV (SP)+,R3 ::POP STACK INTO R3
4509 015462 012602 MOV (SP)+,R2 ::POP STACK INTO R2
4510 015464 012601 MOV (SP)+,R1 ::POP STACK INTO R1
4511 015466 012600 MOV (SP)+,R0 ::POP STACK INTO R0
4512 015470 012737 015346 000024 MOV # $PWRDN,@#PWRVEC ::SET UP THE POWER DOWN VECTOR

```



```
4513 015476 012737 000340 000026      MOV      #340,@#PWRVEC+2  ;;PRIO:7
4514 015504 104401                    TYPE                                ;;REPORT THE POWER FAILURE
4515 015506 015536      SPWRMG: .WORD  SPOWER      ;;POWER FAIL MESSAGE POINTER
4516 015510 012716      MOV      (PC)+,(SP)      ;;RESTART AT SLOOP
4517 015512 014652      SPWRAD: .WORD  SLOOP      ;;RESTART ADDRESS
4518 015514 042766 000020 000002      BIC      #20,2(SP)      ;;CLEAR 'T' BIT
4519 015522 005067 177130      CLR      $TBIT          ;;CLEAR THE 'T' BIT FLAG
4520 015526 000002      RTI
4521 015530 000000      $ILLUP: HALT          ;;THE POWER UP SEQUENCE WAS STARTED
4522 015532 000776      BR      .-2            ;; BEFORE THE POWER DOWN WAS COMPLETE
4523 015534 000000      $$AVR6: 0              ;;PUT THE SP HERE
4524 015536 005015 047520 042527      $POWER: .ASCIZ  <15><12>'POWER'
4525 015544 000122
4526
4527
4528      .SBTTL  ERROR HANDLER ROUTINE
4529
4530      ;*****
4531      ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
4532      ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4533      ;*AND TYPE OUT THE PC OF THE ERROR INSTRUCTION
4534      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4535      ;*SW15=1      HALT ON ERROR
4536      ;*SW13=1      INHIBIT ERROR TYPEOUTS
4537      ;*SW09=1      LOOP ON ERROR
4538      ;*CALL
4539      ;*      ERROR      N      ;;ERROR=EMT AND N=ERROR ITEM NUMBER
4540
4541 015546      $ERROR:
4542 015546 105267 162731      7$:      INCB      $ERFLG      ;;SET THE ERROR FLAG
4543 015552 001775      BEQ      7$            ;;DON'T LET THE FLAG GO TO ZERO
4544 015554 016777 162722 162760      MOV      $STNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
4545 015562 005267 162724      INC      $ERTTL      ;;INC THE ERROR COUNT
4546 015566 011667 162724      MOV      (SP),$ERRPC   ;;GET ADDRESS OF ERROR INSTRUCTION
4547 015572 162767 000002 162716      SUB      #2,$ERRPC
4548 015600 117767 162712 162706      MOV      @ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
4549 015606 032777 020000 162724      BIT      #BIT13,@SWR   ;;SKIP TYPEOUT IF SET
4550 015614 001007      BNE      20$          ;;SKIP TYPEOUTS
4551 015616 104401 000563      TYPE    $CRLF
4552 015622 016746 162670      MOV      $ERRPC,-(SP)  ;;SAVE $ERRPC FOR TYPEOUT
4553
4554 015626 104402      TYPCC   ;;ERROR ADDRESS
4555 015630 104401 000563      TYPE    $CRLF        ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4556 015634      20$:
4557 015634 122767 000001 162744      CMPB    #APTENV,$ENV   ;;RUNNING IN APT MODE
4558 015642 001007      BNE     2$            ;;NO,SKIP APT ERROR REPORT
4559 015644 116767 162644 000004      MOV     $ITEMB,21$    ;;SET ITEM NUMBER AS ERROR NUMBER
4560 015652 004767 000656      JSR    PC,$ATY4      ;;REPORT FATAL ERROR TO APT
4561 015656 000      21$:      .BYTE  0
4562 015657 000      .BYTE  0
4563 015660 000777      22$:      BR     22$           ;;APT ERROR LOOP
4564 015662 005777 162652      2$:      TST   @SWR          ;;HALT ON ERROR
4565 015666 100001      BPL   3$            ;;SKIP IF CONTINUE
4566 015670 000000      HALT
4567 015672 032777 001000 162640      3$:      BIT   #BIT09,@SWR   ;;LOOP ON ERROR SWITCH SET?
4568 015700 001402      BEQ   4$            ;;BR IF NO
```

```

4569 015702 016716 162602      MOV      $LPERR,(SP)      ;;FUDGE RETURN FOR LOOPING
4570 015706 005767 162646      4$:    TST      $ESCAPE     ;;CHECK FOR AN ESCAPE ADDRESS
4571 015712 001402                BEQ      5$              ;;BR IF NONE
4572 015714 016716 162640      MOV      $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
4573 015720                5$:    CMP      #SENDAD,@#42  ;;ACT-11 AUTO-ACCEPT?
4574 015720 022737 014604 000042  BNE      6$              ;;BRANCH IF NO
4575 015726 001001                HALT                    ;;YES
4576 015730 000000                6$:    RTI                    ;;RETURN
4577 015732
4578 015732 000002
4579
4580      .SBTTL  TYPE ROUTINE
4581
4582      ;*****
4583      ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4584      ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4585      ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4586      ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4587      ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4588      ;*
4589      ;*CALL:
4590      ;*1) USING A TRAP INSTRUCTION
4591      ;*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4592      ;*OR
4593      ;*      TYPE
4594      ;*      MESADR
4595      ;*
4596
4597 015734 105767 162617      $TYPE: TSTB     $TPFLG      ;;IS THERE A TERMINAL?
4598 015740 100002                BPL      1$              ;;BR IF YES
4599 015742 000000                HALT                    ;;HALT HERE IF NO TERMINAL
4600 015744 000430                BR                    ;;LEAVE
4601 015746 010046                1$:    MOV      RO,-(SP)     ;;SAVE RO
4602 015750 017600 000002      MOV      @2(SP),RO      ;;GET ADDRESS OF ASCIZ STRING
4603 015754 122767 000001 162624  CMPB     #APTENV,$ENV    ;;RUNNING IN APT MODE
4604 015762 001011                BNE      62$            ;;NO,GO CHECK FOR APT CONSOLE
4605 015764 132767 000100 162615  BITB     #APTPOOL,$ENVM  ;;SPOOL MESSAGE TO APT
4606 015772 001405                BEQ      62$            ;;NO,GO CHECK FOR CONSOLE
4607 015774 010067 000004      MOV      RO,61$         ;;SETUP MESSAGE ADDRESS FOR APT
4608 016000 004767 000520      JSR      PC,$ATY3       ;;SPOOL MESSAGE TO APT
4609 016004 000000                .WORD    0              ;;MESSAGE ADDRESS
4610 016006 132767 000040 162573 62$:    BITB     #APTCSUP,$ENVM ;;APT CONSOLE SUPPRESSED
4611 016014 001003                BNE      60$            ;;YES,SKIP TYPE OUT
4612 016016 112046                2$:    MOVB     (RO)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
4613 016020 001005                BNE      4$              ;;BR IF IT ISN'T THE TERMINATOR
4614 016022 005726                TST      (SP)+          ;;IF TERMINATOR POP IT OFF THE STACK
4615 016024 012600                60$:    MOV      (SP)+,RO    ;;RESTORE RO
4616 016026 062716 000002      3$:    ADD      #2,(SP)     ;;ADJUST RETURN PC
4617 016032 000002                RTI                    ;;RETURN
4618 016034 122716 000011      4$:    CMPB     #HT,(SP)    ;;BRANCH IF <HT>
4619 016040 001430                BEQ      8$              ;;BRANCH IF NOT <CRLF>
4620 016042 122716 000200      CMPB     #CRLF,(SP)
4621 016046 001006                BNE      5$              ;;POP <CR><LF> EQUIV
4622 016050 005726                TST      (SP)+          ;;TYPE A CR AND LF
4623 016052 104401
4624 016054 000563      $CRLF

```



```

4625 016056 105067 000202          CLRB  $CHARCNT      ;;CLEAR CHARACTER COUNT
4626 016062 000755                    BR    2$           ;;GET NEXT CHARACTER
4627 016064 004767 000056          5$: JSR  PC,$TYPEC  ;;GO TYPE THIS CHARACTER
4628 016070 126726 162462          6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?
4629 016074 001350                    BNE  2$           ;;IF NO GO GET NEXT CHAR.
4630 016076 016746 162452          MGV  $NULL,-(SP)  ;;GET # OF FILLER CHARS. NEEDED
4631                                ;;AND THE NULL CHAR.
4632 016102 105366 000001          7$: DECB 1(SP)     ;;DOES A NULL NEED TO BE TYPED?
4633 016106 002770                    BLT  6$           ;;BR IF NO--GO POP THE NULL OFF OF STACK
4634 016110 004767 000032          JSR  PC,$TYPEC  ;;GO TYPE A NULL
4635 016114 105367 000144          DECB $CHARCNT    ;;DO NOT COUNT AS A COUNT
4636 016120 000770                    BR    7$         ;;LOOP
4637
4638                                ;HORIZONTAL TAB PROCESSOR
4639
4640 016122 112716 000040          8$: MOVB #' ,(SP)  ;;REPLACE TAB WITH SPACE
4641 016126 004767 000014          9$: JSR  PC,$TYPEC  ;;TYPE A SPACE
4642 016132 132767 000007 000124 BITB  #7,$CHARCNT  ;;BRANCH IF NOT AT
4643 016140 001372                    BNE  9$           ;;TAB STOP
4644 016142 005726                    TST  (SP)+       ;;POP SPACE OFF STACK
4645 016144 000724                    BR    2$         ;;GET NEXT CHARACTER
4646 016146                                $TYPEC:
4647 016146 105777 162372          TSTB @STKS       ;;CHAR IN KYBD BUFFER?
4648 016152 100022                    BPL  10$         ;;BR IF NOT
4649 016154 017746 162366          MOV  @STKB,-(SP) ;;GET CHAR
4650 016160 042716 177600          BIC  #177600,(SP) ;;STRIP EXTRANEIOUS BITS
4651 016164 122716 000023          CMPB #$XOFF,(SP) ;;WAS CHAR XOFF
4652 016170 001012                    BNE  102$        ;;BR IF NOT
4653 016172                                101$:
4654 016172 105777 162346          TSTB @STKS       ;;WAIT FOR CHAR
4655 016176 100375                    BPL  101$        ;;
4656 016200 117716 162342          MOVB @STKB,(SP)  ;;GET CHAR
4657 016204 042716 177600          BIC  #177600,(SP) ;;STRIP IT
4658 016210 122716 000021          CMPB #$XON,(SP) ;;WAS IT XON?
4659 016214 001366                    BNE  101$        ;;BR IF NOT
4660 016216                                102$:
4661 016216 005726                    TST  (SP)+       ;;FIX STACK
4662 016220                                10$:
4663 016220 105777 162324          TSTB @STPS       ;;WAIT UNTIL PRINTER IS READY
4664 016224 100375                    BPL  10$         ;;
4665 016226 116677 000002 162316 MOVB  2(SP),@STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4666 016234 122766 000015 000002 CMPB  #CR,2(SP)  ;;IS CHARACTER A CARRIAGE RETURN?
4667 016242 001003                    BNE  1$           ;;BRANCH IF NO
4668 016244 105067 000014          CLRB $CHARCNT    ;;YES--CLEAR CHARACTER COUNT
4669 016250 000406                    BR    $TYPEX     ;;EXIT
4670 016252 122766 000012 000002 1$: CMPB  #LF,2(SP)  ;;IS CHARACTER A LINE FEED?
4671 016260 001402                    BEQ  $TYPEX     ;;BRANCH IF YES
4672 016262 105227                    INCB (PC)+       ;;COUNT THE CHARACTER
4673 016264 000000          $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
4674 016266 000207          $TYPEX: RTS     PC
4675
4676
4677                                .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
4678
4679                                ;*****
4680                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT

```

```

4681      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
4682      ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4683      ;*CALL:
4684      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4685      ;*      TYPOS      ;;CALL FOR TYPEOUT
4686      ;*      .BYTE    N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4687      ;*      .BYTE    M      ;;M=1 OR 0
4688      ;*                               ;;1=TYPE LEADING ZEROS
4689      ;*                               ;;0=SUPPRESS LEADING ZEROS
4690
4691      ;*STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
4692      ;*STYPOS OR STYPOC
4693      ;*CALL:
4694      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4695      ;*      TYPON      ;;CALL FOR TYPEOUT
4696
4697      ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
4698      ;*CALL:
4699      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
4700      ;*      TYPOC      ;;CALL FOR TYPEOUT
4701
4702      016270 017646 000000      $TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
4703      016274 116667 000001 000211      MOV      1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
4704      016302 112667 000207      MOV      (SP)+,SOMODE+1      ;;NUMBER OF DIGITS TO TYPE
4705      016306 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
4706      016312 000406      BR      STYPON
4707      016314 112767 000001 000171      $TYPOC: MOV      #1,SOFILL      ;;SET THE ZERO FILL SWITCH
4708      016322 112767 000006 000165      MOV      #6,SOMODE+1      ;;SET FOR SIX(6) DIGITS
4709      016330 112767 000005 000154      $TYPON: MOV      #5,SOCNT      ;;SET THE ITERATION COUNT
4710      016336 010346      MOV      R3,-(SP)      ;;SAVE R3
4711      016340 010446      MOV      R4,-(SP)      ;;SAVE R4
4712      016342 010546      MOV      R5,-(SP)      ;;SAVE R5
4713      016344 116704 000145      MOV      SOMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
4714      016350 005404      NEG      R4
4715      016352 062704 000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
4716      016356 110467 000132      MOV      R4,SOMODE      ;;SAVE IT FOR USE
4717      016362 116704 000125      MOV      SOFILL,R4      ;;GET THE ZERO FILL SWITCH
4718      016366 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
4719      016372 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
4720      016374 006105      1$: ROL      R5      ;;ROTATE MSB INTO 'C'
4721      016376 000404      BR      3$      ;;GO DO MSB
4722      016400 006105      2$: ROL      R5      ;;FORM THIS DIGIT
4723      016402 006105      ROL      R5
4724      016404 006105      ROL      R5
4725      016406 010503      MOV      R5,R3
4726      016410 006103      3$: ROL      R3      ;;GET LSB OF THIS DIGIT
4727      016412 105367 000076      DECB    SOMODE      ;;TYPE THIS DIGIT?
4728      016416 100016      BPL     7$      ;;BR IF NO
4729      016420 042703 177770      BIC     #177770,R3      ;;GET RID OF JUNK
4730      016424 001002      BNE     4$      ;;TEST FOR 0
4731      016426 005704      TST     R4      ;;SUPPRESS THIS 0?
4732      016430 001403      BEQ     5$      ;;BR IF YES
4733      016432 005204      4$: INC     R4      ;;DON'T SUPPRESS ANYMORE 0'S
4734      016434 052703 000060      BIS     #'0,R3      ;;MAKE THIS DIGIT ASCII
4735      016440 052703 000040      5$: BIS     #' ,R3      ;;MAKE ASCII IF NOT ALREADY
4736      016444 110367 000040      MOV     R3,8$      ;;SAVE FOR TYPING

```



```

4737 016450 104401 016510
4738 016454 105367 000032 7$: TYPE 8$      ::GO TYPE THIS DIGIT
4739 016460 003347      DECB $OCNT      ::COUNT BY 1
4740 016462 002402      BGT 2$         ::BR IF MORE TO DO
4741 016464 005204      BLT 6$         ::BR IF DONE
4742 016466 000744      INC R4         ::INSURE LAST DIGIT ISN'T A BLANK
4743 016470 012605      BR 2$         ::GO DO THE LAST DIGIT
4744 016472 012604      6$: MOV (SP)+,R5  ::RESTORE R5
4745 016474 012603      MOV (SP)+,R4  ::RESTORE R4
4746 016476 016666 000002 000004 MOV (SP)+,R3  ::RESTORE R3
4747 016504 012616      MOV 2(SP),4(SP) ::SET THE STACK FOR RETURNING
4748 016506 000002      MOV (SP)+,(SP)
4749 016510 000      RTI          ::RETURN
4750 016511 000      8$: .BYTE 0   ::STORAGE FOR ASCII DIGIT
4751 016512 000      .BYTE 0     ::TERMINATOR FOR TYPE ROUTINE
4752 016513 000      S$OCNT: .BYTE 0 ::OCTAL DIGIT COUNTER
4753 016514 000000      $OFILL: .BYTE 0 ::ZERO FILL SWITCH
4754      SOMODE: .WORD 0 ::NUMBER OF DIGITS TO TYPE
4755      .SBTTL APT COMMUNICATIONS ROUTINE
4756
4757 *****
4758 016516 112767 000001 000236 $ATY1: MOV #1,$FFLG ::TO REPORT FATAL ERROR
4759 016524 112767 000001 000226 $ATY3: MOV #1,$MFLG ::TO TYPE A MESSAGE
4760 016532 000403      BR $ATYC
4761 016534 112767 000001 000220 $ATY4: MOV #1,$FFLG ::TO ONLY REPORT FATAL ERROR
4762 016542      $ATYC:
4763 016542 010046      MOV R0,-(SP)   ::PUSH R0 ON STACK
4764 016544 010146      MOV R1,-(SP)   ::PUSH R1 ON STACK
4765 016546 105767 000206      TSTB $MFLG    ::SHOULD TYPE A MESSAGE?
4766 016552 001450      BEQ 5$        ::IF NOT: BR
4767 016554 122767 000001 162024 CMPB #APTENV,$ENV ::OPERATING UNDER APT?
4768 016562 001031      BNE 3$        ::IF NOT: BR
4769 016564 132767 000100 162015 BITB #APTPOOL,$ENVM ::SHOULD SPOOL MESSAGES?
4770 016572 001425      BEQ 3$        ::IF NOT: BR
4771 016574 017600 000004      MOV @4(SP),R0  ::GET MESSAGE ADDR.
4772 016600 062766 000002 000004 ADD #2,4(SP)    ::BUMP RETURN ADDR.
4773 016606 005767 161754      1$: TST $MSGTYPE ::SEE IF DONE W/ LAST XMISSION?
4774 016612 001375      BNE 1$        ::IF NOT: WAIT
4775 016614 010067 161762      MOV R0,$MSGAD  ::PUT ADDR IN MAILBOX
4776 016620 105720      2$: TSTB (R0)+  ::FIND END OF MESSAGE
4777 016622 001376      BNE 2$
4778 016624 166700 161752      SUB $MSGAD,R0  ::SUB START OF MESSAGE
4779 016630 006200      ASR R0        ::GET MESSAGE LNTH IN WORDS
4780 016632 010067 161746      MOV R0,$MSGGLT ::PUT LENGTH IN MAILBOX
4781 016636 012767 000004 161722 MOV #4,$MSGTYPE ::TELL APT TO TAKE MSG.
4782 016644 000413      BR 5$
4783 016646 017667 000004 000016 3$: MOV @4(SP),4$  ::PUT MSG ADDR IN JSR LINKAGE
4784 016654 062766 000002 000004 ADD #2,4(SP)    ::BUMP RETURN ADDRESS
4785 016662 016746 161110      MOV 177776,-(SP) ::PUSH 177776 ON STACK
4786 016666 004767 177042      JSR PC,$TYPE  ::CALL TYPE MACRO
4787 016672 000000      4$: .WORD 0
4788 016674
4789 016674 105767 000062      5$:
4790 016700 001416      10$: TSTB $FFLG    ::SHOULD REPORT FATAL ERROR?
4791 016702 005767 161700      BEQ 12$       ::IF NOT: BR
4792 016706 001413      TST $ENV      ::RUNNING UNDER APT?
      BEQ 12$    ::IF NOT: BR

```

```

4793 016710 005767 161652
4794 016714 001375
4795 016716 017667 000004 161644
4796 016724 062766 000002 000004
4797 016732 005267 161630
4798 016736 105067 000020
4799 016742 105067 000013
4800 016746 105067 000006
4801 016752 012601
4802 016754 012600
4803 016756 000207
4804 016760 000
4805 016761 000
4806 016762 000
4807 016764
4808 000200
4809 000001
4810 000100
4811 000040
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821 016764 010046
4822 016766 016600 000002
4823 016772 005740
4824 016774 111000
4825 016776 006300
4826 017000 016000 017020
4827 017004 000200
4828
4829
4830
4831
4832 017006 011646
4833 017010 016666 000004 000002
4834 017016 000002
4835
4836
4837
4838
4839
4840
4841
4842
4843 017020 017006
4844 017022 015734
4845 017024 016314
4846 017026 016270
4847 017030 016330
4848

```

```

11$: TST $MSGTYPE ;;FINISHED LAST MESSAGE?
      BNE 11$ ;;IF NOT: WAIT
      MOV @4(SP), $FATAL ;;GET ERROR #
      ADD #2, 4(SP) ;;BUMP RETURN ADDR.
      INC $MSGTYPE ;;TELL APT TO TAKE ERROR
12$: CLRB $FFLG ;;CLEAR FATAL FLAG
      CLRB $LFLG ;;CLEAR LOG FLAG
      CLRB $MFLG ;;CLEAR MESSAGE FLAG
      MOV (SP)+, R1 ;;POP STACK INTO R1
      MOV (SP)+, R0 ;;POP STACK INTO R0
      RTS PC ;;RETURN
      $MFLG: .BYTE 0 ;;MESSG. FLAG
      $LFLG: .BYTE 0 ;;LOG FLAG
      $FFLG: .BYTE 0 ;;FATAL FLAG
      .EVEN

```

```

APTSIZE=200
APTENV=001
APTPOOL=100
APTCSUP=040

```

.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 R0, -(SP) ;;SAVE R0
        MOV 2(SP), R0 ;;GET TRAP ADDRESS
        TST -(R0) ;;BACKUP BY 2
        MOVB (R0), R0 ;;GET RIGHT BYTE OF TRAP
        ASL R0 ;;POSITION FOR INDEXING
        MOV $TRPAD(R0), R0 ;;INDEX TO TABLE
        RTS R0 ;;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)

```


4849
4850
4851 017032 005015 053103 040513 NAME: .ASCIZ <15><12>/CVKAJB/
4852 017040 041112 000
4853 017044 .EVEN
4854
4855 017044 000020 BUF: .BLKW 20
4856 000001 .END

CROSS REFERENCE TABLE -- USER SYMBOLS

CLBUF	015024	768	864	960	1056	1152	1221	1307	1396	1483	1568	1652	1720	1808
		1897	1985	2072	2168	2270	2358	2454	4384#					
CLRINT	015140	4420	4423#											
CMPN =	076052	477#	2563	2637	2712	2788	2865	2937	3011	3087	3160	3234	3308	3382
		3455	3529	3602	3691									
CMPNPC	012656	3679	3691#											
CR =	000015	377#	4666	4676										
CRLF =	000200	378#	4620	4676										
CVTCON	014260	4192	4194	4197#										
CVTNL =	076053	477#	3771	3845	3918	3991	4064	4137	4216					
CVTPC	014356	4204	4216#											
DDISP =	177570	384#	588											
DISPLA	000542	588#	731*	4475*	4544*									
DISPRE	000174	502#	731											
DSTAD	000656	670#	811	907	1003	1099	1189	1257	1271	1344	1358	1433	1447	1520
		1534	1605	1619	1689	1757	1771	1845	1859	1934	1948	2022	2036	2109
		2123	2213	2227	2307	2321	2394	2408	2500	2514	2595	2669	2744	2820
		2897	2969	3043	3119	3192	3266	3340	3414	3487	3561	3634	3725	3807
		3881	3954	4027	4100	4173	4254	4376						
DSTLN	000654	669#	1185	1253	1272	1340	1359	1429	1448	1516	1535	1601	1620	1685
		1753	1772	1841	1860	1930	1949	2018	2037	2105	2124	2209	2228	2303
		2322	2390	2409	2496	2515	2590	2664	2739	2815	2892	2964	3038	3114
		3187	3261	3335	3409	3482	3556	3629	3720	3802	3876	3949	4022	4095
		4168	4249	4365*	4375									
		383#	587											
DSWR =	177570	472#	703*	704*										
EMTVEC=	000030	4294	4327#											
ENDMSG	014664	783	840	850#										
ENDT1	001552	879	936	946#										
ENDT2	002034	2237	2256#											
ENDT21	006276	2524	2543#											
ENDT24	007274	975	1032	1042#										
ENDT3	002316	1071	1128	1138#										
ENDT4	002600	3738	3751#											
ENDT44	013012	4260	4273#											
ENDT53	014504	465#	4447	4448*	4450*	4453*								
ERRVEC=	000004	679#	819	841	915	937	1011	1033	1107	1129	1162	1230	1317	1406
EXPPSW	000700	1493	1578	1662	1730	1818	1907	1995	2082	2186	2280	2367	2473	2567
		2641	2716	2792	2869	2941	3015	3091	3164	3238	3312	3386	3459	3533
		3606	3697	3775	3849	3922	3995	4068	4141	4222	4392*	4396*		
FILL	000660	671#	806	902	998	1094								
GENR	014760	775	871	967	1063	1155	1224	1310	1399	1486	1571	1655	1723	1811
		1900	1988	2075	2178	2273	2361	2464	2559	2633	2709	2784	2861	2933
		3007	3083	3157	3230	3304	3378	3452	3526	3599	3688	3768	3841	3914
		3987	4060	4133	4213	4371#								
		501	4844	4845	4846	4847								
GNS =	***** U	375#	4618	4676										
HT =	000011	511	4350#											
ILLTRP	014722	2170	2456	3680	4205	4416#								
INTR	015114	470#	701*	702*										
IOTVEC=	000020	376#	4670	4676										
LF =	000012	485#	762#	780	781#	788	789#	793	794#	798	799#	803	804#	808
N =	000010	809#	813	814#	821	822#	828	829#	836	837#	858#	876	877#	884
		885#	889	890#	894	895#	899	900#	904	905#	909	910#	917	918#
		924	925#	932	933#	954#	972	973#	980	981#	985	986#	990	991#
		995	996#	1000	1001#	1005	1006#	1013	1014#	1020	1021#	1028	1029#	1050#

CROSS REFERENCE TABLE -- USER SYMBOLS

1068	1069#	1076	1077#	1081	1082#	1086	1087#	1091	1092#	1096	1097#	1101				
1102#	1109	1110#	1116	1117#	1124	1125#	1146#	1164	1165#	1171	1172#	1175				
1176#	1179	1180#	1183	1184#	1187	1188#	1191	1192#	1196	1197#	1215#	1232				
1233#	1239	1240#	1243	1244#	1247	1248#	1251	1252#	1255	1256#	1259	1260#				
1264	1265#	1275	1276#	1301#	1319	1320#	1326	1327#	1330	1331#	1334	1335#				
1338	1339#	1342	1343#	1346	1347#	1351	1352#	1362	1363#	1390#	1408	1409#				
1415	1416#	1419	1420#	1423	1424#	1427	1428#	1431	1432#	1435	1436#	1440				
1441#	1451	1452#	1477#	1495	1496#	1502	1503#	1506	1507#	1510	1511#	1514				
1515#	1518	1519#	1522	1523#	1527	1528#	1538	1539#	1562#	1580	1581#	1587				
1588#	1591	1592#	1595	1596#	1599	1600#	1603	1604#	1607	1608#	1612	1613#				
1623	1624#	1646#	1664	1665#	1671	1672#	1675	1676#	1679	1680#	1683	1684#				
1687	1688#	1691	1692#	1696	1697#	1714#	1732	1733#	1739	1740#	1743	1744#				
1747	1748#	1751	1752#	1755	1756#	1759	1760#	1764	1765#	1775	1776#	1802#				
1820	1821#	1827	1828#	1831	1832#	1835	1836#	1839	1840#	1843	1844#	1847				
1848#	1852	1853#	1863	1864#	1891#	1909	1910#	1916	1917#	1920	1921#	1924				
1925#	1928	1929#	1932	1933#	1936	1937#	1941	1942#	1952	1953#	1979#	1997				
1998#	2004	2005#	2008	2009#	2012	2013#	2016	2017#	2020	2021#	2024	2025#				
2029	2030#	2040	2041#	2066#	2084	2085#	2091	2092#	2095	2096#	2099	2100#				
2103	2104#	2107	2108#	2111	2112#	2116	2117#	2127	2128#	2153#	2188	2189#				
2195	2196#	2199	2200#	2203	2204#	2207	2208#	2211	2212#	2215	2216#	2220				
2221#	2231	2232#	2264#	2282	2283#	2289	2290#	2293	2294#	2297	2298#	2301				
2302#	2305	2306#	2309	2310#	2314	2315#	2325	2326#	2352#	2369	2370#	2376				
2377#	2380	2381#	2384	2385#	2388	2389#	2392	2393#	2396	2397#	2401	2402#				
2412	2413#	2439#	2475	2476#	2482	2483#	2486	2487#	2490	2491#	2494	2495#				
2498	2499#	2502	2503#	2507	2508#	2518	2519#	2551#	2569	2570#	2576	2577#				
2580	2581#	2584	2585#	2588	2589#	2592	2593#	2597	2598#	2603	2604#	2625#				
2643	2644#	2650	2651#	2654	2655#	2658	2659#	2662	2663#	2666	2667#	2671				
2672#	2677	2678#	2701#	2718	2719#	2725	2726#	2729	2730#	2733	2734#	2737				
2738#	2741	2742#	2746	2747#	2752	2753#	2776#	2794	2795#	2801	2802#	2805				
2806#	2809	2810#	2813	2814#	2817	2818#	2822	2823#	2828	2829#	2853#	2871				
2872#	2878	2879#	2882	2883#	2886	2887#	2890	2891#	2894	2895#	2899	2900#				
2905	2906#	2925#	2943	2944#	2950	2951#	2954	2955#	2958	2959#	2962	2963#				
2966	2967#	2971	2972#	2977	2978#	2999#	3017	3018#	3024	3025#	3028	3029#				
3032	3033#	3036	3037#	3040	3041#	3045	3046#	3051	3052#	3075#	3093	3094#				
3100	3101#	3104	3105#	3108	3109#	3112	3113#	3116	3117#	3121	3122#	3127				
3128#	3149#	3166	3167#	3173	3174#	3177	3178#	3181	3182#	3185	3186#	3189				
3190#	3194	3195#	3200	3201#	3222#	3240	3241#	3247	3248#	3251	3252#	3255				
3256#	3259	3260#	3263	3264#	3268	3269#	3274	3275#	3296#	3314	3315#	3321				
3322#	3325	3326#	3329	3330#	3333	3334#	3337	3338#	3342	3343#	3348	3349#				
3370#	3388	3389#	3395	3396#	3399	3400#	3403	3404#	3407	3408#	3411	3412#				
3416	3417#	3422	3423#	3444#	3461	3462#	3468	3469#	3472	3473#	3476	3477#				
3480	3481#	3484	3485#	3489	3490#	3495	3496#	3518#	3535	3536#	3542	3543#				
3546	3547#	3550	3551#	3554	3555#	3558	3559#	3563	3564#	3569	3570#	3591#				
3608	3609#	3615	3616#	3619	3620#	3623	3624#	3627	3628#	3631	3632#	3636				
3637#	3642	3643#	3664#	3699	3700#	3706	3707#	3710	3711#	3714	3715#	3718				
3719#	3722	3723#	3727	3728#	3733	3734#	3760#	3777	3778#	3784	3785#	3788				
3789#	3792	3793#	3798	3799#	3804	3805#	3809	3810#	3833#	3851	3852#	3858				
3859#	3862	3863#	3866	3867#	3872	3873#	3878	3879#	3883	3884#	3906#	3924				
3925#	3931	3932#	3935	3936#	3939	3940#	3945	3946#	3951	3952#	3956	3957#				
3979#	3997	3998#	4004	4005#	4008	4009#	4012	4013#	4018	4019#	4024	4025#				
4029	4030#	4052#	4070	4071#	4077	4078#	4081	4082#	4085	4086#	4091	4092#				
4097	4098#	4102	4103#	4125#	4143	4144#	4150	4151#	4154	4155#	4158	4159#				
4164	4165#	4170	4171#	4175	4176#	4189#	4224	4225#	4231	4232#	4235	4236#				
4239	4240#	4245	4246#	4251	4252#	4256	4257#									
NAME	017032															
NPREP	014732	740	4851#	762	858	954	1050	1146	1215	1301	1390	1477	1562	1646	1714	1802

SW10	=	002000	414#						
SW11	=	004000	413#						
SW12	=	010000	412#						
SW13	=	020000	411#						
SW14	=	040000	410#						
SW15	=	100000	409#						
SW2	=	000004	432#						
SW3	=	000010	431#						
SW4	=	000020	430#						
SW5	=	000040	429#						
SW6	=	000100	428#						
SW7	=	000200	427#						
SW8	=	000400	426#						
SW9	=	001000	425#						
S1ADR		000646	666#	791	887	983	1079	4362*	4372
S1LN		000644	665#	786	882	978	1074	4361*	4371
S1T1		001550	764	844#					
S1T10		003630	1392	1458#					
S1T11		004044	1479	1545#					
S1T12		004256	1564	1630#					
S1T13		004442	1648	1702#					
S1T14		004650	1716	1782#					
S1T15		005066	1804	1870#					
S1T16		005304	1893	1959#					
S1T17		005522	1981	2047#					
S1T2		002032	860	940#					
S1T20		005736	2068	2134#					
S1T21		006260	2164	2238#					
S1T22		006514	2266	2332#					
S1T23		006730	2354	2419#					
S1T24		007256	2450	2525#					
S1T25		007460	2553	2609#					
S1T26		007640	2627	2683#					
S1T27		010020	2703	2758#					
S1T3		002314	956	1036#					
S1T30		010202	2778	2834#					
S1T31		010364	2855	2911#					
S1T32		010542	2927	2983#					
S1T33		010722	3001	3057#					
S1T34		011104	3077	3133#					
S1T35		011262	3151	3206#					
S1T36		011442	3224	3280#					
S1T37		011622	3298	3354#					
S1T4		002576	1052	1132#					
S1T40		012002	3372	3428#					
S1T41		012160	3446	3501#					
S1T42		012336	3520	3575#					
S1T43		012514	3593	3648#					
S1T44		013002	3675	3739#					
S1T5		002772	1148	1202#					
S1T6		003176	1217	1282#					
S1T7		003412	1303	1369#					
S2ADR		000652	668#	801	897	993	1089	4364*	4374
S2LN		000650	667#	796	892	988	1084	4363*	4373
S2T1		001551	766	846#					
S2T10		003634	1394	1463#					

TST15	004662	1781	1801#																		
TST16	005100	1869	1890#																		
TST17	005316	1958	1978#																		
TST2	001566	857#																			
TST20	005532	2046	2065#																		
TST21	005746	2133	2152#																		
TST22	006310	2154	2160	2263#																	
TST23	006526	2331	2351#																		
TST24	006742	2418	2438#																		
TST25	007306	2440	2446	2550#																	
TST26	007466	2608	2624#																		
TST27	007650	2682	2700#																		
TST3	002050	953#																			
TST30	010030	2757	2775#																		
TST31	010212	2833	2852#																		
TST32	010370	2910	2924#																		
TST33	010550	2982	2998#																		
TST34	010732	3056	3074#																		
TST35	011112	3132	3148#																		
TST36	011270	3205	3221#																		
TST37	011450	3279	3295#																		
TST4	002332	1049#																			
TST40	011630	3353	3369#																		
TST41	012010	3427	3443#																		
TST42	012166	3500	3517#																		
TST43	012344	3574	3590#																		
TST44	012522	3647	3663#																		
TST45	013024	3665	3671	3759#																	
TST46	013176	3813	3832#																		
TST47	013352	3887	3905#																		
TST5	002614	1145#																			
TST50	013526	3960	3978#																		
TST51	013702	4033	4051#																		
TST52	014056	4106	4124#																		
TST53	014222	4179	4188#																		
TST6	002774	1201	1214#																		
TST7	003206	1281	1300#																		
TVECT	000670	675#	749*	2170*	2256*	2456*	2543*	3680*	3751*	4205*	4273*										
TYPE =	104401	740	4294	4514	4551	4555	4623	4737	4844#												
TYPOC =	104402	4554	4845#																		
TYPON =	104404	4847#																			
TYPOS =	104403	4846#																			
TZERO	014702	507	4334#																		
T1	001374	778#	825																		
T1CONT	001402	771	785#																		
T2	001656	874#	921																		
T2CONT	001664	867	881#																		
T21CON	006004	2156	2158	2161#																	
T24CON	007000	2442	2444	2447#																	
T3	002140	970#	1017																		
T3CONT	002146	963	977#																		
T4	002422	1066#	1113																		
T4CONT	002430	1059	1073#																		
T44CON	012560	3667	3669	3672#																	
X =	000000	486#																			
XPSW	015044	773	869	965	1061	1153	1222	1308	1397	1434	1569	1653	1721	1809							

CATCH	342#	4334	4342	4350											
CMPNIN	358#	3664													
CMPNTS	357#	2551	2625	2701	2776	2853	2925	2999	3075	3149	3222	3296	3370	3444	3518
	3591														
CMPREP	344#	2551	2625	2701	2776	2853	2925	2999	3075	3149	3222	3296	3370	3444	3518
	3591	3673													
COMMEN	1#	477#													
CVPREP	358#	3760	3833	3906	3979	4052	4125	4197							
CVTINT	360#	4189													
CVTSRC	355#	3814	3888	3961	4034	4107	4180	4261							
CVTTST	359#	3760	3833	3906	3979	4052	4125								
DECTRP	361#	762	858	954	1050										
EHLT	340#	780	788	793	798	803	808	813	821	828	836	876	884	889	894
	899	904	909	917	924	932	972	980	985	990	995	1000	1005	1013	1020
	1028	1068	1076	1081	1086	1091	1096	1101	1109	1116	1124	1164	1171	1175	1179
	1183	1187	1191	1196	1232	1239	1243	1247	1251	1255	1259	1264	1275	1319	1326
	1330	1334	1338	1342	1346	1351	1362	1408	1415	1419	1423	1427	1431	1435	1440
	1451	1495	1502	1506	1510	1514	1518	1522	1527	1538	1580	1587	1591	1595	1599
	1603	1607	1612	1623	1664	1671	1675	1679	1683	1687	1691	1696	1732	1739	1743
	1747	1751	1755	1759	1764	1775	1820	1827	1831	1835	1839	1843	1847	1852	1863
	1909	1916	1920	1924	1928	1932	1936	1941	1952	1997	2004	2008	2012	2016	2020
	2024	2029	2040	2084	2091	2095	2099	2103	2107	2111	2116	2127	2188	2195	2199
	2203	2207	2211	2215	2220	2231	2282	2289	2293	2297	2301	2305	2309	2314	2325
	2369	2376	2380	2384	2388	2392	2396	2401	2412	2475	2482	2486	2490	2494	2498
	2502	2507	2518	2569	2576	2580	2584	2588	2592	2597	2603	2643	2650	2654	2658
	2662	2666	2671	2677	2718	2725	2729	2733	2737	2741	2746	2752	2794	2801	2805
	2809	2813	2817	2822	2828	2871	2878	2882	2886	2890	2894	2899	2905	2943	2950
	2954	2958	2962	2966	2971	2977	3017	3024	3028	3032	3036	3040	3045	3051	3093
	3100	3104	3108	3112	3116	3121	3127	3166	3173	3177	3181	3185	3189	3194	3200
	3240	3247	3251	3255	3259	3263	3268	3274	3314	3321	3325	3329	3333	3337	3342
	3348	3388	3395	3399	3403	3407	3411	3416	3422	3461	3468	3472	3476	3480	3484
	3489	3495	3535	3542	3546	3550	3554	3558	3563	3569	3608	3615	3619	3623	3627
	3631	3636	3642	3699	3706	3710	3714	3718	3722	3727	3733	3777	3784	3788	3792
	3798	3804	3809	3851	3858	3862	3866	3872	3878	3883	3924	3931	3935	3939	3945
	3951	3956	3997	4004	4008	4012	4018	4024	4029	4070	4077	4081	4085	4091	4097
	4102	4143	4150	4154	4158	4164	4170	4175	4224	4231	4235	4239	4245	4251	4256
ENDCOM	1#	477#													
ENDPAS	338#	4294													
ERR	341#	780	788	793	798	803	808	813	821	828	836	876	884	889	894
	899	904	909	917	924	932	972	980	985	990	995	1000	1005	1013	1020
	1028	1068	1076	1081	1086	1091	1096	1101	1109	1116	1124	1164	1171	1175	1179
	1183	1187	1191	1196	1232	1239	1243	1247	1251	1255	1259	1264	1275	1319	1326
	1330	1334	1338	1342	1346	1351	1362	1408	1415	1419	1423	1427	1431	1435	1440
	1451	1495	1502	1506	1510	1514	1518	1522	1527	1538	1580	1587	1591	1595	1599
	1603	1607	1612	1623	1664	1671	1675	1679	1683	1687	1691	1696	1732	1739	1743
	1747	1751	1755	1759	1764	1775	1820	1827	1831	1835	1839	1843	1847	1852	1863
	1909	1916	1920	1924	1928	1932	1936	1941	1952	1997	2004	2008	2012	2016	2020
	2024	2029	2040	2084	2091	2095	2099	2103	2107	2111	2116	2127	2188	2195	2199
	2203	2207	2211	2215	2220	2231	2282	2289	2293	2297	2301	2305	2309	2314	2325
	2369	2376	2380	2384	2388	2392	2396	2401	2412	2475	2482	2486	2490	2494	2498
	2502	2507	2518	2569	2576	2580	2584	2588	2592	2597	2603	2643	2650	2654	2658
	2662	2666	2671	2677	2718	2725	2729	2733	2737	2741	2746	2752	2794	2801	2805
	2809	2813	2817	2822	2828	2871	2878	2882	2886	2890	2894	2899	2905	2943	2950
	2954	2958	2962	2966	2971	2977	3017	3024	3028	3032	3036	3040	3045	3051	3093
	3100	3104	3108	3112	3116	3121	3127	3166	3173	3177	3181	3185	3189	3194	3200
	3240	3247	3251	3255	3259	3263	3268	3274	3314	3321	3325	3329	3333	3337	3342

ERROR	3348	3388	3395	3399	3403	3407	3411	3416	3422	3461	3468	3472	3476	3480	3484
	3489	3495	3535	3542	3546	3550	3554	3558	3563	3569	3608	3615	3619	3623	3627
	3631	3636	3642	3699	3706	3710	3714	3718	3722	3727	3733	3777	3784	3788	3792
	3798	3804	3809	3851	3858	3862	3866	3872	3878	3883	3924	3931	3935	3939	3945
	3951	3956	3997	4004	4008	4012	4018	4024	4029	4070	4077	4081	4085	4091	4097
	4102	4143	4150	4154	4158	4164	4170	4175	4224	4231	4235	4239	4245	4251	4256
	371#	780	788	793	798	803	808	813	821	828	836	876	884	889	894
	899	904	909	917	924	932	972	980	985	990	995	1000	1005	1013	1020
	1028	1068	1076	1081	1086	1091	1096	1101	1109	1116	1124	1164	1171	1175	1179
	1183	1187	1191	1196	1232	1239	1243	1247	1251	1255	1259	1264	1275	1319	1326
	1330	1334	1338	1342	1346	1351	1362	1408	1415	1419	1423	1427	1431	1435	1440
	1451	1495	1502	1506	1510	1514	1518	1522	1527	1538	1580	1587	1591	1595	1599
	1603	1607	1612	1623	1664	1671	1675	1679	1683	1687	1691	1696	1732	1739	1743
	1747	1751	1755	1759	1764	1775	1820	1827	1831	1835	1839	1843	1847	1852	1863
	1909	1916	1920	1924	1928	1932	1936	1941	1952	1997	2004	2008	2012	2016	2020
	2024	2029	2040	2084	2091	2095	2099	2103	2107	2111	2116	2127	2188	2195	2199
	2203	2207	2211	2215	2220	2231	2282	2289	2293	2297	2301	2305	2309	2314	2325
	2369	2376	2380	2384	2388	2392	2396	2401	2412	2475	2482	2486	2490	2494	2498
	2502	2507	2518	2569	2576	2580	2584	2588	2592	2597	2603	2643	2650	2654	2658
	2662	2666	2671	2677	2718	2725	2729	2733	2737	2741	2746	2752	2794	2801	2805
	2809	2813	2817	2822	2828	2871	2878	2882	2886	2890	2894	2899	2905	2943	2950
	2954	2958	2962	2966	2971	2977	3017	3024	3028	3032	3036	3040	3045	3051	3093
	3100	3104	3108	3112	3116	3121	3127	3166	3173	3177	3181	3185	3189	3194	3200
	3240	3247	3251	3255	3259	3263	3268	3274	3314	3321	3325	3329	3333	3337	3342
	3348	3388	3395	3399	3403	3407	3411	3416	3422	3461	3468	3472	3476	3480	3484
	3489	3495	3535	3542	3546	3550	3554	3558	3563	3569	3608	3615	3619	3623	3627
	3631	3636	3642	3699	3706	3710	3714	3718	3722	3727	3733	3777	3784	3788	3792
	3798	3804	3809	3851	3858	3862	3866	3872	3878	3883	3924	3931	3935	3939	3945
	3951	3956	3997	4004	4008	4012	4018	4024	4029	4070	4077	4081	4085	4091	4097
	4102	4143	4150	4154	4158	4164	4170	4175	4224	4231	4235	4239	4245	4251	4256
	4335	4343	4351	4408											
ESCAPE	1#	477#													
GETPRI	1#	477#	4310												
GETSWR	1#	477#													
INT	348#	4416													
MANS	351#	1290	1377	1466	1551	1636	1791	1879	1968	2055	2142	2248	2341	2428	2535
MNPREP	343#	762	858	954	1050	1146	1215	1301	1390	1477	1562	1646	1714	1802	1891
MSRC	1979	2066	2162	2264	2352	2448									
	350#	844	940	1036	1132	1202	1282	1369	1458	1545	1630	1702	1782	1870	1959
	2047	2134	2238	2332	2419	2525	2609	2683	2758	2834	2911	2983	3057	3133	3206
	3280	3354	3428	3501	3575	3648	3739								
MULT	1#	477#													
NEWTST	1#	477#	758	854	950	1046	1142	1211	1297	1386	1473	1558	1642	1710	1798
	1887	1975	2062	2149	2260	2348	2435	2547	2621	2697	2772	2849	2921	2995	3071
	3145	3218	3292	3366	3440	3514	3587	3660	3756	3829	3902	3975	4048	4121	4185
NUMERI	354#	1146	1215	1301	1390	1477	1562	1646	1714	1802	1891	1979	2066	2264	2352
NUMINT	355#	2153	2439												
POP	1#	336#	477#	4505	4506	4301	4802								
PUSH	1#	336#	477#	4485	4491	4762	4764	4785							
REPORT	1#	477#													
RESETN	342#	762	858	954	1050	1146	1215	1301	1390	1477	1562	1646	1714	1802	1891
	1979	2066	2153	2264	2352	2439	2551	2625	2701	2776	2853	2925	2999	3075	3149
	3222	3296	3370	3444	3518	3591	3664	3760	3833	3906	3979	4052	4125	4189	
SCOPE	372#	761	857	953	1049	1145	1214	1300	1389	1476	1561	1645	1713	1801	1890
	1978	2065	2152	2263	2351	2438	2550	2624	2700	2775	2852	2924	2998	3074	3148
	3221	3295	3369	3443	3517	3590	3663	3759	3832	3905	3978	4051	4124	4188	4284

.SEOP	1#	336#	4275
.SERRO	1#	336#	4528
.SERRT	1#		
.SMULT	1#		
.SPOWE	1#	336#	4479
.SRAND	1#		
.SRDDE	1#		
.SRDOC	1#		
.SREAD	1#		
.SR2AZ	1#		
.SSAVE	1#		
.SSB2D	1#		
.SSB2O	1#		
.SSCOP	1#	336#	4428
.SSIZE	1#		
.SSUPR	1#		
.STRAP	1#	336#	4813
.STYPB	1#		
.STYPD	1#		
.STYPE	1#	336#	4580
.STYPO	1#	336#	4677
.S4OCA	1#		
.1170	1#		

. ABS. 017104 000

ERRORS DETECTED: 0

CVKAJB.BIN, CVKAJB.LST/CRF/SOL/NL:TOC=SYSMAC.SML, CVKAJB.P11
RUN-TIME: 19 21 1 SECONDS
RUN-TIME RATIO: 115/42=2.7
CORE USED: 39K (78 PAGES)