

LSI-11

DIS MOV/STRG INST
CVKAIBO

AH-8216B MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 77-82
MADE IN USA



A grid of approximately 15 columns and 15 rows of small, illegible text or data points, possibly representing a technical specification or a data table. The text is too small to be read accurately.



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

.REM %

IDENTIFICATION

PRODUCT CODE: AC-8214B-MC
PRODUCT NAME: CVKAIB0 DIS MOV/STRG INST 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

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

1.0 GENERAL INFORMATION

1.1 ABSTRACT

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

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

CVKAI* [THIS DIAGNOSTIC]
CVKAJ* DIS DECIMAL INSTRUCTION TEST
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 000000 TO 17310

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

107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

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.

152
153
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

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*

186
187
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
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241

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
 @ <--ODT ENTERED

WHERE 'XXXXXX' IS AN OCTAL VALUE

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.

242 4.3 ERROR
 243 -----
 244 THIS ROUTINE CALL IS PLACED WHEREEVER AN ERROR REPORT IS DESIRED. THE
 245 LOWER BYTE OF THIS CALL IS USED AS THE ERROR NUMBER. THIS ROUTINE
 246 REPORTS ERRORS TO APT, TYPES ERRORS TO THE CONSOLE USING THE
 247 "STYPE" & "TYPOCT" ROUTINES, AND HANDLES ERROR RESPONSES VIA SWR SETTINGS.
 248
 249
 250 4.4 \$POWER
 251 -----
 252 THIS ROUTINE SAVES ALL GENERAL REGISTERS DURING POWER-DOWN AND
 253 RESTORES THEM AT POWER-UP. IF A POWER FAILURE OCCURS "POWER" IS
 254 PRINTED AT THE CONSOLE AFTER POWER IS RESTORED AND THE PROGRAM
 255 IS RESTARTED AT TEST# 1.
 256
 257
 258
 259
 260 4.5 NPREP
 261 -----
 262 THIS ROUTINE IS USED TO STORE A COPY OF THE INSTRUCTION TEST
 263 ARGUMENTS TO BE STORED IN R0-->R5.
 264
 265
 266 4.6 GENR
 267 -----
 268 THIS ROUTINE IS USED TO TRANSFER INSTRUCTION TEST ARGUMENTS
 269 TO THE GENERAL REGISTERS AND TO COPY THE STACK POINTER BEFORE THE
 270 TEST INSTRUCTION EXECUTION.
 271
 272
 273 4.7 XPSW
 274 -----
 275 THIS ROUTINE IS USED TO STORED THE EXPECTED PSW OF THE INSTRUCTION
 276 TEST AND TO SET THE T-BIT IN THE EXPECTED PSW ON PASSES USING
 277 THE TRACE TRAP.
 278
 279
 280 4.8 INTR
 281 -----
 282 THIS ROUTINE IS USED TO DETECT WHEN THE TEST INSTRUCTION HAS
 283 BEEN INTERRUPTED AND TO CONTINUE THE INTERRUPT STREAM UNTIL THE
 284 INSTRUCTION IS INTERRUPTED.
 285
 286 4.9 CKCC
 287 -----
 288 THIS ROUTINE IS USED TO CHECK THE PSW'S CONTENT, AFTER THE
 289 OPERATION, WITH THE EXPECTED PSW'S CONTENT (EXPPSW).
 290 **NOTE: CONDITION CODES GENERATED IN THIS ROUTINE WILL BE USED
 291 IN THE MAIN PROGRAM. NO CODES SHOULD BE ADDED IN THIS
 292 ROUTINE THAT WOULD AFFECT THE CONDITIONS CODES.
 293
 294
 295 4.10 SKPINT
 296 -----
 297 THIS ROUTINE IS USED TO CKECK IF WE SHOULD SKIP THE CURRENT TEST.

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
344
345
346
347
348
349
350
351
352
353

LOC. -TEMP- IS USED AS A FLAG THAT PASSES BACK TO THE MAIN PROGRAM.
TEMP = 0 ==> CONTINUE WITH THE CURRENT TEST
TEMP = 1 ==> GO TO DO THE NEXT TEST

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 HANDLED IT.

%

.ENABLE ABS

.LIST ME
.NLIST MC,MD,CND

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
400
401
402
403
404
405
406
407
408
409

.SBTTL BASIC DEFINITIONS

;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

001100

STACK= 1100

.EQUIV EMT,ERROR ::BASIC DEFINITION OF ERROR CALL

.EQUIV IOT,SCOPE ::BASIC DEFINITION OF SCOPE CALL

;*MISCELLANEOUS DEFINITIONS

000011

HT= 11 ::CODE FOR HORIZONTAL TAB

000012

LF= 12 ::CODE FOR LINE FEED

000015

CR= 15 ::CODE FOR CARRIAGE RETURN

000200

CRLF= 200 ::CODE FOR CARRIAGE RETURN-LINE FEED

177776

PS= 177776 ::PROCESSOR STATUS WORD

.EQUIV PS,PSW

177774

STKLMT= 177774 ::STACK LIMIT REGISTER

177772

PIRQ= 177772 ::PROGRAM INTERRUPT REQUEST REGISTER

177570

DSWR= 177570 ::HARDWARE SWITCH REGISTER

177570

DDISP= 177570 ::HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS

000000

R0= X0 ::GENERAL REGISTER

000001

R1= X1 ::GENERAL REGISTER

000002

R2= X2 ::GENERAL REGISTER

000003

R3= X3 ::GENERAL REGISTER

000004

R4= X4 ::GENERAL REGISTER

000005

R5= X5 ::GENERAL REGISTER

410	000006	R6=	%6	::GENERAL REGISTER
411	000007	R7=	%7	::GENERAL REGISTER
412	000006	SP=	%6	::STACK POINTER
413	000007	PC=	%7	::PROGRAM COUNTER

414
415
416
417
418
419
420
421
422
423
424

425 : *PRIORITY LEVEL DEFINITIONS

416	000000	PR0=	0	::PRIORITY LEVEL 0
417	000010	PR1=	40	::PRIORITY LEVEL 1
418	000100	PR2=	100	::PRIORITY LEVEL 2
419	000140	PR3=	140	::PRIORITY LEVEL 3
420	000200	PR4=	200	::PRIORITY LEVEL 4
421	000240	PR5=	240	::PRIORITY LEVEL 5
422	000300	PR6=	300	::PRIORITY LEVEL 6
423	000340	PR7=	340	::PRIORITY LEVEL 7

425 : *"SWITCH REGISTER" SWITCH DEFINITIONS

426	100000	SW15=	100000
427	040000	SW14=	40000
428	020000	SW13=	20000
429	010000	SW12=	10000
430	004000	SW11=	4000
431	002000	SW10=	2000
432	001000	SW09=	1000
433	000400	SW08=	400
434	000200	SW07=	200
435	000100	SW06=	100
436	000040	SW05=	40
437	000020	SW04=	20
438	000010	SW03=	10
439	000004	SW02=	4
440	000002	SW01=	2
441	000001	SW00=	1
442		.EQUIV	SW09,SW9
443		.EQUIV	SW08,SW8
444		.EQUIV	SW07,SW7
445		.EQUIV	SW06,SW6
446		.EQUIV	SW05,SW5
447		.EQUIV	SW04,SW4
448		.EQUIV	SW03,SW3
449		.EQUIV	SW02,SW2
450		.EQUIV	SW01,SW1
451		.EQUIV	SW00,SW0

452
453 : *DATA BIT DEFINITIONS (BIT00 TO BIT15)

454	100000	BIT15=	100000
455	040000	BIT14=	40000
456	020000	BIT13=	20000
457	010000	BIT12=	10000
458	004000	BIT11=	4000
459	002000	BIT10=	2000
460	001000	BIT09=	1000
461	000400	BIT08=	400
462	000200	BIT07=	200
463	000100	BIT06=	100
464	000040	BIT05=	40
465	000020	BIT04=	20

```

466      000010      BIT03= 10
467      000004      BIT02= 4
468      000002      BIT01= 2
469      000001      BIT00= 1
470      .EQUIV      BIT09,BIT9
471      .EQUIV      BIT08,BIT8
472      .EQUIV      BIT07,BIT7
473      .EQUIV      BIT06,BIT6
474      .EQUIV      BIT05,BIT5
475      .EQUIV      BIT04,BIT4
476      .EQUIV      BIT03,BIT3
477      .EQUIV      BIT02,BIT2
478      .EQUIV      BIT01,BIT1
479      .EQUIV      BIT00,BIT0

```

```

480
481      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
482      ERRVEC= 4      ;;TIME OUT AND OTHER ERRORS
483      RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
484      TBITVEC=14     ;;"T" BIT
485      TRTVEC= 14     ;;TRACE TRAP
486      BPTVEC= 14     ;;BREAKPOINT TRAP (BPT)
487      IOTVEC= 20     ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
488      PWRVEC= 24     ;;POWER FAIL
489      EMTVEC= 30     ;;EMULATOR TRAP (EMT) **ERROR**
490      TRAPVEC=34     ;;"TRAP" TRAP
491      TKVEC= 60      ;;TTY KEYBOARD VECTOR
492      TPVEC= 64      ;;TTY PRINTER VECTOR
493      PIRQVEC=240    ;;PROGRAM INTERRUPT REQUEST VECTOR

```

```

494      APTSIZE=      200
495      APTENV= 001
496      APTSPool=    100
497      APTCSUP=     040
498      $SWR= 171400
499      $SWRMK= 300
500      TBIT= 20
501      $TN= 1
502      N= 1
503      NXM= 177777
504      ABASE= 177560
505      AVECT1= 60
506      .=0

```

507
508 .SBTTL TRAP CATCHER

```

509      .=0
510      000000
511      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
512      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
513      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
514      .=174
515      000174 000000      DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
516      000176 000000      SWREG: .WORD 0      ;;SOFTWARE SWITCH REGISTER
517
518      =0
519      000000 014144      ;ZERO      ;SET LOCATIONS 0,4,6 TO ERROR REPORT
520      000002 000340
521      000004 014154      TIMTRP

```



```

522 000006 000340          340
523 000010 014164          ILLTRP
524 000012 000340          340
525
526
527
528                000100          .=100
529 000100 000102          .WORD 102          ;HANDLE EVENT LINE INTERRUPTS
530 000102 000002          .WORD 2
531
532
533                000200          .=200
534 000200 000167 000512    JMP START          ;STARTING ADDRESS OF PROGRAM
535
536                000400          .=400
537                .SBTTL ACT11 HOOKS
538
539                ;*****
540                ;HOOKS REQUIRED BY ACT11
541                000400          $SVPC=.          ;SAVE PC
542                000046          .=46
543 000046 014050          $ENDAD          ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
544                000052          .=52
545 000052 000000          .WORD 0          ;:2)SET LOC.52 TO ZERO
546                000400          .= $SVPC          ;: RESTORE PC
547                .SBTTL APT PARAMETER BLOCK
548
549                ;*****
550                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
551                ;*****
552                000400          .$X=.          ;:SAVE CURRENT LOCATION
553                000024          .=24          ;:SET POWER FAIL TO POINT TO START OF PROGRAM
554 000024 000200          200          ;:FOR APT START UP
555                000044          .=44          ;:POINT TO APT INDIRECT ADDRESS PNTR.
556 000044 000400          $APTHDR        ;:POINT TO APT HEADER BLOCK
557                000400          .=.$X          ;:RESET LOCATION COUNTER
558                ;*****
559                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
560                ;INTERFACE SPEC.
561
562                $APTHD:
563 000400 000000          $HIBTS: .WORD 0          ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
564 000402 000566          $MBADR: .WORD $MAIL      ;;ADDRESS OF APT MAILBOX (BITS 0-15)
565 000404 000030          $STMT: .WORD 30         ;;RUN TIM OF LONGEST TEST
566 000406 000040          $PASTM: .WORD 40        ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
567 000410 000000          $UNITM: .WORD          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
568 000412 000027          .WORD SETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
569
570

```

571
572
573
574
575
576
577 000500
578 000500
579 000500 000000
580 000502 000
581 000503 000
582 000504 000000
583 000506 000000
584 000510 000000
585 000512 000000
586 000514 000
587 000515 001
588 000516 000000
589 000520 000000
590 000522 000000
591 000524 000000
592 000526 000000
593 000530 000000
594 000532 000000
595 000534 000
596 000535 000
597 000536 000000
598 000540 177570
599 000542 177570
600 000544 177560
601 000546 177562
602 000550 177564
603 000552 177566
604 000554 000
605 000555 002
606 000556 012
607 000557 000
608 000560 000000
609 000562 077
610 000563 015
611 000564 000012
612
613
614
615
616
617 000566
618 000566 000000
619 000570 000000
620 000572 000000
621 000574 000000
622 000576 000000
623 000600 000000
624 000602 000000
625 000604 000000
626 000606

.SBTTL COMMON TAGS

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

.=500

\$CMTAG: .WORD 0
\$STNM: .BYTE 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: .ASCIZ <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

627	000606	000	\$ENV:	.BYTE	AENV	::ENVIRONMENT BYTE
628	000607	000	\$ENVM:	.BYTE	AENVM	::ENVIRONMENT MODE BITS
629	000610	000000	\$SWREG:	.WORD	ASWREG	::APT SWITCH REGISTER
630	000612	000000	\$USWR:	.WORD	AUSWR	::USER SWITCHES
631	000614	000000	\$CPUOP:	.WORD	ACPUOP	::CPU TYPE,OPTIONS
632			*			BITS 15-11=CPU TYPE
633			*			11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
634			*			11/70=06,PDQ=07,Q=10
635			*			BIT 10=REAL TIME CLOCK
636			*			BIT 9=FLOATING POINT PROCESSOR
637			*			BIT 8=MEMORY MANAGEMENT
638	000616	000	\$MAMS1:	.BYTE	AMAMS1	::HIGH ADDRESS,M.S. BYTE
639	000617	000	\$MTYP1:	.BYTE	AMTYP1	::MEM. TYPE,BLK#1
640			*			MEM.TYPE BYTE -- (HIGH BYTE)
641			*			900 NSEC CORE=001
642			*			300 NSEC BIPOLAR=002
643			*			500 NSEC MOS=003
644	000620	000000	\$MADR1:	.WORD	AMADR1	::HIGH ADDRESS,BLK#1
645			*			MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
646	000622	000	\$MAMS2:	.BYTE	AMAMS2	::HIGH ADDRESS,M.S. BYTE
647	000623	000	\$MTYP2:	.BYTE	AMTYP2	::MEM. TYPE,BLK#2
648	000624	000000	\$MADR2:	.WORD	AMADR2	::MEM.LAST ADDRESS,BLK#2
649	000626	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS,M.S.BYTE
650	000627	000	\$MTYP3:	.BYTE	AMTYP3	::MEM. TYPE,BLK#3
651	000630	000000	\$MADR3:	.WORD	AMADR3	::MEM.LAST ADDRESS,BLK#3
652	000632	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS,M.S.BYTE
653	000633	000	\$MTYP4:	.BYTE	AMTYP4	::MEM. TYPE,BLK#4
654	000634	000000	\$MADR4:	.WORD	AMADR4	::MEM.LAST ADDRESS,BLK#4
655	000636	000060	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1,BUS PRIORITY#1
656	000640	000000	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2BUS PRIORITY#2
657	000642	177560	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
658	000644		\$ETEND:			
659			.MEXIT			

```

660 .SBTTL ERROR POINTER TABLE
661
662 ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
663 ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
664 ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
665 ;*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
666 ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
667
668 ;* EM ;:POINTS TO THE ERROR MESSAGE
669 ;* DH ;:POINTS TO THE DATA HEADER
670 ;* DT ;:POINTS TO THE DATA
671 ;* DF ;:POINTS TO THE DATA FORMAT
672
673
674 000644 $ERRTB:
675
676 000644 000000 SRCLN: .WORD 0 ;:SOURCE LENGTH
677 000646 000000 SRCAD: .WORD ;:SOURCE ADDRESS
678 000650 000000 DSTLN: .WORD ;:DESTINATION LENGTH
679 000652 000000 DSTAD: .WORD ;:DESTINATION ADDRESS
680 000654 000000 FILL: .WORD ;:FILL CHARACTER
681 000656 177777 TABLE: .WORD 177777 ;:TRANSLATION TABLE ADDRESS
682 000660 177564 TCSR: 177564 ;:TCSR ADDRESS OF SLU USED FOR INTERRUPTS
683 000662 177566 TBUF: 177566 ;:TBUF ADDRESS
684 000664 000064 TVECT: 64 ;:TRANSMIT INTERRUPT VECTOR
685 000666 000066 TPSW: 66 ;: AND PSW LOCATION
686 000670 000000 PCI: 0 ;:ADDRESS OF TEST INSTRUCTION TO INTERRUPT
687 000672 000000 CCODES: 0 ;:CONDITION CODES AFTER TEST INSTRUCTION EXECUTION
688 000674 000000 EXPPSW: 0 ;:EXPECTED CONDITION CODES
689 000676 000000 SAVR6: 0 ;:STACK POINTER VALUE BEFORE TEST INSTRUCTION EXECUTION
690 000700 000000 BADR6: 0 ;:BAD STACK POINTER VALUE
691 000702 000000 OLDPC: 0 ;:PC WHERE UNEXPECTED TRAP OR INTERRUPT OCCURRED
692 000704 000000 TEMP: 0
693 000706 000000 TEMP1: 0
694 000710 000000 TEMP2: 0
695
696 000712 177777 ONES: .WORD -1
697 000714 000377 ONEBYT: 377
698
699
700 000716 005067 177646 START: CLR $FATAL ;:CLEAR ERROR NO.
701 000722 005067 177640 CLR $MSGTYP ;:CLEAR MESSAGE TYPE
702 000726 005067 177640 CLR $TESTN ;:CLEAR TEST NO.
703
704 .SBTTL INITIALIZE THE COMMON TAGS
705 000732 012706 000500 MOV #$SCMTAG,R6 ;:CLEAR THE COMMON TAGS ($SCMTAG) AREA
706 000736 005026 CLR (R6)+ ;:FIRST LOCATION TO BE CLEARED
707 000740 022706 000540 CMP #SWR,R6 ;DONE? ;:CLEAR MEMORY LOCATION
708 000744 001374 BNE -6 ;:LOOP BACK IF NO
709 000746 012706 000500 MOV #500,SP ;:SETUP THE STACK POINTER
710 ;:INITIALIZE A FEW VECTORS
711 000752 012737 014542 000020 MOV #$SCOPE,#IOTVEC ;:IOT VECTOR FOR SCOPE ROUTINE
712 000760 012737 000340 000022 MOV #340,#IOTVEC+2 ;:LEVEL 7
713 000766 012737 015140 000030 MOV #$ERROR,#EMTVEC ;:EMT VECTOR FOR ERROR ROUTINE
714 000774 012737 000340 000032 MOV #340,#EMTVEC+2 ;:LEVEL 7
715 001002 012737 016236 000034 MOV #$STRAP,#TRAPVEC ;:TRAP VECTOR FOR TRAP CALLS

```



```

716 001010 012737 000340 000036      MOV      #340,@#TRAPVEC+2:LEVEL 7
717 001016 012737 014740 000024      MOV      #SPWRDN,@#PWRVEC :POWER FAILURE VECTOR
718 001024 012737 000340 000026      MOV      #340,@#PWRVEC+2 :LEVEL 7
719 001032 016767 012754 012744      MOV      $ENDCT,$EOPCT   :SETUP END-OF-PROGRAM COUNTER
720 001040 005067 177514                CLR      $ESCAPE        :CLEAR THE ESCAPE ON ERROR ADDRESS
721 001044 112767 000001 177443      MOVVB   #1,$SERMAX      :ALLOW ONE ERROR PER TEST
722                                :INITIALIZE THE 'T-BIT' TRAP VECTOR. THEN LOAD LOCATION,
723                                :THE 'END-OF-PASS' ($EOP) ROUTINE, WITH A 'RTI' OR 'RTT'
724 001052 012737 014114 000014      MOV      #SRTN,@#TBITVEC :SET 'T' BIT VECTOR TO SRTN
725 001060 012737 000340 000016      MOV      #340,@#TBITVEC+2:LEVEL 7
726 001066 012767 000002 013020      MOV      #RTI,$SRTN     :SET SRTN TO A RTI
727 001074 013767 000010 177602      MOV      @#RESVEC,TEMP  :SAVE ILLEGAL INSTRUCTION TRAP VECTOR
728 001102 012737 001130 000010      MOV      #2,$@#RESVEC  :TRY TO DO A RTT
729 001110 005046                CLR      -(SP)         :DUMMY PS
730 001112 012746 001120                MOV      #1$,-(SP)    :AND PC
731 001116 000006                RTT                    :TRY THE RTT
732 001120 012767 000006 012766 1$:    MOV      #RTT,$SRTN   :RTT IS LEGAL--SET SRTN TO A RTT
733 001126 000402                BR      3$
734 001130 062706 000010                ADD     #10,SP        :RTT ILLEGAL--CLEAN OFF THE STACK
735 001134 016737 177544 000010 2$:    MOV      TEMP,@#RESVEC :RESTORE ILLEGAL INSTRUCTION TRAP VECTOR
736 001142 005067 012754                CLR     $TBIT        :CLEAR 'T' BIT SWITCH
737 001146 012767 001146 177332      MOV      #.,$SLPADR    :INITIALIZE THE LOOP ADDRESS FOR SCOPE
738 001154 012767 001154 177326      MOV      #.,$SLPERR    :SETUP THE ERROR LOOP ADDRESS
739                                :SETUP FOR A SOFTWARE SWITCH REGISTER.
740 001162 012767 000176 177350      MOV      #SWREG,$SWR   :POINT TO SOFTWARE SWR
741 001170 012767 000174 177344      MOV      #DISPREG,$DISPLAY
742
743 001176 005067 177372                CLR     $PASS        :CLEAR PASS COUNT
744 001202 132767 000200 177377      BITB   #APTSIZE,$ENVM  :TEST USER SIZE UNDER APT
745 001210 001403                BEQ    4$
746 001212 012767 000610 177320      MOV     #SSWREG,$SWR  :NO,USE APT SWITCH REGISTER
747 001220                4$:
748 001220 026737 012624 000042      CMP     $ENDAD,@#42   :UNDER ACT11 AUTO-ACCEPT?
749 001226 001424                BEQ    PATGEN        :BR IF YES, & SKIP PROGRAM ID TYPEOUT
750 001230 104401 016276                TYPE, NAME
751                                :SET UP ADDRESSES OF SLU TO USE FOR INTERRUPTABILITY TES
752 001234 013700 000642                MOV     @#$BASE,$R0   :GET ADDRESS OF THE SLU
753 001240 062700 000004                ADD     #4,$R0        :ADJUST TO TCSR ADDRESS
754 001244 010037 000660                MOV     $R0,@#TCSR   :STORE TCSR ADDRESS
755 001250 005720                TST    ($R0)+         :ADJUST TO TBUF ADDRESS
756 001252 010037 000662                MOV     $R0,@#TBUF   :STORE TBUF ADDRESS
757 001256 013700 000636                MOV     @#$VECT1,$R0 :GET SLU INTERRUPT VECTOR
758 001262 062700 000004                ADD     #4,$R0        :ADJUST TO TRANSMIT INTERRUPT VECTOR
759 001266 010037 000664                MOV     $R0,@#TVECT  :STORE TRANSMIT INTERRUPT VECTOR
760 001272 005720                TST    ($R0)+         :ADJUST TO TRANSMIT INTERRUPT PSW
761 001274 010037 000666                MOV     $R0,@#TPSW   :STORE TRANSMIT INTERRUPT PSW LOCATION
762 001300 012701 016310                MOV     #BUF1,$R1    :GENERATE TEST PATTERN IN BUF1
763 001304 012700 000001                MOV     #1,$R0       :THE PATTERN IA A BINARY COUNT ALTERNATED
764 001310 110021                1$:    MOVVB  $R0,($R1)+     :WITH ITS COMPLEMENT
765 001312 105100                COMB   $R0           :001,376,002,375,003,374,.....,177,200,200,177
766 001314 110021                MOVVB  $R0,($R1)+
767 001316 105100                COMB   $R0
768 001320 105200                INCB  $R0           :INCREMENT PATTERN
769 001322 020027 000200                CMP    $R0,#200     :IS PATTERN FINISHED?
770 001326 003770                BLE   1$           :BR IF NOT
771

```

772 001330 106427 000200

BEGIN: MTPS #200 ;SET PRIORITY TO 7

773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827

001334 000004
001336 004567 012706
001342 000000
001344 177777
001346 000005
001350 016711
001352 000377
001354 004767 012716
001360 004567 012760
001364 000211
001366 004767 012612
001372 000251
001374 000266
001376 076030
001400 004767 012764
001404 001401
001406 104001
001410
001410 023706 000676
001414 001403
001416 010637 000700
001422 104002
001424
001424 005700
001426 001401
001430 104003
001432
001432 005701
001434 001401
001436 104004
001440 005702
001442 001401
001444 104005
001446 005703

:TEST 1 TEST 'MOVC' INSTRUCTION WITH ZERO SOURCE LENGTH
:SUCCESSFUL COMPLETION OF THIS TEST RESULTS IN
: 'FILL' ('377') BYTES WRITTEN THROUGHOUT THE DESTINATION
:FIELD AND CONDITION CODES-->N,C=1 & Z,V=0

TST1: SCOPE ;SET UP INSTRUCTION ARGUMENTS
JSR R5,PREP ;SOURCE LENGTH
0 ;SOURCE ADDRESS
NXM ;DESTINATION LENGTH
5 ;DESTINATION ADDRESS
BUF2+1 ;FILL CHARACTER
377 ;CLEAR DESTINATION
JSR PC,CLDST ;STORE EXPECTED PSW VALUE
JSR R5,XPSW
.WORD 211 ;SET UP GENERAL REGISTERS
JSR PC,GENR ;CLEAR CONDITION CODES N & C
+CLN!CLC ;SET CONDITION CODES Z & V
+SEZ!SEV ;EXECUTE 'MOVE CHARACTER' INSTRUCTION
MOVC ;CHECK RESULTS
JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES
BEQ 64\$
ERROR 1 ;*****TEST 1 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT 'EXPPSW'
;ACTUAL PSW IS STORED AT 'CCODES'
64\$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65\$;BR IF OK
MOV SP,@#BADR6 ;STORE BAD SP VALUE
ERROR 2 ;*****TEST 1 - ERROR 2*****
;STACK POINTER NOT RESTORED BY INSTRUCTION
;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
;ERRONEOUS VALUE IS AT 'BADR6'
65\$: TST R0 ;CHECK R0 FOR ZERO
BEQ 1\$;BR, IF ZERO
ERROR 3 ;*****TEST 1 - ERROR 3*****
;R0 SHOULD BE ZERO
1\$: TST R1 ;CHECK STATE OF OTHER GENERAL REGISTERS
BEQ 66\$;TEST R1
ERROR 4 ;BR, IF ZERO
;*****TEST 1 - ERROR 4*****
;R1 SHOULD BE ZERO
66\$: TST R2 ;TEST R2
BEQ 67\$;BR IF ZERO
ERROR 5 ;*****TEST 1 - ERROR 5*****
;R2 SHOULD BE ZERO
67\$: TST R3 ;TEST R3

TEST 'MOVC' INSTRUCTION WITH ZERO SOURCE LENGTH

```

828 001450 001401      BEQ    68$      ;BR, IF ZERO
829 001452 104006      ERROR   6        ;*****TEST 1 - ERROR 6*****
830                                     ;R3 SHOULD BE ZERO
831 001454                                     68$:
832 001454 026704 177174  CMP    FILL,R4    ;CHECK R4 UNCHANGED
833 001460 001401      BEQ    69$      ;BR IF OK
834 001462 104007      ERROR   7        ;*****TEST 1 - ERROR 7*****
835                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
836 001464 026705 177166 69$:  CMP    TABLE,R5 ;CHECK R5 UNCHANGED
837 001470 001401      BEQ    70$      ;BR IF OK
838 001472 104010      ERROR   10       ;*****TEST 1 - ERROR 10*****
839                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
840 001474                                     70$:
841                                     ;VERIFY DESTINATION CONTENTS
842 001474 012700 016710  MOV    #BUF2,R0   ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
843 001500 105720      TSTB   (R0)+      ;TEST CONTENTS OF BOUNDARY
844 001502 001401      BEQ    71$      ;BR, IF STILL ZERO
845 001504 104011      ERROR   11       ;*****TEST 1 - ERROR 11*****
846                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
847                                     ; SHOULD STILL EQUAL ZERO
848 001506                                     71$:
849 001506 016701 177136  MOV    DSTLN,R1   ;STORE TRANSFER BYTE COUNT IN R1
850 001512 122067 177136  T1E12:  CMPB   (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
851 001516 001401      BEQ    +4        ;BR IF OK
852 001520 104012      ERROR   12       ;*****TEST 1 - ERROR 12*****
853                                     ;COMPARE ERROR IN DESTINATION
854                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
855
856 001522 005301      DEC    R1        ;DECREMENT BYTE COUNT
857 001524 001372      BNE    T1E12     ;BR, IF NOT FINISHED CHECKING
858 001526 105720      TSTB   (R0)+      ;TEST CONTENTS OF DEST. UPPER BOUNDARY
859 001530 001401      BEQ    ENDT1    ;BR, IF STILL ZERO
860 001532 104013      ERROR   13       ;*****TEST 1 - ERROR 13*****
861                                     ;UPPER BOUNDARY OF DEST. CHANGED
862                                     ; SHOULD STILL EQUAL ZERO
863 001534      ENDT1:
864
865
866
867
868
869
870
871
872
873
874
875 001534 000004      TST2:  SCOPE
876 001536 012767 000712 177102  MOV    #ONES,SRCAD ;SET SOURCE ADDRESS
877 001544 004567 012500      JSR    R5,PREP    ;SET UP INSTRUCTION ARGUMENTS
878 001550 000001      1        ;SOURCE LENGTH
879 001552 177777      NXM      ;SOURCE ADDRESS
880 001554 000000      0        ;DESTINATION LENGTH
881 001556 016711      BUF2+1  ;DESTINATION ADDRESS
882 001560 000377      377     ;FILL CHARACTER
883                                     ;CLEAR DESTINATION

```

```

:*****
:*TEST 2      TEST 'MOVC' INSTRUCTION WITH ZERO DESTINATION LENGTH
:*****
:*SUCCESSFUL COMPLETION OF THIS TEST RESULTS IN
:*NO CHARACTERS TRANSFERED TO THE DESTINATION
:*AREA AND ALL CONDITION CODES CLEAR
:*****
:*****

```

884	001562	012700	016710	MOV	#BUF2,R0	:POINT R0 TO LOWER BYTE BOUNDARY
885	001566	005020		CLR	(R0)+	:CLEAR BOUNDARY & DEST. BYTES
886	001570	005020		CLR	(R0)+	:CLEAR UPPER BYTE BOUNDARY
887	001572	004567	012546	JSR	R5,XPSW	:STORE EXPECTED PSW VALUE
888	001576	000200		.WORD	200	
889	001600	004767	012400	JSR	PC,GENR	:SET UP GENERAL REGISTERS
890	001604	000277		SCC		:SET ALL CONDITION CODES
891						:EXECUTE 'MOVE CHARACTER' INSTRUCTION
892	001606	076030		MOVC		
893						:CHECK RESULTS
894	001610	004767	012554	JSR	PC,CKCC	:CHECK PSW, GENERATE CONDITION CODES
895	001614	001401		BEQ	64\$	
896	001616	104001		ERROR	1	:*****TEST 2 - ERROR 1*****
897						:PSW ERROR
898						:EXPECTED PSW IS STORED AT 'EXPPSW'
899						:ACTUAL PSW IS STORED AT 'CCODES'
900	001620					64\$:
901	001620	023706	000676	CMP	@#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
902	001624	001403		BEQ	65\$:BR IF OK
903	001626	010637	000700	MOV	SP,@#BADR6	:STORE BAD SP VALUE
904	001632	104002		ERROR	2	:*****TEST 2 - ERROR 2*****
905						:STACK POINTER NOT RESTORED BY INSTRUCTION
906						:EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
907						:ERRONEOUS VALUE IS AT 'BADR6'
908	001634					65\$:
909	001634	020067	177004	CMP	R0,SRCLN	:CHECK R0 EQUAL TO SOURCE LENGTH
910	001640	001401		BEQ	1\$:BR, IF FQUAL
911	001642	104003		ERROR	3	:*****TEST 2 - ERROR 3*****
912						:R0 SHOULD EQUAL SOURCE LENGTH
913	001644					1\$:
914	001644	005701		TST	R1	:CHECK OTHER GENERAL REGISTERS
915	001646	001401		BEQ	66\$:TEST R1
916	001650	104004		ERROR	4	:BR, IF ZERO
917						:*****TEST 2 - ERROR 4*****
918	001652	005702		TST	R2	:R1 SHOULD BE ZERO
919	001654	001401		BEQ	67\$:TEST R2
920	001656	104005		ERROR	5	:BR IF ZERO
921						:*****TEST 2 - ERROR 5*****
922	001660	005703		TST	R3	:R2 SHOULD BE ZERO
923	001662	001401		BEQ	68\$:TEST R3
924	001664	104006		ERROR	6	:BR, IF ZERO
925						:*****TEST 2 - ERROR 6*****
926	001666					68\$:
927	001666	026704	176762	CMP	FILL,R4	:CHECK R4 UNCHANGED
928	001672	001401		BEQ	69\$:BR IF OK
929	001674	104007		ERROR	7	:*****TEST 2 - ERROR 7*****
930						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
931	001676	026705	176754	CMP	TABLE,R5	:CHECK R5 UNCHANGED
932	001702	001401		BEQ	70\$:BR IF OK
933	001704	104010		ERROR	10	:*****TEST 2 - ERROR 10*****
934						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
935	001706					70\$:
936						:VERIFY DESTINATION UNCHANGED
937	001706	012700	016710	MOV	#BUF2,R0	:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
938	001712	105720		TSTB	(R0)+	:TEST CONTENTS OF BOUNDARY
939	001714	001401		BEQ	71\$:BR, IF STILL ZERO

TEST 'MOVC' INSTRUCTION WITH ZERO DESTINATION LENGTH

```

940 001716 104011          ERROR 11          :*****TEST 2 - ERROR 11*****
941                                     :LOWER BOUNDARY OF DESTINATION CHANGED
942                                     : SHOULD STILL EQUAL ZERO
943 001720          71$:          TSTB      (R0)+          :TEST CONTENTS OF DESTINATION BYTE
944 001720 105720          BEQ      2$          :BR, IF STILL ZERO
945 001722 001401          ERROR 12          :*****TEST 2 - ERROR 12*****
946 001724 104012          :DESTINATION WAS CHANGED
947
948 001726          2$:          TSTB      (R0)+          :TEST CONTENTS OF DEST. UPPER BOUNDARY
949 001726 105720          BEQ      ENDT2         :BR, IF STILL ZERO
950 001730 001401          ERROR 13          :*****TEST 2 - ERROR 13*****
951 001732 104013          :UPPER BOUNDARY OF DEST. CHANGED
952                                     : SHOULD STILL EQUAL ZERO
953
954 001734          ENDT2:

```

```

959 :*****
960 :*TEST 3      TEST 'MOVC' INSTRUCTION WITH SRCAD .LT. DSTAD, SL .GT. DL
961 :*****
962 :*PROPER TERMINATION FOR THIS INSTRUCTION TEST
963 :*IS A TRUNCATED SOURCE STORED IN THE DESTINATION
964 :*(LEAST SIGNIFICANT BYTES NOT MOVED), R0 EQUALS THE
965 :*NUMBER OF UNMOVED SOURCE BYTES (SRCLN-DSTLN),
966 :*R1-->R3 EQUAL TO ZERO, AND ALL CONDITION
967 :*CODES CLEAR
968 :*****
969 :*****

```

```

970 001734 000004          TST3:  SCOPE
971 001736 004567 012306  JSR      R5,PREP          :SET UP INSTRUCTION ARGUMENTS
972 001742 000020          20          :SOURCE LENGTH
973 001744 016310          BUF1          :SOURCE ADDRESS
974 001746 000011          11          :DESTINATION LENGTH
975 001750 016711          BUF2+1        :DESTINATION ADDRESS
976 001752 000377          377          :FILL CHARACTER
977 001754 004767 012316  JSR      PC,CLDST         :CLEAR DESTINATION
978 001760 004567 012360  JSR      R5,XPSW         :STORE EXPECTED PSW VALUE
979 001764 000200          .WORD    200
980 001766 004767 012212  JSR      PC,GENR         :SET UP GENERAL REGISTERS
981 001772 000277          SCC          :SET ALL CONDITION CODES
982                                     :EXECUTE 'MOVE CHARACTER' INSTRUCTION
983 001774 076030          MOVC          :CHECK RESULTS
984                                     :CHECK PSW, GENERATE CONDITION CODES
985 001776 004767 012366  JSR      PC,CKCC
986 002002 001401          BEQ      64$
987 002004 104001          ERROR 1          :*****TEST 3 - ERROR 1*****
988                                     :PSW ERROR
989                                     :EXPECTED PSW IS STORED AT 'EXPPSW'
990                                     :ACTUAL PSW IS STORED AT 'CCODES'
991
992 002006          64$:          CMP      @#SAVR6,SP       :VERIFY STACK POINTER IS RESTORED
993 002012 001403          BEQ      65$          :BR IF OK
994 002014 010637 000700  MOV      SP,@#BADR6      :STORE BAD SP VALUE
995 002020 104002          ERROR 2          :*****TEST 3 - ERROR 2*****

```


1052 002144

ENDT3:

1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107

002144 000004
002146 004567 012076
002152 000374
002154 016310
002156 000376
002160 016711
002162 000377
002164 004767 012106
002170 004567 012150
002174 000211
002176 004767 012002
002202 000251
002204 000266
002206 076030
002210 004767 012154
002214 001401
002216 104001
002220
002220 023706 000676
002224 001403
002226 010637 000700
002232 104002
002234
002234 005700
002236 001401
002240 104003
002242
002242 005701
002244 001401
002246 104004
002250 005702
002252 001401

```
*****
*TEST 4 TEST 'MOVC' INSTRUCTION WITH SRCAD .LT. DSTAD, DL .GT.SL
*****
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS
*A TRANSFER OF ALL BYTES FROM SOURCE TO DESTINATION
*AND 'FILL' CHARACTERS IN THE LSB OF THE DESTINATION,
*RO-->R3 EQUAL TO ZERO, AND CONDITION CODES N,C=1
*AND Z,V=0
*****
TST4: SCOPE
      JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
          374 ;SOURCE LENGTH
          BUF1 ;SOURCE ADDRESS
          376 ;DESTINATION LENGTH
          BUF2+1 ;DESTINATION ADDRESS
          377 ;FILL CHARACTER
      JSR PC,CLDST ;CLEAR DESTINATION AREA & SET DEST. ADDRESS
      JSR R5,XPSW ;STORE EXPECTED PSW VALUE
      .WORD 211
      JSR PC,GENR ;SET UP GENERAL REGISTERS
      +CLN!CLC ;CLEAR CONDITION CODES N&C
      +SEV!SEZ ;SET CONDITION CODES V&Z
      MOVC ;EXECUTE 'MOVE CHARACTER' INSTRUCTION
          ;CHECK RESULTS
          JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES
          BEQ 64$
          ERROR 1 ;*****TEST 4 - ERROR 1*****
          ;PSW ERROR
          ;EXPECTED PSW IS STORED AT 'EXPPSW'
          ;ACTUAL PSW IS STORED AT 'CCODES'
64$:
      CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
      BEQ 65$ ;BR IF OK
      MOV SP,@#BALR6 ;STORE BAD SP VALUE
      ERROR 2 ;*****TEST 4 - ERROR 2*****
          ;STACK POINTER NOT RESTORED BY INSTRUCTION
          ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
          ;ERRONEOUS VALUE IS AT 'BADR6'
65$:
      TST R0 ;CHECK R0=ZERO
      BEQ 1$ ;BR, IF ZERO
      ERROR 3 ;*****TEST 4 - ERROR 3*****
          ;R0 SHOULD BE ZERO
          ;CHECK OTHER GENERAL REGISTERS
1$:
      TST R1 ;TEST R1
      BEQ 66$ ;BR, IF ZERO
      ERROR 4 ;*****TEST 4 - ERROR 4*****
          ;R1 SHOULD BE ZERO
66$:
      TST R2 ;TEST R2
      BEQ 67$ ;BR IF ZERO
```

```

1108 002254 104005          ERROR 5          :*****TEST 4 - ERROR 5*****
1109                               :R2 SHOULD BE ZERO
1110 002256 005703      67$:  TST      R3          :TEST R3
1111 002260 001401          BEQ      68$         :BR, IF ZERO
1112 002262 104006          ERROR 6          :*****TEST 4 - ERROR 6*****
1113                               :R3 SHOULD BE ZERO
1114 002264                               68$:
1115 002264 026704 176364    CMP      FILL,R4       :CHECK R4 UNCHANGED
1116 CJ2270 001401          BEQ      69$         :BR IF OK
1117 002272 104007          ERROR 7          :*****TEST 4 - ERROR 7*****
1118                               :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1119 002274 026705 176356    69$:  CMP      TABLE,R5    :CHECK R5 UNCHANGED
1120 002300 001401          BEQ      70$         :BR IF OK
1121 002302 104010          ERROR 10         :*****TEST 4 - ERROR 10*****
1122                               :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
1123 002304                               70$:
1124                               :VERIFY DESTINATION
1125 002304 012700 016710    MOV      #BUF2,R0     :POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1126 002310 105720          TSTB    (R0)+         :TEST CONTENTS OF BOUNDARY
1127 002312 001401          BEQ      71$         :BR, IF STILL ZERO
1128 002314 104011          ERROR 11         :*****TEST 4 - ERROR 11*****
1129                               :LOWER BOUNDARY OF DESTINATION CHANGED
1130                               : SHOULD STILL EQUAL ZERO
1131 002316                               71$:
1132 002316 016702 176324    2$:  MOV      SRCAD,R2     :POINT R2 TO SOURCE ADDRESS
1133 002322 016701 176316    MOV      SRCLN,R1     :STORE TRANSFER BYTE COUNT IN R1
1134 002326 122022      T4E12:  CMPB    (R0)+,(R2)+  :CHECK CHARACTERS IN DESTINATION
1135 002330 001401          BEQ      +4          :BR IF OK
1136 002332 104012          ERROR 12         :*****TEST 4 - ERROR 12*****
1137                               :COMPARE ERROR IN DESTINATION
1138                               :R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1139
1140 002334 005301          DEC      R1           :DECREMENT BYTE COUNT
1141 002336 001373          BNE     T4E12        :BR, IF NOT FINISHED CHECKING
1142 002340 016705 176304    MOV      DSTLN,R5     :CALCULATE THE NUMBER OF 'FILL'
1143 002344 166705 176274    SUB     SRCLN,R5     :CHARACTERS THAT SHOULD APPEAR IN DEST.
1144 002350 010501          MOV     R5,R1        :STORE TRANSFER BYTE COUNT IN R1
1145 002352 122067 176276    T4E13:  CMPB    (R0)+,FILL  :CHECK CHARACTERS IN DESTINATION
1146 002356 001401          BEQ     +4          :BR IF OK
1147 002360 104013          ERROR 13         :*****TEST 4 - ERROR 13*****
1148                               :COMPARE ERROR IN DESTINATION
1149                               :R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1150
1151 002362 005301          DEC     R1           :DECREMENT BYTE COUNT
1152 002364 001372          BNE     T4E13        :BR, IF NOT FINISHED CHECKING
1153 002366 105720          TSTB   (R0)+         :TEST CONTENTS OF DEST. UPPER BOUNDARY
1154 002370 001401          BEQ     ENDT4       :BR, IF STILL ZERO
1155 002372 104014          ERROR 14         :*****TEST 4 - ERROR 14*****
1156                               :UPPER BOUNDARY OF DEST. CHANGED
1157                               : SHOULD STILL EQUAL ZERO
1158 002374          ENDT4:
1159
1160          :*****
1161          :*TEST 5      TEST 'MOVC' INSTRUCTION WITH SRCAD=DSTAD, SL .LT. DL
1162          :*****
1163          :*PROPER TERMINATION OF THIS INSTRUCTION TEST IS

```



```

1164
1165
1166
1167
1168
1169
1170
1171
1172
1173 002374 000004
1174 002376 004567 011646
1175 002402 000010
1176 002404 016711
1177 002406 000015
1178 002410 016711
1179 002412 000377
1180 002414 004767 011656
1181 002420 004567 011700
1182 002424 000010
1183 002426 016711
1184 002430 004567 011710
1185 002434 000211
1186 002436 004767 011542
1187 002442 000245
1188 002444 000272
1189
1190 002446 076030
1191
1192 002450 004767 011714
1193 002454 001401
1194 002456 104001
1195
1196
1197
1198 002460
1199 002460 023706 000676
1200 002464 001403
1201 002466 010637 000700
1202 002472 104002
1203
1204
1205
1206 002474
1207 002474 005700
1208 002476 001401
1209 002500 104003
1210
1211 002502
1212 002502 005701
1213 002504 001401
1214 002506 104004
1215
1216 002510 005702
1217 002512 001401
1218 002514 104005
1219

```

```

;*NO CHARACTERS TRANSFERED TO THE DESTINATION,
;*R0-->R3 EQUAL TO ZERO, AND CONDITION CODES
;*N,V=1 AND C,Z=0.
;*BOTH SOURCE AND DESTINATION ADDRESSES SHOULD
;*BE DON'T CARE VALUES, THEREFORE A PROBABLE
;*NON-EXISTANT MEMORY LOCATION WILL BE USED FOR
;*BOTH ADDRESSES
:*****
:*****
TST5:  SCOPE
      JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
      10          ;SOURCE LENGTH
      BUF2+1      ;SOURCE ADDRESS
      15          ;DESTINATION LENGTH
      BUF2+1      ;DESTINATION ADDRESS
      377         ;FILL CHARACTER
      JSR      PC,CLDST
      JSR      R5,GENSRC       ;GENERATE A SOURCE STRING
      .WORD    10
      .WORD    BUF2+1
      JSR      R5,XPSW        ;STORE EXPECTED PSW VALUE
      .WORD    211
      JSR      PC,GENR        ;SET UP GENERAL REGISTERS
      +CLZ!CLC              ;CLEAR CONDITION CODES Z & C
      +SEV!SEN              ;SET CONDITION CODES V & N
                               ;EXECUTE 'MOVE CHARACTER' INSTRUCTION
      MOVC
                               ;CHECK RESULTS
      JSR      PC,CKCC        ;CHECK PSW, GENERATE CONDITION CODES
      BEQ     64$
      ERROR   1
                               ;*****TEST 5 - ERROR 1*****
                               ;PSW ERROR
                               ;EXPECTED PSW IS STORED AT 'EXPPSW'
                               ;ACTUAL PSW IS STORED AT 'CCODES'
64$:
      CMP     @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
      BEQ     65$
      MOV     SP,@#BADR6     ;BR IF OK
                               ;STORE BAD SP VALUE
      ERROR   2
                               ;*****TEST 5 - ERROR 2*****
                               ;STACK POINTER NOT RESTORED BY INSTRUCTION
                               ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
                               ;ERRONEOUS VALUE IS AT 'BADR6'
65$:
      TST     R0
      BEQ     1$
      ERROR   3
                               ;CHECK R0 FOR ZERO
                               ;BR, IF EQUAL
                               ;*****TEST 5 - ERROR 3*****
                               ;R0 SHOULD BE ZERO
                               ;CHECK OTHER GENERAL REGISTERS
1$:
      TST     R1
      BEQ     66$
      ERROR   4
                               ;TEST R1
                               ;BR, IF ZERO
                               ;*****TEST 5 - ERROR 4*****
                               ;R1 SHOULD BE ZERO
66$:
      TST     R2
      BEQ     67$
      ERROR   5
                               ;TEST R2
                               ;BR IF ZERO
                               ;*****TEST 5 - ERROR 5*****
                               ;R2 SHOULD BE ZERO

```

```

1220 002516 005703      67$:  TST      R3          ;TEST R3
1221 002520 001401      BEQ      68$          ;BR, IF ZERO
1222 002522 104006      ERROR    6           ;*****TEST 5 - ERROR 6*****
1223                                     ;R3 SHOULD BE ZERO
1224 002524                                     68$:
1225 002524 026704 176124  CMP      FILL,R4      ;CHECK R4 UNCHANGED
1226 002530 001401      BEQ      69$          ;BR IF OK
1227 002532 104007      ERROR    7           ;*****TEST 5 - ERROR 7*****
1228                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1229 002534 026705 176116 69$:  CMP      TABLE,R5   ;CHECK R5 UNCHANGED
1230 002540 001401      BEQ      70$          ;BR IF OK
1231 002542 104010      ERROR    10          ;*****TEST 5 - ERROR 10*****
1232                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
1233 002544                                     70$:
1234 002544 012700 016710  MOV      #BUF2,R0     ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1235 002550 105720      TSTB    (R0)+         ;TEST CONTENTS OF BOUNDARY
1236 002552 001401      BEQ      71$          ;BR, IF STILL ZERO
1237 002554 104011      ERROR    11          ;*****TEST 5 - ERROR 11*****
1238                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
1239                                     ; SHOULD STILL EQUAL ZERO
1240 002556                                     71$:
1241 002556 012702 016711  MOV      #BUF2+1,R2   ;STORE TRANSFER BYTE COUNT IN R1
1242 002562 016701 176056  MOV      SRCLN,R1     ;CHECK CHARACTERS IN DESTINATION
1243 002566 122022      CMPB    (R0)+,(R2)+   ;BR IF OK
1244 002570 001401      BEQ      +4           ;*****TEST 5 - ERROR 12*****
1245 002572 104012      ERROR    12          ;COMPARE ERROR IN DESTINATION
1246                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1247
1248
1249 002574 005301      DEC      R1           ;DECREMENT BYTE COUNT
1250 002576 001373      BNE     T5E12        ;BR, IF NOT FINISHED CHECKING
1251 002600 016702 176044  MOV      DSTLN,R2     ;CALCULATE FILL LENGTH
1252 002604 166702 176034  SUB      SRCLN,R2
1253 002610 010201      MOV      R2,R1       ;STORE TRANSFER BYTE COUNT IN R1
1254 002612 122067 176036  T5E13:  CMPB    (R0)+,FILL   ;CHECK CHARACTERS IN DESTINATION
1255 002616 001401      BEQ      +4           ;BR IF OK
1256 002620 104013      ERROR    13          ;*****TEST 5 - ERROR 13*****
1257                                     ;COMPARE ERROR IN DESTINATION
1258                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1259
1260 002622 005301      DEC      R1           ;DECREMENT BYTE COUNT
1261 002624 001372      BNE     T5E13        ;BR, IF NOT FINISHED CHECKING
1262 002626 105720      TSTB    (R0)+         ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1263 002630 001401      BEQ      ENDT5       ;BR, IF STILL ZERO
1264 002632 104014      ERROR    14          ;*****TEST 5 - ERROR 14*****
1265                                     ;UPPER BOUNDARY OF DEST. CHANGED
1266                                     ; SHOULD STILL EQUAL ZERO
1267 002634      ENDT5:
1268
1269
1270      ;*****
1271      ;*TEST 6      TEST 'MOVC' WITH SRCAD .LT. DSTAD, SRCLN .GT. DSTLN
1272      ;*****
1273      ;*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN
1274      ;*THE DESTINATION AREA IS CONTAINED WITHIN THE SOURCE AREA.
1275      ;*THEREFORE, SRCAD<DSTAD AND SRCLN>DSTLN

```



```

1332                                     :R1 SHOULD BE ZERO
1333 002770 005702 68$: TST R2 :TEST R2
1334 002772 001401 BEQ 69$ :BR IF ZERO
1335 002774 104007 ERROR 7 :*****TEST 6 - ERROR 7*****
1336                                     :R2 SHOULD BE ZERO
1337 002776 005703 69$: TST R3 :TEST R3
1338 003000 001401 BEQ 70$ :BR, IF ZERO
1339 003002 104010 ERROR 10 :*****TEST 6 - ERROR 10*****
1340                                     :R3 SHOULD BE ZERO
1341 003004 70$:
1342                                     :VERIFY DESTINATION & NON-OVERLAP SOURCE
1343 003004 012700 016310 MOV #BUF1,R0 :CHECK BEGINNING OF OLD SOURCE AREAS
1344 003010 012701 016710 MOV #BUF2,R1
1345 003014 122021 3$: CMPB (R0)+,(R1)+
1346 003016 001401 BEQ 4$
1347 003020 104011 ERROR 11 :*****TEST 6 - ERROR 11*****
1348                                     :NON-OVERLAP SOURCE AREA CHANGED
1349                                     :R0 CONTAINS PC+1 OF CHANGED SOURCE CHARACTER
1350 003022 020167 175624 4$: CMP R1,DSTAD :REACHED START OF DESTINATION?
1351 003026 002772 BLT 3$ :BR, IF NO
1352 003030 012702 016310 MOV #BUF1,R2 :CHECK DESTINATION AREA
1353 003034 016703 175610 MOV DSTLN,R3
1354 003040 122221 5$: CMPB (R2)+,(R1)+
1355 003042 001401 BEQ 6$
1356 003044 104012 ERROR 12 :*****TEST 6 - ERROR 12*****
1357                                     :COMPARE ERROR IN DESTINATION
1358                                     :R1 CONTAINS PC+1 OF BAD DESTINATION CHARACTER
1359 003046 005200 6$: INC R0 :UPDATE POINTER TO OLD SOURCE
1360 003050 005303 DEC R3 :DECREMENT DEST. BYTE COUNT
1361 003052 001372 BNE 5$ :BR, IF NOT FINISHED
1362 003054 016703 175566 MOV SRCAD,R3 :CHECK LOWER PORTION OF NON-OVERLAP SOURCE
1363 003060 066703 175560 ADD SRCLN,R3
1364 003064 122021 7$: CMPB (R0)+,(R1)+
1365 003066 001401 BEQ 10$
1366 003070 104013 ERROR 13 :*****TEST 6 - ERROR 13*****
1367                                     :NON-OVERLAP SOURCE AREA CHANGED
1368                                     :R0 CONTAINS PC+1 OF CHANGED SOURCE CHARACTER
1369 003072 020103 10$: CMP R1,R3 :FINISHED CHECK?
1370 003074 002773 BLT 7$ :BR, IF NO

```

```

1371
1372
1373 :*****
1374 :*TEST 7 TEST 'MOVC' WITH DSTAD .GT. SRCAD, SRCLN .LT. DSTLN
1375 :*****
1376 :*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN
1377 :*THE SOURCE AREA IS CONTAINED WITHIN THE DESTINATION AREA.
1378 :*THEREFORE, SRCAD>DSTAD AND DSTLN>SRCLN.
1379 :*THE RESULT IS A FULL TRANSFER OF THE SOURCE TO THE
1380 :*DESTINATION AND "FILL" CHARACTERS IN THE LSB OF THE
1381 :*DESTINATION; R0-->R3 EQUAL ZERO, AND CONDITION
1382 :*CODES N,C=1 AND Z,V=0
1383 :*****
1384 :*****
1385 003076 000004 TST7: SCOPE
1386 003100 004567 JSR R5,PREP :SET UP INSTRUCTION ARGUMENTS
1387 003104 000007 7 :SOURCE LENGTH

```


1388	003106	016713			BUF2+3	:SOURCE ADDRESS
1389	003110	000017			17	:DESTINATION LENGTH
1390	003112	016711			BUF2+1	:DESTINATION ADDRESS
1391	003114	000377			377	:FILL CHARACTER
1392	003116	004767	011154	JSR	PC,CLDST	:CLEAR DESTINATION
1393	003122	004567	011176	JSR	R5,GENSRC	:GENERATE A SOURCE STRING
1394	003126	000007		.WORD	7	
1395	003130	016713		.WORD	BUF2+3	
1396	003132	004567	011206	JSR	R5,XPSW	:STORE EXPECTED PSW VALUE
1397	003136	000211		.WORD	211	
1398	003140	004767	011040	JSR	PC,GENR	:SET UP GENERAL REGISTERS
1399	003144	000251		+CLN!CLC		:CLEAR CONDITION CODES N & C
1400	003146	000266		+SEV!SEZ		:SET CONDITION CODES V & Z
1401						:EXECUTE 'MOVE CHARACTER' INSTRUCTION
1402	003150	076030		MOVC		
1403						:CHECK RESULTS
1404	003152	004767	011212	JSR	PC,CKCC	:CHECK PSW, GENERATE CONDITION CODES
1405	003156	001401		BEQ	64\$	
1406	003160	104001		ERROR	1	:*****TEST 7 - ERROR 1*****
1407						:PSW ERROR
1408						:EXPECTED PSW IS STORED AT 'EXPPSW'
1409						:ACTUAL PSW IS STORED AT 'CCODES'
1410	003162			64\$:		
1411	003162	023706	000676	CMP	@SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
1412	003166	001403		BEQ	65\$:BR IF OK
1413	003170	010637	000700	MOV	SP,@BADR6	:STORE BAD SP VALUE
1414	003174	104002		ERROR	2	:*****TEST 7 - ERROR 2*****
1415						:STACK POINTER NOT RESTORED BY INSTRUCTION
1416						:EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
1417						:ERRONEOUS VALUE IS AT 'BADR6'
1418	003176			65\$:		
1419	003176	005700		TST	R0	:CHECK R0 FOR ZERO
1420	003200	001401		BEQ	1\$:BR, IF ZERO
1421	003202	104003		ERROR	3	:*****TEST 7 - ERROR 3*****
1422						:R0 SHOULD BE ZERO
1423	003204			1\$:		
1424	003204	005701		TST	R1	:TEST R1
1425	003206	001401		BEQ	66\$:BR, IF ZERO
1426	003210	104004		ERROR	4	:*****TEST 7 - ERROR 4*****
1427						:R1 SHOULD BE ZERO
1428	003212	005702		66\$:		
1429	003214	001401		TST	R2	:TEST R2
1430	003216	104005		BEQ	67\$:BR IF ZERO
1431				ERROR	5	:*****TEST 7 - ERROR 5*****
1432	003220	005703				:R2 SHOULD BE ZERO
1433	003222	001401		67\$:		
1434	003224	104006		TST	R3	:TEST R3
1435				BEQ	68\$:BR, IF ZERO
1436	003226			ERROR	6	:*****TEST 7 - ERROR 6*****
1437	003226	026704	175422			:R3 SHOULD BE ZERO
1438	003232	001401		68\$:		
1439	003234	104007		CMP	FILL,R4	:CHECK R4 UNCHANGED
1440				BEQ	69\$:BR IF OK
1441	003236	026705	175414	ERROR	7	:*****TEST 7 - ERROR 7*****
1442	003242	001401				:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1443	003244	104010		69\$:		
				CMP	TABLE,R5	:CHECK R5 UNCHANGED
				BEQ	70\$:BR IF OK
				ERROR	10	:*****TEST 7 - ERROR 10*****

1444
 1445 003246
 1446
 1447 003246 012700 016710
 1448 003252 105720
 1449 003254 001401
 1450 003256 104011
 1451
 1452
 1453 003260
 1454 003260 012702 016310
 1455 003264 016701 175354
 1456 003270 122022
 1457 003272 001401
 1458 003274 104012
 1459
 1460
 1461
 1462 003276 005301
 1463 003300 001373
 1464 003302 016705 175342
 1465 003306 166705 175332
 1466 003312 122067 175336
 1467 003316 001401
 1468 003320 104013
 1469
 1470 003322 105305
 1471 003324 001372
 1472 003326 105720
 1473 003330 001401
 1474 003332 104014
 1475
 1476
 1477 003334

70\$:

MOV #BUF2,R0
 TSTB (R0)+
 BEQ 71\$
 ERROR 11

;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
 ;VERIFY DESTINATION
 ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
 ;TEST CONTENTS OF BOUNDARY
 ;BR, IF STILL ZERO
 ;*****TEST 7 - ERROR 11*****
 ;LOWER BOUNDARY OF DESTINATION CHANGED
 ; SHOULD STILL EQUAL ZERO

71\$:

2\$:

T7E12:

MOV #BUF1,R2
 MOV SRCLN,R1
 CMPB (R0)+,(R2)+
 BEQ +4
 ERROR 12

;POINT R2 TO ORIGINAL SOURCE BYTES
 ;STORE TRANSFER BYTE COUNT IN R1
 ;CHECK CHARACTERS IN DESTINATION
 ;BR IF OK
 ;*****TEST 7 - ERROR 12*****
 ;COMPARE ERROR IN DESTINATION
 ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER

3\$:

DEC R1
 BNE T7E12
 MOV DSTLN,R5
 SUB SRCLN,R5
 CMPB (R0)+,FILL
 BEQ 4\$
 ERROR 13

;DECREMENT BYTE COUNT
 ;BR, IF NOT FINISHED CHECKING
 ;CHECK FILL
 ;*****TEST 7 - ERROR 13*****
 ;'FILL' ERROR IN DESTINATION

4\$:

DECB R5
 BNE 3\$
 TSTB (R0)+
 BEQ ENDT7
 ERROR 14

;TEST CONTENTS OF DEST. UPPER BOUNDARY
 ;BR, IF STILL ZERO
 ;*****TEST 7 - ERROR 14*****
 ;UPPER BOUNDARY OF DEST. CHANGED
 ; SHOULD STILL EQUAL ZERO

ENDT7:

1475
 1476
 1477 003334
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1490
 1491 003334 000004
 1492 003336 004567 010706
 1493 003342 000020
 1494 003344 016730
 1495 003346 000007
 1496 003350 016711
 1497 003352 000377
 1498 003354 004767 010716
 1499 003360 004567 010740

 *TEST 10 TEST 'MOVC' WITH DSTAD .LT. SRCAD & SRCLN .GT. DSTLN, (TRUNCATON)

 *THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN THE
 *SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION AND THE TRANSFER
 *IS TRUNCATED (SRCLN > DSTLN).
 *THE RESULT IS A TRUNCATED SOURCE IN THE DESTINATION, R0-->R3
 *EQUAL ZERO, AND ALL CONDITION CODES CLEAR.

TST10:

SCOPE
 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
 20 ;SOURCE LENGTH
 BUF2+20 ;SOURCE ADDRESS
 7 ;DESTINATION LENGTH
 BUF2+1 ;DESTINATION ADDRESS
 377 ;FILL CHARACTER
 JSR PC,CLDST ;CLEAR DESTINATION AREA
 JSR R5,GENSRC ;GENERATE SOURCE STRING

MAIN. MACY11 30(1046) 22-JAN-82 08:44 PAGE 30
 CVKAIB.P11 22-JAN-82 08:43 T10

TEST 'MOVC' WITH DSTAD .LT. SRCAD & SRCLN .GT. DSTLN, (TRUNCATON)

SEQ 0029

1500	003364	000020		.WORD	20		
1501	003366	016730		.WORD	BUF2+20		
1502	003370	004567	010750	JSR	R5,XPSW		;STORE EXPECTED PSW VALUE
1503	003374	000200		.WORD	200		
1504	003376	004767	010602	JSR	PC,GENR		;SET UP GENERAL REGISTERS
1505	003402	000277		SCC			;SET ALL CONDITION CODES
1506							
1507	003404	076030		MOVC			;EXECUTE 'MOVE CHARACTER' INSTRUCTION
1508							
1509	003406	004767	010756	JSR	PC,CKCC		;CHECK PSW, GENERATE CONDITION CODES
1510	003412	001401		BEQ	64\$		
1511	003414	104001		ERROR	1		*****TEST 10 - ERROR 1*****
1512							;PSW ERROR
1513							;EXPECTED PSW IS STORED AT 'EXPPSW'
1514							;ACTUAL PSW IS STORED AT 'CCODES'
1515	003416					64\$:	
1516	003416	023706	000676	CMP	@#SAVR6,SP		;VERIFY STACK POINTER IS RESTORED
1517	003422	001403		BEQ	65\$;BR IF OK
1518	003424	010637	000700	MOV	SP,@#BADR6		;STORE BAD SP VALUE
1519	003430	104002		ERROR	2		*****TEST 10 - ERROR 2*****
1520							;STACK POINTER NOT RESTORED BY INSTRUCTION
1521							;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
1522							;ERRONEOUS VALUE IS AT 'BADR6'
1523	003432					65\$:	
1524	003432	026704	175216	CMP	FILL,R4		;CHECK R4 UNCHANGED
1525	003436	001401		BEQ	66\$;BR IF OK
1526	003440	104003		ERROR	3		*****TEST 10 - ERROR 3*****
1527							;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1528	003442	026705	175210	CMP	TABLE,R5		;CHECK R5 UNCHANGED
1529	003446	001401		BEQ	67\$;BR IF OK
1530	003450	104004		ERROR	4		*****TEST 10 - ERROR 4*****
1531							;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
1532	003452					67\$:	
1533	003452	016705	175166	MOV	SRCLN,R5		;CHECK R0=SOURCE LENGTH MINUS
1534	003456	166705	175166	SUB	DSTLN,R5		; DESTINATION LENGTH
1535	003462	020005		CMP	R0,R5		
1536	003464	001401		BEQ	68\$		
1537	003466	104005		ERROR	5		*****TEST 10 - ERROR 5*****
1538							;R0 NOT EQUAL TO (SRCLN-DSTLN)
1539	003470					68\$:	;CHECK OTHER GENERAL REGISTERS
1540	003470	005701		TST	R1		;TEST R1
1541	003472	001401		BEQ	69\$;BR, IF ZERO
1542	003474	104006		ERROR	6		*****TEST 10 - ERROR 6*****
1543							;R1 SHOULD BE ZERO
1544	003476	005702		TST	R2		;TEST R2
1545	003500	001401		BEQ	70\$;BR IF ZERO
1546	003502	104007		ERROR	7		*****TEST 10 - ERROR 7*****
1547							;R2 SHOULD BE ZERO
1548	003504	005703		TST	R3		;TEST R3
1549	003506	001401		BEQ	71\$;BR, IF ZERO
1550	003510	104010		ERROR	10		*****TEST 10 - ERROR 10*****
1551							;R3 SHOULD BE ZERO
1552	003512					71\$:	
1553							;VERIFY DESTINATION AREA
1554	003512	012700	016710	MOV	#BUF2,R0		;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1555	003516	105720		TSTB	(R0)+		;TEST CONTENTS OF BOUNDARY

```

1556 003520 001401          BEQ      72$          ;BR, IF STILL ZERO
1557 003522 104011          ERROR    11          ;*****TEST 10 - ERROR 11*****
1558                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
1559                                     ; SHOULD STILL EQUAL ZERO
1560 003524          72$:
1561 003524 016702 175116    MOV      SRCAD,R2
1562 003530 016701 175114    MOV      DSTLN,R1
1563 003534 122022          T10E12: CMPB     (R0)+,(R2)+ ;STORE TRANSFER BYTE COUNT IN R1
1564 003536 001401          BEQ      +4          ;CHECK CHARACTERS IN DESTINATION
1565 003540 104012          ERROR    12          ;BR IF OK
1566                                     ;*****TEST 10 - ERROR 12*****
1567                                     ;COMPARE ERROR IN DESTINATION
1568                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1569 003542 005301          DEC      R1          ;DECREMENT BYTE COUNT
1570 003544 001373          BNE     T10E12      ;BR, IF NOT FINISHED CHECKING
1571 003546 105720          TSTB    (R0)+       ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1572 003550 001401          BEQ      ENDT10     ;BR, IF STILL ZERO
1573 003552 104013          ERROR    13          ;*****TEST 10 - ERROR 13*****
1574                                     ;UPPER BOUNDARY OF DEST. CHANGED
1575                                     ; SHOULD STILL EQUAL ZERO
1576 003554          ENDT10:
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587 003554 000004          TST11: SCOPE
1588 003556 004567 010466    JSR     R5,PREP     ;SET UP INSTRUCTION ARGUMENTS
1589 003562 016310          BUF1    10          ;SOURCE LENGTH
1590 003564 000010          BUF2    10          ;SOURCE ADDRESS
1591 003566 016710          377     377        ;DESTINATION LENGTH
1592 003570 000010          MOV     #076032,a#BD11 ;DESTINATION ADDRESS
1593 003572 000377          MOV     a#10,TEMP1  ;FILL CHARACTER
1594 003574 012737 076032 003622 ;STORE THE FIRST BAD MOVE OPCODE
1595 003602 013767 000010 175076 ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR
1596 003610 012737 003632 000010 ;POINT ILLEGAL INSTRUCTION VECTOR TO CONTINUE TEST
1597 003616 004767 010362          REP11: JSR     PC,GENR ;SET UP GENERAL REGISTERS
1598
1599 003622 076032          BD11:  .WORD    076032 ;EXECUTE BAD MOVE INSTRUCTION
1600
1601 003624 016700 177772    MOV     BD11,R0     ;STORE BAD OPCODE THAT DID NOT TRAP
1602 003630 104001          ERROR    1          ;*****TEST 11 - ERROR 1*****
1603                                     ;BAD MOVE OPCODE DID NOT TRAP
1604                                     ;R0 CONTAINS THE BAD OPCODE
1605
1606 003632 012626          T11CONT:MOV    (SP)+,(SP)+ ;RESTORE THE STACK POINTER AFTER THE TRAP
1607 003634 005267 177762    INC     BD11        ;INCREMENT INSTRUCTION OPCODE
1608 003640 022767 076040 177754 ;FINISHED WITH BAD MOVE OPCODES?
1609 003646 001363          BNE     REP11       ;BR IF NOT
1610 003650 016737 175032 000010 ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR
1611

```


1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667

003656 000004
003660 004767 010526
003664 005767 175014
003670 001155
003672 004567 010352
003676 000374
003700 016310
003702 000376
003704 016711
003706 000377
003710 004767 010362
003714 012767 003776 174746
003722 012777 014506 174734
003730 005077 174732
003734 004767 010522
003740 013777 000554 174714
003746 004567 010372
003752 000011
003754 106427 000000
003760 052777 000100 174672
003766 004767 010212
003772 000251
003774 000266
003776 076030
004000 106767 174666
004004 032777 000100 174646
004012 001365
004014 042767 177400 174650
004022 023767 000674 174642
004030 001401
004032 104001
004034
004034 023706 000676
004040 001403
004042 010637 000700

```
*****  
:TEST 12 TEST INTERRUPTABILITY OF 'MOVC' INSTRUCTION  
:*****  
:THIS TEST INTERRUPTS THE EXECUTION OF THE 'MOVC'  
:INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE  
:INTERRUPT, AND VERIFIES THE RESULTS.  
:THE PROPER RESULT IS ALL BYTES MOVED  
:TO THE DESTINATION, R0-->R3 EQUAL ZERO  
:AND CONDITION CODES N,C=1 AND Z,V=0  
:*****  
TST12: SCOPE  
JSR PC,SKPINT ;SET FLAG -- TEMP, WHICH INDICATES IF  
;NEXT TEST IS TO BE EXECUTED NEXT.  
TST TEMP ;CHECK IF -TEMP- IS SET  
BNE TST13 ;TEMP IS SET, GO TO NEXT TEST  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
374 ;SOURCE LENGTH  
BUF1 ;SOURCE ADDRESS  
376 ;DESTINATION LENGTH  
BUF2+1 ;DESTINATION ADDRESS  
377 ;FILL CHARACTER  
JSR PC,CLDST ;CLEAR DESTINATION  
MOV #MC,PCI ;STORE PC OF TEST INSTRUCTION  
MOV #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE  
CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT  
JSR PC,TDONE  
MOV @#NULL,@TBUF ;SEND CARRIAGE RETURN  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 11  
MTPS #0 ;SET PSW TO ALLOW INTERRUPTS  
BIS #100,@TCSR ;ENABLE TTY INTERRUPTS  
REPMC: JSR PC,GENR ;SET UP GENERAL REGISTERS  
+CLN!CLC  
+SEV!SEZ  
MC: MOVC ;EXECUTE 'MOVE CHARACTER' INSTRUCTION  
MFPS CCODES ;STORE THE PSW  
BIT #100,@TCSR ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE  
BNE REPMC ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED  
BIC #177400,CCODES ;CLEAR ALL 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 'EXPPSW'  
;ACTUAL PSW IS STORED AT 'CCODES'  
64$: ;CHECK RESULTS  
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE
```

TEST INTERRUPTABILITY OF 'MOVC' INSTRUCTION

```

1668 004046 104002          ERROR 2          :*****TEST 12 - ERROR 2*****
1669                                     :STACK POINTER NOT RESTORED BY INSTRUCTION
1670                                     :EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
1671                                     :ERRONEOUS VALUE IS AT 'BADR6'
1672 004050          65$:          TST      R0          :CHECK R0=ZERO
1673 004050 005700          BEQ      1$          :
1674 004052 001401          ERROR    3          :*****TEST 12 - ERROR 3*****
1675 004054 104003          :R0 SHOULD BE ZERO
1676                                     :CHECK OTHER GENERAL REGISTERS
1677 004056          1$:          TST      R1          :TEST R1
1678 004056 005701          BEQ      66$         :BR, IF ZERO
1679 004060 001401          ERROR    4          :*****TEST 12 - ERROR 4*****
1680 004062 104004          :R1 SHOULD BE ZERO
1681                                     :TEST R2
1682 004064 005702          66$:          TST      R2          :BR IF ZERO
1683 004066 001401          BEQ      67$         :*****TEST 12 - ERROR 5*****
1684 004070 104005          ERROR    5          :R2 SHOULD BE ZERO
1685                                     :TEST R3
1686 004072 005703          67$:          TST      R3          :BR, IF ZERO
1687 004074 001401          BEQ      68$         :*****TEST 12 - ERROR 6*****
1688 004076 104006          ERROR    6          :R3 SHOULD BE ZERO
1689                                     :
1690 004100          68$:          CMP      FILL,R4         :CHECK R4 UNCHANGED
1691 004100 026704 174550          BEQ      69$         :BR IF OK
1692 004104 001401          ERROR    7          :*****TEST 12 - ERROR 7*****
1693 004106 104007          :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1694                                     :CHECK R5 UNCHANGED
1695 004110 026705 174542          69$:          CMP      TABLE,R5        :BR IF OK
1696 004114 001401          BEQ      70$         :*****TEST 12 - ERROR 10*****
1697 004116 104010          ERROR    10         :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
1698                                     :
1699 004120          70$:          MOV      #BUF2,R0         :POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1700 004120 012700 016710          MOV      #BUF2,R0         :TEST CONTENTS OF BOUNDARY
1701 004124 012700 016710          TSTB    (R0)+           :BR, IF STILL ZERO
1702 004130 105720          BEQ      71$         :*****TEST 12 - ERROR 11*****
1703 004132 001401          ERROR    11         :LOWER BOUNDARY OF DESTINATION CHANGED
1704 004134 104011          : SHOULD STILL EQUAL ZERO
1705                                     :
1706 004136          71$:          MOV      SRCAD,R2         :POINT R2 TO THE SOURCE ADDRESS
1707 004136 016702 174504          4$:          MOV      SRCLN,R1        :STORE TRANSFER BYTE COUNT IN R1
1708 004142 016701 174476          T12E12: CMPB    (R0)+,(R2)+ :CHECK CHARACTERS IN DESTINATION
1709 004146 122022          BEQ      +4           :BR IF OK
1710 004150 001401          ERROR    12         :*****TEST 12 - ERROR 12*****
1711 004152 104012          :COMPARE ERROR IN DESTINATION
1712                                     :R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1713                                     :
1714 004154 005301          DEC      R1          :DECREMENT BYTE COUNT
1715 004156 001373          BNE     T12E12        :BR, IF NOT FINISHED CHECKING
1716 004160 016701 174464          MOV      DSTLN,R1       :CALCULATE THE NUMBER OF 'FILL'
1717 004164 166701 174454          SUB     SRCLN,R1       :CHARACTERS THAT SHOULD APPEAR IN DEST.
1718 004170 122067 174460          5$:          CMPB    (R0)+,FILL     :CHECK LSB'S OF DEST. FOR FILLS
1719 004174 001401          BEQ     6$           :BR, IF EQUAL
1720 004176 104013          ERROR    13         :*****TEST 12 - ERROR 13*****
1721                                     :'FILL' ERROR IN DESTINATION
1722
1723

```



```

1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850 004444 000004
1851 004446 004567 007576
1852 004452 000011
1853 004454 016310
1854 004456 000020
1855 004460 016711
1856 004462 000377
1857 004464 004767 007606
1858 004470 004567 007650
1859 004474 000211
1860 004476 004767 007502
1861 004502 000251
1862 004504 000266
1863
1864 004506 076031
1865
1866 004510 004767 007654
1867 004514 001401
1868 004516 104001
1869
1870
1871
1872 004520
1873 004520 023706 000676
1874 004524 001403
1875 004526 010637 000700
1876 004532 104002
1877
1878
1879
1880 004534
1881 004534 005700
1882 004536 001401
1883 004540 104003
1884
1885 004542
1886 004542 005701
1887 004544 001401
1888 004546 104004
1889
1890 004550 005702
1891 004552 001401

```

```

*****
*TEST 14 TEST 'MOVRC' INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), DL .GT. SL
*****
*PROPER TERMINATION FOR THIS INSTRUCTION TEST IS A
*TRANSFER OF ALL BYTES FROM SOURCE TO DESTINATION
*AND 'FILL' CHARACTERS IN THE MSB OF THE DESTINATION,
*(R0-->R3 EQUAL ZERO, AND CONDITION CODES N,C=1 AND
*Z,V=0
*****
TST14: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      11          ;SOURCE LENGTH
      BUF1       ;SOURCE ADDRESS
      20          ;DESTINATION LENGTH
      BUF2+1     ;DESTINATION ADDRESS
      377        ;FILL CHARACTER
      JSR      PC,CLDST     ;CLEAR DESTINATION
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD     211
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      +CLN!CLC          ;CLEAR CONDITION CODES N&C
      +SEV!SEZ          ;SET CONDITION CODES V&Z
                          ;EXECUTE 'MOVE REVERSE CHARACTER'
      MOVRC
                          ;CHECK RESULTS
      JSR      PC,CKCC      ;CHECK PSW, GENERATE CONDITION CODES
      BEQ      64$
      ERROR    1
                          ;*****TEST 14 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT 'EXPPSW'
                          ;ACTUAL PSW IS STORED AT 'CCODES'
64$:
      CMP      @#SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$
      MOV      SP,@#BADR6  ;STORE BAD SP VALUE
      ERROR    2
                          ;*****TEST 14 - ERROR 2*****
                          ;STACK POINTER NOT RESTORED BY INSTRUCTION
                          ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
                          ;ERRONEOUS VALUE IS AT 'BADR6'
65$:
      TST      R0
      BEQ      1$
      ERROR    3
                          ;CHECK R0=ZERO
                          ;BR, IF ZERO
                          ;*****TEST 14 - ERROR 3*****
                          ;R0 SHOULD BE ZERO
                          ;CHECK OTHER GENERAL REGISTERS
1$:
      TST      R1
      BEQ      66$
      ERROR    4
                          ;TEST R1
                          ;BR, IF ZERO
                          ;*****TEST 14 - ERROR 4*****
                          ;R1 SHOULD BE ZERO
66$:
      TST      R2
      BEQ      67$
                          ;TEST R2
                          ;BR IF ZERO

```

```

1892 004554 104005          ERROR 5          :*****TEST 14 - ERROR 5*****
1893                                     :R2 SHOULD BE ZERO
1894 004556 005703          67$: TST R3          :TEST R3
1895 004560 001401          BEQ 68$          :BR, IF ZERO
1896 004562 104006          ERROR 6          :*****TEST 14 - ERROR 6*****
1897                                     :R3 SHOULD BE ZERO
1898 004564                                     68$:
1899 004564 026704 174064    CMP FILL,R4        :CHECK R4 UNCHANGED
1900 004570 001401          BEQ 69$          :BR IF OK
1901 004572 104007          ERROR 7          :*****TEST 14 - ERROR 7*****
1902                                     :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1903 004574 026705 174056    69$: CMP TABLE,R5   :CHECK R5 UNCHANGED
1904 004600 001401          BEQ 70$          :BR IF OK
1905 004602 104010          ERROR 10         :*****TEST 14 - ERROR 10*****
1906                                     :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
1907 004604                                     70$:
1908 004604 012700 016710    MOV #BUF2,R0       :POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1909 004610 105720          TSTB (R0)+        :TEST CONTENTS OF BOUNDARY
1910 004612 001401          BEQ 71$          :BR, IF STILL ZERO
1911 004614 104011          ERROR 11         :*****TEST 14 - ERROR 11*****
1912                                     :LOWER BOUNDARY OF DESTINATION CHANGED
1913                                     : SHOULD STILL EQUAL ZERO
1914 004616                                     71$:
1915 004616 016702 174026    MOV DSTIN,R2       :CALCULATE THE NUMBER OF 'FILL'
1916 004622 166702 174016    SUB SRCLN,R2       : CHARACTERS THAT SHOULD APPEAR IN DEST.
1917 004626 010201          MOV R2,R1          :STORE TRANSFER BYTE COUNT IN R1
1918 004630 122067 174020    T14E12: CMPB (R0)+,FILL :CHECK CHARACTERS IN DESTINATION
1919 004634 001401          BEQ +4           :BR IF OK
1920 004636 104012          ERROR 12         :*****TEST 14 - ERROR 12*****
1921                                     :COMPARE ERROR IN DESTINATION
1922                                     :R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1923
1924 004640 005301          DEC R1            :DECREMENT BYTE COUNT
1925 004642 001372          BNE T14E12       :BR, IF NOT FINISHED CHECKING
1926 004644 016702 173776    MOV SRCAD,R2       :POINT R2 TO SOURCE STRING
1927 004650 016701 173770    MOV SRCLN,R1       :STORE TRANSFER BYTE COUNT IN R1
1928 004654 122022          T14E13: CMPB (R0)+,(R2)+ :CHECK CHARACTERS IN DESTINATION
1929 004656 001401          BEQ +4           :BR IF OK
1930 004660 104013          ERROR 13         :*****TEST 14 - ERROR 13*****
1931                                     :COMPARE ERROR IN DESTINATION
1932                                     :R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1933
1934 004662 005301          DEC R1            :DECREMENT BYTE COUNT
1935 004664 001373          BNE T14E13       :BR, IF NOT FINISHED CHECKING
1936 004666 105720          TSTB (R0)+        :TEST CONTENTS OF DEST. UPPER BOUNDARY
1937 004670 001401          BEQ ENDT14       :BR, IF STILL ZERO
1938 004672 104014          ERROR 14         :*****TEST 14 - ERROR 14*****
1939                                     :UPPER BOUNDARY OF DEST. CHANGED
1940                                     : SHOULD STILL EQUAL ZERO
1941 004674          ENDT14:
1942
1943
1944
1945
1946
1947
:*****
:*TEST 15 TEST 'MOVRC' WITH DSTAD(LSB) .LT. SRCAD(LSB), SL .GT. DL, (TRUNCATED)
:*****
:*THIS TEST CHECKS 'MOVE REVERSE' WHEN THE LEAST SIGNIFICANT

```



```

1948      ;*BYTE OF THE SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION LSB
1949      ;*AND THE TRANSFER IS TRUNCATED.
1950      ;*THE RESULT IS THE MOST SIGNIFICANT BYTES ARE TRUNCATED, R0 = THE
1951      ;*NUMBER OF BYTES TRUNCATED, R1-->R3 EQUAL ZERO, AND ALL
1952      ;*CONDITION CODES CLEAR.
1953      ;*****
1954      ;*****
1955      TST15: SCOPE
1956      004674 000004      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
1957      004676 004567 007346      20      ;SOURCE LENGTH
1958      004702 000020      BUF2+20 ;SOURCE ADDRESS
1959      004704 016730      7      ;DESTINATION LENGTH
1960      004706 000007      BUF2+1  ;DESTINATION ADDRESS
1961      004710 016711      377    ;FILL CHARACTER
1962      004712 000377      JSR      PC,CLDST    ;CLEAR DESTINATION AREA
1963      004714 004767 007356      JSR      R5,GENSRC  ;GENERATE A SOURCE STRING
1964      004720 004567 007400      .WORD   20
1965      004724 000020      .WORD   BUF2+20
1966      004726 016730      JSR      R5,XPSW    ;STORE EXPECTED PSW VALUE
1967      004730 004567 007410      .WORD   200
1968      004734 000200      JSR      PC,GENR    ;SET UP GENERAL REGISTERS
1969      004736 004767 007242      SCC      ;SET ALL CONDITION CODES
1970      004742 000277
1971      004744 076031      MOVRC   ;EXECUTE 'MOVE REVERSE CHARACTER' INSTRUCTION
1972
1973      004746 004767 007416      JSR      PC,CKCC    ;CHECK PSW, GENERATE CONDITION CODES
1974      004752 001401      BEQ     64$
1975      004754 104001      ERROR   1
1976      ;*****TEST 15 - ERROR 1*****
1977      ;PSW ERROR
1978      ;EXPECTED PSW IS STORED AT 'EXPPSW'
1979      ;ACTUAL PSW IS STORED AT 'CCODES'
1980      004756 023706 000676      64$:   CMP     @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
1981      004762 001403      BEQ     65$         ;BR IF OK
1982      004764 010637 000700      MOV     SP,@#BADR6 ;STORE BAD SP VALUE
1983      004770 104002      ERROR   2         ;*****TEST 15 - ERROR 2*****
1984      ;STACK POINTER NOT RESTORED BY INSTRUCTION
1985      ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
1986      ;ERRONEOUS VALUE IS AT 'BADR6'
1987      004772 026704 173656      65$:   CMP     FILL,R4   ;CHECK R4 UNCHANGED
1988      004776 001401      BEQ     66$         ;BR IF OK
1989      005000 104003      ERROR   3         ;*****TEST 15 - ERROR 3*****
1990      ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
1991      005002 026705 173650      66$:   CMP     TABLE,R5 ;CHECK R5 UNCHANGED
1992      005006 001401      BEQ     67$         ;BR IF OK
1993      005010 104004      ERROR   4         ;*****TEST 15 - ERROR 4*****
1994      ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
1995
1996      005012 016705 173626      67$:   MOV     SRCLN,R5   ;CHECK R0=SOURCE LENGTH MINUS
1997      005016 166705 173626      SUB     DSTLN,R5   ; DESTINATION LENGTH
1998      005022 020005      CMP     R0,R5
1999      005024 001401      BEQ     68$
2000      005026 104005      ERROR   5
2001      ;*****TEST 15 - ERROR 5*****
2002      ;R0 NOT EQUAL TO (SRCLN-DSTLN)
2003      005030      68$:   ;CHECK OTHER GENERAL REGISTERS

```

```

2004 C05030 005701          TST      R1          ;TEST R1
2005 005032 001401          BEQ      69$         ;BR, IF ZERO
2006 005034 104006          ERROR    6           ;*****TEST 15 - ERROR 6*****
2007                                     ;R1 SHOULD BE ZERO
2008 005036 005702          69$:  TST      R2          ;TEST R2
2009 005040 001401          BEQ      70$         ;BR IF ZERO
2010 005042 104007          ERROR    7           ;*****TEST 15 - ERROR 7*****
2011                                     ;R2 SHOULD BE ZERO
2012 005044 005703          70$:  TST      R3          ;TEST R3
2013 005046 001401          BEQ      71$         ;BR, IF ZERO
2014 005050 104010          ERROR    10          ;*****TEST 15 - ERROR 10*****
2015                                     ;R3 SHOULD BE ZERO
2016 005052          71$:
2017                                     ;VERIFY DESTINATION AREA
2018 005052 012700 016710    MOV      #BUF2,R0    ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2019 005056 105720          TSTB     (R0)+       ;TEST CONTENTS OF BOUNDARY
2020 005060 001401          BEQ      72$         ;BR, IF STILL ZERO
2021 005062 104011          ERROR    11          ;*****TEST 15 - ERROR 11*****
2022                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
2023                                     ; SHOULD STILL EQUAL ZERO
2024 005064          72$:
2025 005064 016702 173554    MOV      SRCLN,R2    ;CALCULATE ADDRESS OF MSB OF
2026 005070 166702 173554    SUB      DSTLN,R2    ; THE SOURCE TO BE MOVED
2027 005074 066702 173546    ADD      SRCAD,R2
2028 005100 016701 173544    MOV      DSTLN,R1
2029 005104 122022          T15E12: CMPB     (R0)+,(R2)+ ;STORE TRANSFER BYTE COUNT IN R1
2030 005106 001401          BEQ      +4          ;CHECK CHARACTERS IN DESTINATION
2031 005110 104012          ERROR    12          ;BR IF OK
2032                                     ;*****TEST 15 - ERROR 12*****
2033                                     ;COMPARE ERROR IN DESTINATION
2034                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2035 005112 005301          DEC      R1          ;DECREMENT BYTE COUNT
2036 005114 001373          BNE     T15E12      ;BR, IF NOT FINISHED CHECKING
2037 005116 105720          TSTB     (R0)+       ;TEST CONTENTS OF DEST. UPPER BOUNDARY
2038 005120 001401          BEQ      ENDT15     ;BR, IF STILL ZERO
2039 005122 104013          ERROR    13          ;*****TEST 15 - ERROR 13*****
2040                                     ;UPPER BOUNDARY OF DEST. CHANGED
2041                                     ; SHOULD STILL EQUAL ZERO
2042 005124          ENDT15:
2043
2044
2045
2046
2047                                     ;*****
2048                                     ;*TEST 16      TEST 'MOVRC' WITH DSTAD(LSB) .LT. SRCAD(LSB) & SRCLN .GT. DSTLN, (FILL)
2049                                     ;*****
2050                                     ;*THIS TEST CHECKS 'MOVE REVERSE' WHEN THE LEAST SIGNIFICANT
2051                                     ;*BYTE OF THE SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION LSB
2052                                     ;*AND ALL BYTES ARE TRANSFERED.
2053                                     ;*THE RESULT IS THE MSB OF THE DESTINATION ARE FILLS, R0-->R3 EQUAL
2054                                     ;*ZERO, AND CONDITION CODES N,C=1 & Z,V=0.
2055                                     ;*****
2056 005124 000004          TST16: SCOPE
2057 005126 004567 007116    JSR      R5,PREP    ;SET UP INSTRUCTION ARGUMENTS
2058 005132 000010          10
2059 005134 016730          BUF2+20          ;SOURCE LENGTH
                                     ;SOURCE ADDRESS

```


2060	005136	000015			15	:DESTINATION LENGTH
2061	005140	016711			BUF2+1	:DESTINATION ADDRESS
2062	005142	000377			377	:FILL CHARACTER
2063	005144	004767	007126	JSR	PC,CLDST	:CLEAR DESTINATION AREA
2064	005150	004567	007150	JSR	R5,GENSRC	:GENERATE SOURCE STRING
2065	005154	000010		.WORD	10	
2066	005156	016730		.WORD	BUF2+20	
2067	005160	004567	007160	JSR	R5,XPSW	:STORE EXPECTED PSW VALUE
2068	005164	000211		.WORD	211	
2069	005166	004767	007012	JSR	PC,GENR	:SET UP GENERAL REGISTERS
2070	005172	000251		+CLN!CLC		
2071	005174	000266		+SEV!SEZ		
2072						
2073	005176	076031		MOVRC		:EXECUTE 'MOVE REVERSE CHARACTER'
2074						
2075	005200	004767	007164	JSR	PC,CKCC	:CHECK PSW, GENERATE CONDITION CODES
2076	005204	001401		BEQ	64\$	
2077	005206	104001		ERROR	1	:*****TEST 16 - ERROR 1*****
2078						:PSW ERROR
2079						:EXPECTED PSW IS STORED AT 'EXPPSW'
2080						:ACTUAL PSW IS STORED AT 'CCODES'
2081	005210				64\$:	
2082	005210	023706	000676	CMP	@#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
2083	005214	001403		BEQ	65\$:BR IF OK
2084	005216	010637	000700	MOV	SP,@#BADR6	:STORE BAD SP VALUE
2085	005222	104002		ERROR	2	:*****TEST 16 - ERROR 2*****
2086						:STACK POINTER NOT RESTORED BY INSTRUCTION
2087						:EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
2088						:ERRONEOUS VALUE IS AT 'BADR6'
2089	005224				65\$:	
2090	005224	005700		TST	R0	
2091	005226	001401		BEQ	3\$	
2092	005230	104003		ERROR	3	:*****TEST 16 - ERROR 3*****
2093						:R0 SHOULD BE ZERO
2094	005232				3\$:	
2095	005232	005701		TST	R1	:TEST R1
2096	005234	001401		BEQ	66\$:BR, IF ZERO
2097	005236	104004		ERROR	4	:*****TEST 16 - ERROR 4*****
2098						:R1 SHOULD BE ZERO
2099	005240	005702		TST	R2	:TEST R2
2100	005242	001401		BEQ	67\$:BR IF ZERO
2101	005244	104005		ERROR	5	:*****TEST 16 - ERROR 5*****
2102						:R2 SHOULD BE ZERO
2103	005246	005703		TST	R3	:TEST R3
2104	005250	001401		BEQ	68\$:BR, IF ZERO
2105	005252	104006		ERROR	6	:*****TEST 16 - ERROR 6*****
2106						:R3 SHOULD BE ZERO
2107	005254				68\$:	
2108	005254	026704	173374	CMP	FILL,R4	:CHECK R4 UNCHANGED
2109	005260	001401		BEQ	69\$:BR IF OK
2110	005262	104007		ERROR	7	:*****TEST 16 - ERROR 7*****
2111						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
2112	005264	026705	173366	CMP	TABLE,R5	:CHECK R5 UNCHANGED
2113	005270	001401		BEQ	70\$:BR IF OK
2114	005272	104010		ERROR	10	:*****TEST 16 - ERROR 10*****
2115						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'

```

2116 005274          70$:      MOV      #BUF2,R0      ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2117 005274 012700 016710    TSTB     (R0)+      ;TEST CONTENTS OF BOUNDARY
2118 005300 105720          BEQ      71$        ;BR, IF STILL ZERO
2119 005302 001401          ERROR    11          ;*****TEST 16 - ERROR 11*****
2120 005304 104011          ;LOWER BOUNDARY OF DESTINATION CHANGED
2121 ; SHOULD STILL EQUAL ZERO
2122
2123 005306          71$:      MOV      DSTLN,R2      ;CALCULATE 'FILL' COUNT
2124 005306 016702 173336    SUB      SRCLN,R2
2125 005312 166702 173326    MOV      R2,R1      ;STORE TRANSFER BYTE COUNT IN R1
2126 005316 010201          T16E12: CMPB     (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
2127 005320 122067 173330    BEQ      .+4        ;BR IF OK
2128 005324 001401          ERROR    12          ;*****TEST 16 - ERROR 12*****
2129 005326 104012          ;COMPARE ERROR IN DESTINATION
2130 ;RO CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2131
2132
2133 005330 005301          DEC      R1          ;DECREMENT BYTE COUNT
2134 005332 001372          BNE     T16E12      ;BR, IF NOT FINISHED CHECKING
2135 005334 016702 173306    MOV      SRCAD,R2
2136 005340 016701 173300    MOV      SRCLN,R1
2137 005344 122022          T16E13: CMPB     (R0)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
2138 005346 001401          BEQ      .+4        ;BR IF OK
2139 005350 104013          ERROR    13          ;*****TEST 16 - ERROR 13*****
2140 ;COMPARE ERROR IN DESTINATION
2141 ;RO CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2142
2143 005352 005301          DEC      R1          ;DECREMENT BYTE COUNT
2144 005354 001373          BNE     T16E13      ;BR, IF NOT FINISHED CHECKING
2145 005356 105720          TSTB     (R0)+      ;TEST CONTENTS OF DEST. UPPER BOUNDARY
2146 005360 001401          BEQ      ENDT16     ;BR, IF STILL ZERO
2147 005362 104014          ERROR    14          ;*****TEST 16 - ERROR 14*****
2148 ;UPPER BOUNDARY OF DEST. CHANGED
2149 ; SHOULD STILL EQUAL ZERO
2150 005364          ENDT16:

```

```

2151
2152
2153 ;*****
2154 ;*TEST 17      TEST INTERRUPTABILITY OF 'MOVRC' INSTRUCTION
2155 ;*****
2156 ;*THIS TEST INTERRUPTS THE EXECUTION OF THE 'MOVRC'
2157 ;*INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
2158 ;*INTERRUPT, AND VERIFIES THE RESULTS. THE PROPER
2159 ;*RESULT IS ALL BYTES MOVED TO THE DESTINATION,
2160 ;*R0-->R3 EQUAL ZERO, AND CONDITION CODES N,C=1
2161 ;*AND Z,V=0.
2162 ;*****
2163 ;*****
2164
2165
2166 005364 000004          TST17: SCOPE
2167 005366 004767 007020    JSR     PC,SKPINT  ;SET FLAG -- TEMP, WHICH INDICATES IF
2168 ;NEXT TEST IS TO BE EXECUTED NEXT.
2169 005372 005767 173306    TST     TEMP       ;CHECK IF -TEMP- IS SET
2170 005376 001156          BNE     TST20      ;TEMP IS SET, GO TO NEXT TEST
2171 005400 004567 006644    JSR     R5,PREP    ;SET UP INSTRUCTION ARGUMENTS

```



```

2228 005606 001401          BEQ    68$      ;BR, IF ZERO
2229 005610 104006          ERROR   6      ;*****TEST 17 - ERROR 6*****
2230                                     ;R3 SHOULD BE ZERO
2231 005612                                     68$:
2232 005612 026704 173036    CMP    FILL,R4  ;CHECK R4 UNCHANGED
2233 005616 001401          BEQ    69$      ;BR IF OK
2234 005620 104007          ERROR   7      ;*****TEST 17 - ERROR 7*****
2235                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
2236 005622 026705 173030    69$:  CMP    TABLE,R5 ;CHECK R5 UNCHANGED
2237 005626 001401          BEQ    70$      ;BR IF OK
2238 005630 104010          ERROR   10     ;*****TEST 17 - ERROR 10*****
2239                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
2240 005632                                     70$:
2241 005632 012700 016710    MOV    #BUF2,R0 ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2242 005636 105720          TSTB   (R0)+    ;TEST CONTENTS OF BOUNDARY
2243 005640 001401          BEQ    71$      ;BR, IF STILL ZERO
2244 005642 104011          ERROR   11     ;*****TEST 17 - ERROR 11*****
2245                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
2246                                     ; SHOULD STILL EQUAL ZERO
2247 005644                                     71$:
2248 005644 016702 173000    MOV    DSTLN,R2 ;CALCULATE THE NUMBER OF FILL
2249 005650 166702 172770    SUB    SRCLN,R2 ;CHARACTERS THAT SHOULD APPEAR IN DEST.
2250 005654 010201          MOV    R2,R1    ;STORE TRANSFER BYTE COUNT IN R1
2251 005656 122067 172772    T17E12: CMPB   (R0)+,FILL ;CHECK CHARACTERS IN DESTINATION
2252 005662 001401          BEQ    +4       ;BR IF OK
2253 005664 104012          ERROR   12     ;*****TEST 17 - ERROR 12*****
2254                                     ;COMPARE ERROR IN DESTINATION
2255                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2256
2257 005666 005301          DEC    R1       ;DECREMENT BYTE COUNT
2258 005670 001372          BNE    T17E12  ;BR, IF NOT FINISHED CHECKING
2259 005672 016702 172750    MOV    SRCAD,R2 ;POINT R2 TO SOURCE STRING
2260 005676 016701 172742    MOV    SRCLN,R1 ;STORE TRANSFER BYTE COUNT IN R1
2261 005702 122022          TE13:  CMPB   (R0)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
2262 005704 001401          BEQ    +4       ;BR IF OK
2263 005706 104013          ERROR   13     ;*****TEST 17 - ERROR 13*****
2264                                     ;COMPARE ERROR IN DESTINATION
2265                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2266
2267 005710 005301          DEC    R1       ;DECREMENT BYTE COUNT
2268 005712 001373          BNE    TE13    ;BR, IF NOT FINISHED CHECKING
2269 005714 105720          TSTB   (R0)+    ;TEST CONTENTS OF DEST. UPPER BOUNDARY
2270 005716 001401          BEQ    ENDT17  ;BR, IF STILL ZERO
2271 005720 104014          ERROR   14     ;*****TEST 17 - ERROR 14*****
2272                                     ;UPPER BOUNDARY OF DEST. CHANGED
2273                                     ; SHOULD STILL EQUAL ZERO
2274 005722          ENDT17:
2275 005722 106427 000200    40$:  MTPS   #200    ;RESTORE PSW TO PRIORITY 7
2276 005726 016777 172734 172730  MOV    TPSW,@TVECT ;RESTORE TRAP CATCHER
2277
2278
2279
2280
2281
2282
2283
;*****
;*TEST 20 TEST 'CMPC' WITH SOURCE1 & SOURCE2 LENGTHS = 0
;*****
;*THIS TEST VERIFIES THAT 'COMPARE CHARACTER' INDICATES EQUAL

```



```

2284
2285
2286
2287
2288
2289 005734 000004
2290 005736 004567 006306
2291 005742 000000
2292 005744 177777
2293 005746 000000
2294 005750 177777
2295 005752 000377
2296 005754 004567 006364
2297 005760 000204
2298 005762 004767 006216
2299 005766 000277
2300 005770 000244
2301
2302 005772 076044
2303
2304 005774 004767 006370
2305 006000 001401
2306 006002 104001
2307
2308
2309
2310 006004
2311 006004 023706 000676
2312 006010 001403
2313 006012 010637 000700
2314 006016 104002
2315
2316
2317
2318 006020
2319 006020 026700 172620
2320 006024 001401
2321 006026 104003
2322
2323 006030 026701 172612
2324 006034 001401
2325 006036 104004
2326
2327 006040 026702 172604
2328 006044 001401
2329 006046 104005
2330
2331 006050 026703 172576
2332 006054 001401
2333 006056 104006
2334
2335 006060
2336 006060 026704 172570
2337 006064 001401
2338 006066 104007
2339

: *STRINGS WITH SOURCE LENGTHS EQUAL TO ZERO.
: *THE RESULT IS R0-->R3 ARE UNCHANGED, AND ALL CONDITION
: *CODES CLEAR EXCEPT Z=1.
: *****
: *****
TST20: SCOPE
        JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
        0          ;SOURCE1 LENGTH
        NXM       ;SOURCE1 ADDRESS
        0          ;SOURCE2 LENGTH
        NXM       ;SOURCE2 ADDRESS
        377       ;FILL CHARACTER
        JSR      R5,XPSW
        .WORD     204
        JSR      PC,GENR        ;SET UP GENERAL REGISTERS
        SCC          ;SET ALL CONDITION CODES, EXCEPT
        CLZ         ;CLEAR Z
        CMPC      ;EXECUTE "CMPC" INSTRUCTION
        JSR      PC,CKCC        ;CHECK PSW, GENERATE CONDITION CODES
        BEQ      64$
        ERROR    1          ;*****TEST 20 - ERROR 1*****
        ;PSW ERROR
        ;EXPECTED PSW IS STORED AT 'EXPPSW'
        ;ACTUAL PSW IS STORED AT 'CCODES'
64$:
        CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
        BEQ      65$           ;BR IF OK
        MOV     SP,@#BADR6      ;STORE BAD SP VALUE
        ERROR    2          ;*****TEST 20 - ERROR 2*****
        ;STACK POINTER NOT RESTORED BY INSTRUCTION
        ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
        ;ERRONEOUS VALUE IS AT 'BADR6'
65$:
        CMP      SRCLN,R0       ;CHECK R0 UNCHANGED
        BEQ      1$
        ERROR    3          ;*****TEST 20 - ERROR 3*****
        ;SRC1 LENGTH ERROR
        ;CHECK R1 UNCHANGED
1$:
        CMP      SRCAD,R1
        BEQ      2$
        ERROR    4          ;*****TEST 20 - ERROR 4*****
        ;SRC1 ADDRESS ERROR
        ;CHECK R2 UNCHANGED
2$:
        CMP      DSTLN,R2
        BEQ      3$
        ERROR    5          ;*****TEST 20 - ERROR 5*****
        ;SRC2 LENGTH ERROR
        ;CHECK R3 UNCHANGED
3$:
        CMP      DSTAD,R3
        BEQ      4$
        ERROR    6          ;*****TEST 20 - ERROR 6*****
        ;SRC2 ADDRESS ERROR
4$:
        CMP      FILL,R4       ;CHECK R4 UNCHANGED
        BEQ      66$
        ERROR    7          ;*****TEST 20 - ERROR 7*****
        ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'

```

```

2340 006070 026705 172562 66$: CMP TABLE,R5 ;CHECK R5 UNCHANGED
2341 006074 001401 BEQ 67$ ;BR IF OK
2342 006076 104010 ERROR 10 ;*****TEST 20 - ERROR 10*****
2343 ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```

2344 006100 67\$:

```

2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358 006100 000004
2359
2360 006102 004567 006142 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
2361 006106 000000 0 ;SOURCE1 LENGTH
2362 006110 177777 NXM ;SOURCE1 ADDRESS
2363 006112 000004 4 ;SOURCE2 LENGTH
2364 006114 016710 BUF2 ;SOURCE2 ADDRESS
2365 006116 000377 377 ;FILL CHARACTER
2366 006120 012700 016710 MOV #BUF2,R0 ;GENERATE SOURCE2 STRING STARTING
2367 006124 012720 177777 MOV #177777,(R0)+ ;WITH TWO FILL BYTES
2368 006130 005020 CLR (R0)+ ;THEN TWO NON-FILL BYTES
2369 006132 004567 006206 JSR R5,XPSW ;STORE EXPECTED PSW VALUE
2370 006136 000210 .WORD 210
2371 006140 004767 006040 JSR PC,GENR ;SET UP GENERAL REGISTERS
2372 006144 000277 SCC ;SET ALL CONDITION CODES
2373
2374 006146 076044 CMPC ;EXECUTE "CMPC" INSTRUCTION
2375
2376 006150 004767 006214 JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES
2377 006154 001401 BEQ 64$
2378 006156 104001 ERROR 1 ;*****TEST 21 - ERROR 1*****
2379 ;PSW ERROR
2380 ;EXPECTED PSW IS STORED AT "EXPPSW"
2381 ;ACTUAL PSW IS STORED AT "CCODES"

```

```

2382 006160 64$:
2383 006160 023706 000676 CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2384 006164 001403 BEQ 65$ ;BR IF OK
2385 006166 010637 000700 MOV SP,@#BADR6 ;STORE BAD SP VALUE
2386 006172 104002 ERROR 2 ;*****TEST 21 - ERROR 2*****
2387 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2388 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2389 ;ERRONEOUS VALUE IS AT "BADR6"

```

```

2390 006174 65$:
2391 006174 005700 TST R0 ;CHECK R0=0
2392 006176 001401 BEQ 1$
2393 006200 104003 ERROR 3 ;*****TEST 21 - ERROR 3*****
2394 ;SRC1 LENGTH ERROR
2395 006202 026701 172440 1$: CMP SRCAD,R1 ;CHECK R1 UNCHANGED

```



```

2396 006206 001401 BEQ 2$
2397 006210 104004 ERROR 4 ;*****TEST 21 - ERROR 4*****
2398 ;SRC1 ADDRESS ERROR
2399 006212 022702 000002 2$: CMP #2,R2 ;CHECK R2=2
2400 006216 001401 BEQ 3$
2401 006220 104005 ERROR 5 ;*****TEST 21 - ERROR 5*****
2402 ;SRC2 LENGTH ERROR
2403 006222 022703 016712 3$: CMP #BUF2+2,R3 ;CHECK R3=BUF2+2
2404 006226 001401 BEQ 4$
2405 006230 104006 ERROR 6 ;*****TEST 21 - ERROR 6*****
2406 ;SRC2 ADDRESS ERROR
2407 006232 4$:
2408 006232 026704 172416 CMP FILL,R4 ;CHECK R4 UNCHANGED
2409 006236 001401 BEQ 66$ ;BR IF OK
2410 006240 104007 ERROR 7 ;*****TEST 21 - ERROR 7*****
2411 ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
2412 006242 026705 172410 66$: CMP TABLE,R5 ;CHECK R5 UNCHANGED
2413 006246 001401 BEQ 67$ ;BR IF OK
2414 006250 104010 ERROR 10 ;*****TEST 21 - ERROR 10*****
2415 ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
2416 006252 67$:

```

```

2417
2418
2419
2420 ;*****
2421 ;*TEST 22 TEST "CMPC" WITH S1L .LT. S2L AND SRC1.GT.SRC2, NON-COMPARE IN SOURCE
2422 ;*****
2423 ;*THIS TEST VERIFIES THAT "CMPC" RETURNS THE CORRECT SUBSTRING IDENTIFIERS
2424 ;*WHEN SOURCE1 STRING AND SOURCE2 STRING DO NOT COMPARE.
2425 ;*THE RESULT IS R0 EQUALS SRC1 SUBSTRING LENGTH, R1 EQUALS
2426 ;*SRC1 SUBSTRING POINTER, R2 EQUALS SRC2 SUBSTRING LENGTH, R3 EQUALS
2427 ;*SRC2 SUBSTRING POINTER, AND ALL CONDITION CODES CLEAR EXCEPT N=1.
2428 ;*****

```

```

2429 006252 000004 TST22: SCOPE
2430 006254 004567 005770 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
2431 006260 000007 7 ;SOURCE1 LENGTH
2432 006262 016310 BUF1 ;SOURCE1 ADDRESS
2433 006264 000012 12 ;SOURCE2 LENGTH
2434 006266 016710 BUF2 ;SOURCE2 ADDRESS
2435 006270 000377 377 ;FILL CHARACTER
2436 006272 004767 006000 JSR PC,CLDST ;CLEAR SOURCE2 AREA
2437 006276 004567 006022 JSR R5,GENSRC ;GENERATE SOURCE2 STRING
2438 006302 000012 .WORD 12
2439 006304 016710 .WORD BUF2
2440 006306 105037 016715 CLRB #BUF2+5 ;CREATE A NON-COMPARE CHARACTER IN SRC2 STRING
2441 006312 004567 006026 JSR R5,XPSW ;STORE EXPECTED PSW VALUE
2442 006316 000210 .WORD 210
2443 006320 004767 005660 JSR PC,GENR ;SET UP GENERAL REGISTERS
2444 006324 000277 SCC ;SET ALL CONDITION CODES
2445
2446 006326 076044 CMPC ;EXECUTE "CMPC" INSTRUCTION
2447
2448 006330 004767 006034 JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES
2449 006334 001401 BEQ 64$
2450 006336 104001 ERROR 1 ;*****TEST 22 - ERROR 1*****
2451 ;PSW ERROR

```

```
2452                                     :EXPECTED PSW IS STORED AT 'EXPPSW'  
2453                                     :ACTUAL PSW IS STORED AT 'CCODES'  
2454 006340 64$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
2455 006340 023706 000676 BEQ 65$ ;BR IF OK  
2456 006344 001403 MOV SP,@#BADR6 ;STORE BAD SP VALUE  
2457 006346 010637 000700 ERROR 2 ;*****TEST 22 - ERROR 2*****  
2458 006352 104002 ;STACK POINTER NOT RESTORED BY INSTRUCTION  
2459 ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'  
2460 ;ERRONEOUS VALUE IS AT 'BADR6'  
2461  
2462 006354 65$: CMP FILL,R4 ;CHECK R4 UNCHANGED  
2463 006354 026704 172274 BEQ 66$ ;BR IF OK  
2464 006360 001401 ERROR 3 ;*****TEST 22 - ERROR 3*****  
2465 006362 104003 ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'  
2466 ;CHECK R5 UNCHANGED  
2467 006364 026705 172266 66$: CMP TABLE,R5 ;CHECK R5 UNCHANGED  
2468 006370 001401 BEQ 67$ ;BR IF OK  
2469 006372 104004 ERROR 4 ;*****TEST 22 - ERROR 4*****  
2470 ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'  
2471  
2472 006374 67$: CLR TEMP ;POINT R4 TO SOURCE1 ADDRESS  
2473 006374 005067 172304 1$: MOV SRCAD,R4 ;POINT R5 TO SOURCE2 ADDRESS  
2474 006400 016704 172242 MOV DSTAD,R5 ;COMPARE SOURCES  
2475 006410 121415 2$: CMPB (R4),(R5) ;BR IF NOT EQUAL  
2476 006412 001004 BNE 3$  
2477 006414 005267 172264 INC TEMP ;POINT STRINGS TO NEXT BYTES  
2478 006420 122425 CMPB (R4)+,(R5)+  
2479 006422 000772 BR 2$  
2480 006424 3$: CMP R4,R1 ;CHECK ADDRESS & LENGTH DESCRIPTORS  
2481 006424 020401 CMP 4$ ;CHECK R1, SOURCE1 ADDRESS  
2482 006426 001401 BEQ 5$ ;BR IF OK  
2483 006430 104005 ERROR 5 ;*****TEST 22 - ERROR 5*****  
2484 ;SOURCE1 ADDRESS ERROR  
2485 ;R4 CONTAINS THE EXPECTED ADDRESS POINTER  
2486 ;R1 CONTAINS THE BAD VALUE  
2487 006432 020503 4$: CMP R5,R3 ;CHECK R3,SOURCE2 ADDRESS  
2488 006434 001401 BEQ 5$ ;BR IF OK  
2489 006436 104006 ERROR 6 ;*****TEST 22 - ERROR 6*****  
2490 ;SOURCE2 ADDRESS ERROR  
2491 ;R5 CONTAINS THE EXPECTED ADDRESS POINTER  
2492 ;R3 CONTAINS THE BAD VALUE  
2493 006440 016704 172200 5$: MOV SRCLN,R4 ;CALCULATE LENGTH OF SRC1 REMAINDER  
2494 006444 166704 172234 SUB TEMP,R4  
2495 006450 020400 CMP R4,R0 ;CHECK R0, SOURCE1 LENGTH  
2496 006452 001401 BEQ 6$ ;BR IF OK  
2497 006454 104007 ERROR 7 ;*****TEST 22 - ERROR 7*****  
2498 ;SOURCE1 LENGTH ERROR  
2499 ;R4 CONTAINS THE EXPECTED LENGTH DESCRIPTOR  
2500 ;R0 CONTAINS THE BAD VALUE  
2501 006456 016704 172166 6$: MOV DSTLN,R4 ;CALCULATE LENGTH OF SRC2 REMAINDER  
2502 006462 166704 172216 SUB TEMP,R4  
2503 006466 020402 CMP R4,R2 ;CHECK R2,SOURCE2 LENGTH  
2504 006470 001401 BEQ 7$ ;BR IF OK  
2505 006472 104010 ERROR 10 ;*****TEST 22 - ERROR 10*****  
2506 ;SOURCE2 LENGTH ERROR  
2507 ;R4 CONTAINS THE EXPECTED LENGTH DESCRIPTOR
```


:R2 CONTAINS THE BAD VALUE

```

2508
2509 006474 000240 7$: NOP
2510
2511 ::*****
2512 :*TEST 23 TEST "CMPC" FOR S1L=S2L, SOURCE1=SOURCE2 STRINGS
2513 :*****
2514 :*PROPER TERMINATION OF THIS INSTRUCTION TEST IS THE
2515 :*INDICATION THAT SOURCE1 STRING EQUALS SOURCE2 STRING:
2516 :*R0 & R2 EQUAL ZERO, R1 & R3 EQUAL THE ADDRESS+1 OF THE
2517 :*LSB OF THE SOURCE1 & SOURCE2 STRINGS RESPECTIVELY,
2518 :*AND ALL CONDITION CODES CLEAR, EXCEPT Z=1
2519 :*****
2520 :*****
2521 006476 000004 TST23: SCOPE
2522 006500 004567 005544 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
2523 006504 000007 JSR 7 ;SOURCE1 LENGTH
2524 006506 016310 BUF1 ;SOURCE1 ADDRESS
2525 006510 000007 7 ;SOURCE2 LENGTH
2526 006512 016310 BUF1 ;SOURCE2 ADDRESS
2527 006514 000377 377 ;FILL CHARACTER
2528 006516 004567 005622 JSR R5,XPSW ;STORE EXPECTED PSW VALUE
2529 006522 000204 .WORD 204
2530 006524 004767 005454 JSR PC,GENR ;SET UP GENERAL REGISTERS
2531 006530 000277 SCC ;SET ALL CONDITION CODES, EXCEPT
2532 006532 000244 CLZ ; CLEAR 'Z'
2533 ;EXECUTE "CMPC" INSTRUCTION
2534 006534 076044 CMPC ;CHECK RESULTS
2535 ;CHECK PSW, GENERATE CONDITION CODES
2536 006536 004767 005626 JSR PC,CKCC
2537 006542 001401 BEQ 64$
2538 006544 104001 ERROR 1 ;*****TEST 23 - ERROR 1*****
2539 ;PSW ERROR
2540 ;EXPECTED PSW IS STORED AT 'EXPPSW'
2541 ;ACTUAL PSW IS STORED AT 'CCODES'
2542 006546 64$:
2543 006546 023706 000676 CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2544 006552 001403 BEQ 65$ ;BR, IF OK
2545 006554 010637 000700 MOV SP,@#BADR6 ;STORE BAD SP VALUE
2546 006560 1040C2 ERROR 2 ;*****TEST 23 - ERROR 2*****
2547 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2548 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2549 ;ERRONEOUS VALUE IS AT "BADR6"
2550 006562 65$:
2551 006562 105700 TSTB R0 ;CHECK R0
2552 006564 001401 BEQ 1$ ;BR, IF ZERO
2553 006566 104003 ERROR 3 ;*****TEST 23 - ERROR 3*****
2554 ;R0 SHOULD BE ZERO
2555 006570 105702 1$: TSTB R2 ;CHECK R2
2556 006572 001401 BEQ 2$ ;BR, IF ZERO
2557 006574 104004 ERROR 4 ;*****TEST 23 - ERROR 4*****
2558 ;R2 SHOULD BE ZERO
2559 006576 016700 172044 2$: MOV SRCAD,R0 ;CALCULATE ADDRESS+1 OF LSB
2560 006602 066700 172036 ADD SRCLN,R0
2561 006606 020001 CMP R0,R1 ;CHECK R1
2562 006610 001401 BEQ 3$ ;BR, IF EQUAL TO ADDRESS+1 OF LSB
2563 006612 104005 ERROR 5 ;*****TEST 23 - ERROR 5*****

```

```

2564                                     ;R1 SHOULD EQUAL (SRCAD)+(SRCLN)
2565 006614 020003 3$: CMP R0,R3 ;CHECK R3
2566 006616 001401 BEQ 40$ ;BR, IF EQUAL TO ADDRESS+1 OF LSB
2567 006620 104006 ERROR 6 ;*****TEST 23 - ERROR 6*****
2568                                     ;R3 SHOULD EQUAL (DSTAD)+(DSTLN)
2569 006622 40$: CMP FILL,R4 ;CHECK R4 UNCHANGED
2570 006622 026704 172026 BEQ 66$ ;BR IF OK
2571 006626 001401 ERROR 7 ;*****TEST 23 - ERROR 7*****
2572 006630 104007 ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2573                                     ;CHECK R5 UNCHANGED
2574 006632 026705 172020 66$: CMP TABLE,R5 ;BR IF OK
2575 006636 001401 BEQ 67$ ;*****TEST 23 - ERROR 10*****
2576 006640 104010 ERROR 10 ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2577
2578 006642 67$:
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589

```

```

:*****
:*TEST 24 TEST "CMPC" WITH SOURCE2=0 & SOURCE1 CONTAINS ALL FILLS
:*****
:*THIS TEST VERIFIES THAT "CMPC" COMPARES SOURCE1 WITH THE FILL
:*CHARACTER WHEN SOURCE2 LENGTH IS ZERO.
:*RESULT IS EQUAL STINGS WHEN SOURCE1 IS ALL FILLS.
:*****

```

```

2590 006642 000004 TST24: SCOPE
2591
2592 006644 004567 005400 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
2593 006650 000004 4 ;SOURCE1 LENGTH
2594 006652 016710 BUF2 ;SOURCE1 ADDRESS
2595 006654 000000 0 ;SOURCE2 LENGTH
2596 006656 177777 NXM ;SOURCE2 ADDRESS
2597 006660 000377 377 ;FILL CHARACTER
2598 006662 012700 016710 MOV #BUF2,R0 ;GENERATE SOURCE1 STRING WITH
2599 006666 012710 177777 MOV #177777,(R0) ;ALL FILLS
2600 006672 012010 MOV (R0)+,(R0)
2601 006674 004567 005444 JSR R5,XPSW ;STORE EXPECTED PSW VALUE
2602 006700 000204 .WORD 204
2603 006702 004767 005276 JSR PC,GENR ;SET UP GENERAL REGISTERS
2604 006706 000277 SCC ;SET ALL CONDITION CODES, EXCEPT
2605 006710 000244 CLZ ;CLEAR Z
2606
2607 006712 076044 CMPC ;EXECUTE "CMPC" INSTRUCTION
2608
2609 006714 004767 005450 JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES
2610 006720 001401 BEQ 64$
2611 006722 104001 ERROR 1 ;*****TEST 24 - ERROR 1*****
2612                                     ;PSW ERROR
2613                                     ;EXPECTED PSW IS STORED AT "EXPPSW"
2614                                     ;ACTUAL PSW IS STORED AT "CCODES"
2615
2616 006724 023706 000676 64$: CMP @SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2617 006730 001403 BEQ 65$ ;BR IF OK
2618 006732 010637 000700 MOV SP,@BADR6 ;STORE BAD SP VALUE
2619 006736 104002 ERROR 2 ;*****TEST 24 - ERROR 2*****

```



```

2620                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
2621                                     ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2622                                     ;ERRONEOUS VALUE IS AT "BADR6"
2623 006740                               65$: TST      R0
2624 006740 005700                       BEQ      1$
2625 006742 001401                       ERROR   3
2626 006744 104003                       ;*****TEST 24 - ERROR 3*****
2627                                     ;SOURCE1 LENGTH ERROR
2628                                     ;R0 SHOULD BE ZERO
2629 006746 016700 171674                1$:  MOV     SRCAD,R0
2630 006752 066700 171666                ADD     SRCLN,R0
2631 006756 020001                       CMP     R0,R1
2632 006760 001401                       BEQ     2$
2633 006762 104004                       ERROR   4
2634                                     ;*****TEST 24 - ERROR 4*****
2635                                     ;SOURCE1 ADDRESS ERROR
2636                                     ;R0 CONTAINS THE EXPECTED ADDRESS POINTER
2637 006764 005702                       2$:  TST     R2
2638 006766 001401                       BEQ     3$
2639 006770 104005                       ERROR   5
2640                                     ;*****TEST 24 - ERROR 5*****
2641                                     ;SOURCE2 LENGTH ERROR
2642                                     ;R2 SHOULD BE ZERO
2643 006772 016702 171654                3$:  MOV     DSTAD,R2
2644 006776 066702 171646                ADD     DSTLN,R2
2645 007002 020203                       CMP     R2,R3
2646 007004 001401                       BEQ     4$
2647 007006 104006                       ERROR   6
2648                                     ;*****TEST 24 - ERROR 6*****
2649                                     ;SOURCE2 ADDRESS ERROR
2650                                     ;R2 CONTAINS THE EXPECTED ADDRESS POINTER
2651                                     ;R3 CONTAINS THE BAD VALUE
2652 007010                               4$:  CMP     FILL,R4
2653 007010 026704 171640                BEQ     66$
2654 007014 001401                       ERROR   7
2655                                     ;*****TEST 24 - ERROR 7*****
2656                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2657 007020 026705 171632                66$:  CMP     TABLE,R5
2658 007024 001401                       BEQ     67$
2659 007026 104010                       ERROR   10
2660 007030                               67$:
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673 007030 000004                       ;*****
2674 007032 004567 005212                ;*TEST 25 TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO
2675 007036 000004                       ;*****
;*****
;*PROPER TERMINATION OF THIS INSTRUCTION TEST IS R1&R0
;*EQUAL ADDRESS & LENGTH OF UNEQUAL SOURCE1 STRING,
;*R2 EQUALS SOURCE2 ADDRESS, R2 EQUALS ZERO, AND ALL
;*CONDITION CODES CLEAR., EXCEPT N=1
;*****
;*****
TST25:  SCOPE
        JSR     R5,PREP
        4
;SET UP INSTRUCTION ARGUMENTS
;SOURCE1 LENGTH

```

2676	007040	016710			BUF2	:SOURCE1 ADDRESS
2677	007042	000000			0	:SOURCE2 LENGTH
2678	007044	177777			NXM	:SOURCE2 ADDRESS
2679	007046	000377			377	:FILL CHARACTER
2680	007050	012700	016710	MOV	#BUF2,R0	:GENERATE SOURCE1 STRING STARTING
2681	007054	012720	177777	MOV	#177777,(R0)+	:WITH TWO FILL BYTES
2682	007060	005020		CLR	(R0)+	:THEN TWO "ZERO" BYTES
2683	007062	004567	005256	JSR	R5,XPSW	:STORE EXPECTED PSW VALUE
2684	007066	000201		.WORD	201	
2685	007070	004767	005110	JSR	PC,GENR	:SET UP GENERAL REGISTERS
2686	007074	000277		SCC		:SET ALL CONDITION CODES, EXCEPT
2687	007076	000250		CLN		:CLEAR N=1
2688						:EXECUTE "CMPC" INSTRUCTION
2689	007100	076044		CMPC		
2690						:CHECK RESULTS
2691	007102	004767	005262	JSR	PC,CKCC	:CHECK PSW, GENERATE CONDITION CODES
2692	007106	001401		BEQ	64\$	
2693	007110	104001		ERROR	1	:*****TEST 25 - ERROR 1*****
2694						:PSW ERROR
2695						:EXPECTED PSW IS STORED AT "EXPPSW"
2696						:ACTUAL PSW IS STORED AT "CCODES"
2697	007112				64\$:	
2698	007112	023706	000676	CMP	@SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
2699	007116	001403		BEQ	65\$:BR IF OK
2700	007120	010637	000700	MOV	SP,@BADR6	:STORE BAD SP VALUE
2701	007124	104002		ERROR	2	:*****TEST 25 - ERROR 2*****
2702						:STACK POINTER NOT RESTORED BY INSTRUCTION
2703						:EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2704						:ERRONEOUS VALUE IS AT "BADR6"
2705	007126				65\$:	
2706	007126	022700	000002	CMP	#2,R0	:CHECK R0=2
2707	007132	001401		BEQ	1\$	
2708	007134	104003		ERROR	3	:*****TEST 25 - ERROR 3*****
2709						:SRC1 LENGTH ERROR
2710	007136	022701	016712	CMP	#BUF2+2,R1	:CHECK R1=BUF2+2
2711	007142	001401		BEQ	2\$	
2712	007144	104004		ERROR	4	:*****TEST 25 - ERROR 4*****
2713						:SRC1 ADDRESS ERROR
2714	007146	005702		TST	R2	:CHECK R2=0
2715	007150	001401		BEQ	3\$	
2716	007152	104005		ERROR	5	:*****TEST 25 - ERROR 5*****
2717						:SRC2 LENGTH ERROR
2718	007154	026703	171472	CMP	DSTAD,R3	:CHECK R3=SRC2 ADDRESS
2719	007160	001401		BEQ	4\$	
2720	007162	104006		ERROR	6	:*****TEST 25 - ERROR 6*****
2721						:SRC2 ADDRESS ERROR
2722	007164				4\$:	
2723	007164	026704	171464	CMP	FILL,R4	:CHECK R4 UNCHANGED
2724	007170	001401		BEQ	66\$:BR IF OK
2725	007172	104007		ERROR	7	:*****TEST 25 - ERROR 7*****
2726						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2727	007174	026705	171456	CMP	TABLE,R5	:CHECK R5 UNCHANGED
2728	007200	001401		BEQ	67\$:BR IF OK
2729	007202	104010		ERROR	10	:*****TEST 25 - ERROR 10*****
2730						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2731	007204				67\$:	

2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745 007204 000004
2746 007206 004567 005036
2747 007212 000007
2748 007214 016710
2749 007216 000003
2750 007220 016310
2751 007222 000377
2752 007224 004567 005074
2753 007230 000007
2754 007232 016710
2755 007234 105037 016712
2756 007240 004567 005100
2757 007244 000211
2758 007246 004767 004732
2759 007252 000251
2760 007254 000266
2761
2762 007256 076044
2763
2764 007260 004767 005104
2765 007264 001401
2766 007266 104001
2767
2768
2769
2770 007270
2771 007270 023706 000676
2772 007274 001403
2773 007276 010637 000700
2774 007302 104002
2775
2776
2777
2778 007304
2779 007304 026704 171344
2780 007310 001401
2781 007312 104003
2782
2783 007314 026705 171336
2784 007320 001401
2785 007322 104004
2786
2787 007324

```
*****  
*TEST 26 TEST "CMPC" FOR SOURCE1 LESS THAN SOURCE2  
*****  
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS  
*R0&R2 EQUAL THE REMAINING LENGTH OF THE UNEQUAL BYTES  
*OF THE SOURCE1&SOURCE2 STRINGS, R1&R3 EQUAL THE ADDRESS  
*OF THE FIRST UNEQUAL BYTES IN THE SOURCE1 AND SOURCE2  
*STRINGS, AND CONDITION CODE C,N=1 AND V,Z=0  
*****  
TST26: SCOPE  
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS  
7 ;SOURCE1 LENGTH  
BUF2 ;SOURCE1 ADDRESS  
3 ;SOURCE2 LENGTH  
BUF1 ;SOURCE2 ADDRESS  
377 ;FILL CHARACTER  
JSR R5,GENSRC ;GENERATE SOURCE1 STRING  
.WORD 7 ;STRING LENGTH  
.WORD BUF2 ;STRING ADDRESS  
CLRB @#BUF2+2 ;PLACE A NON-COMPARE CHARACTER IN SOURCE1 STRING  
JSR R5,XPSW ;STORE EXPECTED PSW VALUE  
.WORD 211  
JSR PC,GENR ;SET UP GENERAL REGISTERS  
+CLC!CLN ;CLEAR CONDITION CODES C&N  
+SEZ!SEV ;SET CONDITION CODES Z&V  
CMPC ;EXECUTE "CMPC" INSTRUCTION  
;CHECK RESULT  
JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES  
BEQ 64$  
ERROR 1 ;*****TEST 26 - ERROR 1*****  
;PSW ERROR  
;EXPECTED PSW IS STORED AT 'EXPPSW'  
;ACTUAL PSW IS STORED AT 'CCODES'  
64$:  
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED  
BEQ 65$ ;BR IF OK  
MOV SP,@#BADR6 ;STORE BAD SP VALUE  
ERROR 2 ;*****TEST 26 - ERROR 2*****  
;STACK POINTER NOT RESTORED BY INSTRUCTION  
;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'  
;ERRONEOUS VALUE IS AT 'BADR6'  
65$:  
CMP FILL,R4 ;CHECK R4 UNCHANGED  
BEQ 66$ ;BR IF OK  
ERROR 3 ;*****TEST 26 - ERROR 3*****  
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'  
66$:  
CMP TABLE,R5 ;CHECK R5 UNCHANGED  
BEQ 67$ ;BR IF OK  
ERROR 4 ;*****TEST 26 - ERROR 4*****  
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'  
67$:
```

```

2788 007324          1$:          ;CHECK OTHER REGISTERS
2789 007324 005067 171354      CLR      TEMP
2790 007330 016704 171312      MOV      SRCAD,R4          ;POINT R4 TO SOURCE1 ADDRESS
2791 007334 016705 171312      MOV      DSTAD,R5          ;POINT R5 TO SOURCE2 ADDRESS
2792 007340 121415          2$:      CMPB     (R4),(R5)          ;COMPARE SOURCES
2793 007342 001004               BNE      3$                ;BR, IF NOT EQUAL
2794 007344 005267 171334      INC      TEMP
2795 007350 122425               CMPB     (R4)+,(R5)+      ;POINT STRINGS TO NEXT BYTES
2796 007352 000772               BR       2$
2797 007354          3$:          ;CHECK ADDRESS REGISTERS
2798 007354 020401               CMP      R4,R1            ;CHECK R1 (SOURCE1 STRING)
2799 007356 001401               BEQ     4$
2800 007360 104005               ERROR   5
2801          ;*****TEST 26 - ERROR 5*****
2802 007362 020503          4$:      CMP      R5,R3            ;SRC1 ADDRESS ERROR
2803 007364 001401               BEQ     5$                ;CHECK R3 (SOURCE2 STRING)
2804 007366 104006               ERROR   6                ;BR IF EQUAL
2805          ;*****TEST 26 - ERROR 6*****
2806 007370 016704 171250          5$:      MOV      SRCLN,R4          ;SRC2 ADDRESS ERROR
2807 007374 166704 171304      SUB      TEMP,R4          ;CALCULATE LENGTH OF SRC1 REMAINDER
2808 007400 020400               CMP      R4,R0            ;CHECK R0 (SRC1 LENGTH)
2809 007402 001401               BEQ     6$                ;BR, IF EQUAL
2810 007404 104007               ERROR   7
2811          ;*****TEST 26 - ERROR 7*****
2812 007406 016704 171236          6$:      MOV      DSTLN,R4          ;SRC1 STRING LENGTH ERROR
2813 007412 166704 171266      SUB      TEMP,R4          ;CALCULATE LENGTH OF SRC2 REMAINDER
2814 007416 020402               CMP      R4,R2            ;CHECK R2 (SRC2 LENGTH)
2815 007420 001401               BEQ     40$              ;BR, IF EQUAL
2816 007422 104010               ERROR   10             ;*****TEST 26 - ERROR 10*****
2817          ;SRC2 STRING LENGTH ERROR
2818 007424          40$:

```

```

2819
2820
2821          ;*****
2822          ;*TEST 27      TEST "CMPC" WITH SOURCE1 LENGTH .GT. SOURCE2 LENGTH & SOURCE1=SOURCE2
2823          ;*****
2824          ;*THIS TEST VERIFIES THAT "CMPC" INDICATES EQUAL STRINGS WHEN THE
2825          ;*STRING LENGTHS ARE NOT EQUAL BUT THE EXCESS OF THE LONGER
2826          ;*STRING CONTAINS ALL FILLS.
2827          ;*****
2828          ;*****

```

```

2829 007424 000004          TST27: SCOPE
2830 007426 004567 004616      JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
2831 007432 000010          10          ;SOURCE1 LENGTH
2832 007434 016310          BUF1        ;SOURCE1 ADDRESS
2833 007436 000012          12          ;SOURCE2 LENGTH
2834 007440 015710          BUF2        ;SOURCE2 ADDRESS
2835 007442 000377          377        ;FILL CHARACTER
2836 007444 004767 004626      JSR      PC,CLDST        ;CLEAR SOURCE2 AREA
2837 007450 004567 004650      JSR      R5,GENSRC       ;GENERATE SOURCE2 STRING
2838 007454 000010          .WORD     10            ;MAKE FIRST 10 BYTES IDENTICAL
2839 007456 016710          .WORD     BUF2          ;TO SOURCE1 STRING AND THE
2840 007460 012721 177777      MOV      #177777,(R1)+   ;LAST TWO BYTES EQUAL TO FILL CHARACTER
2841 007464 004567 004654          JSR      R5,XPSW         ;STORE EXPECTED PSW VALUE
2842 007470 000204          .WORD     204
2843 007472 004767 004506          JSR      PC,GENR        ;SET UP GENERAL REGISTERS

```



```

2844 007476 000277 SCC ;SET ALL CONDITION CODES, EXCEPT
2845 007500 000244 CLZ ;CLEAR Z
2846 007502 076044 CMPC ;EXECUTE "CMPC" INSTRUCTION
2848 007504 004767 004660 JSR PC,CKCC ;CHECK PSW, GENERATE CONDITION CODES
2850 007510 001401 BEQ 64$
2851 007512 104001 ERROR 1 ;*****TEST 27 - ERROR 1*****
2852 ;PSW ERROR
2853 ;EXPECTED PSW IS STORED AT 'EXPPSW'
2854 ;ACTUAL PSW IS STORED AT 'CCODES'
2855 007514 64$:
2856 007514 023706 000676 CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2857 007520 001403 BEQ 65$ ;BR IF OK
2858 007522 010637 000700 MOV SP,@#BADR6 ;STORE BAD SP VALUE
2859 007526 104002 ERROR 2 ;*****TEST 27 - ERROR 2*****
2860 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2861 ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
2862 ;ERRONEOUS VALUE IS AT 'BADR6'
2863 007530 65$:
2864 007530 005700 TST R0 ;CHECK R0=0
2865 007532 001401 BEQ 1$
2866 007534 104003 ERROR 3 ;*****TEST 27 - ERROR 3*****
2867 ;SOURCE1 LENGTH ERROR
2868 007536 016700 171104 1$: MOV SRCAD,R0 ;CALCULATE ADDRESS OF LSB+1
2869 007542 066700 171076 ADD SRCLN,R0
2870 007546 020001 CMP R0,R1 ;CHECK R1, SOURCE1 ADDRESS
2871 007550 001401 BEQ 2$
2872 007552 104004 ERROR 4 ;*****TEST 27 - ERROR 4*****
2873 ;SOURCE1 ADDRESS ERROR
2874 007554 005702 2$: TST R2 ;CHECK R2=0
2875 007556 001401 BEQ 3$
2876 007560 104005 ERROR 5 ;*****TEST 27 - ERROR 5*****
2877 ;SOURCE2 LENGTH ERROR
2878 007562 016702 171064 3$: MOV DSTAD,R2 ;CALCULATE ADDRESS OF LSB+1
2879 007566 066702 171056 ADD DSTLN,R2
2880 007572 020203 CMP R2,R3 ;CHECK R3,SOURCE2 ADDRESS
2881 007574 001401 BEQ 4$
2882 007576 104006 ERROR 6 ;*****TEST 27 - ERROR 6*****
2883 ;SOURCE2 ADDRESS ERROR
2884 007600 4$:
2885 007600 026704 171050 CMP FILL,R4 ;CHECK R4 UNCHANGED
2886 007604 001401 BEQ 66$ ;BR IF OK
2887 007606 104007 ERROR 7 ;*****TEST 27 - ERROR 7*****
2888 ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
2889 007610 026705 171042 66$: CMP TABLE,R5 ;CHECK R5 UNCHANGED
2890 007614 001401 BEQ 67$ ;BR IF OK
2891 007616 104010 ERROR 10 ;*****TEST 27 - ERROR 10*****
2892 ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
2893 007620 67$:
2894
2895
2896 ;*****
2897 ;*TEST 30 TEST THAT BAD "SEARCH" OPCODES TRAP
2898 ;*****
2899 ;*THIS TEST VERIFIES THAT OPCODES 076045-->076047 TRAP TO

```

```

2900      ;*LOCATION 10.
2901      ;*****
2902      ;*****
2903 007620 000004          TST30: SCOPE
2904 007622 004567 004422  JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
2905 007626 016310          BUF1      ;SOURCE LENGTH
2906 007630 000010          10       ;SOURCE ADDRESS
2907 007632 016710          BUF2      ;DESTINATION LENGTH
2908 007634 000010          10       ;DESTINATION ADDRESS
2909 007636 000377          377      ;FILL CHARACTER
2910 007640 012737 076045 007666  MOV      #076045,@#BD30 ;STORE THE FIRST BAD MOVE OPCODE
2911 007646 013767 000010 171032  MOV      @#10,TEMP1    ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR
2912 007654 012737 007676 000010  MOV      #T30CONT,@#10 ;POINT ILLEGAL INSTRUCTION VECTOR TO CONTINUE TEST
2913 007662 004767 004316  REP30: JSR      PC,GENR    ;SET UP GENERAL REGISTERS
2914
2915 007666 076045          BD30:  .WORD  076045    ;EXECUTE BAD SEARCH INSTRUCTION
2916
2917 007670 016700 177772          MOV      BD30,R0      ;STORE BAD OPCODE THAT DID NOT TRAP
2918 007674 104001          ERROR  1          ;*****TEST 30 - ERROR 1*****
2919
2920
2921
2922
2922 007676 012626          T30CONT:MOV    (SP)+,(SP)+ ;RESTORE THE STACK POINTER AFTER THE TRAP
2923 007700 005267 177762          INC      BD30        ;INCREMENT INSTRUCTION OPCODE
2924 007704 022767 076050 177754  CMP      #076050,BD30 ;FINISHED WITH BAD SEARCH OPCODES?
2925 007712 001363          BNE      REP30       ;BR IF NOT
2926 007714 016737 170766 000010  MOV      TEMP1,@#10   ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR
2927
2928
2929
2930
2931      ;*****
2932      ;*TEST 31      TEST INTERRUPTABILITY OF "CMPC" INSTRUCTION
2933      ;*****
2934      ;*THIS TEST INTERRUPTS THE EXECUTION OF THE "CMPC"
2935      ;*INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
2936      ;*INTERRUPT, VERIFIES THE RESULT.  THE PROPER RESULT
2937      ;*IS FOR SOURCE1=SOURCE2.
2938      ;*****
2939      ;*****
2940 007722 000004          TST31: SCOPE
2941 007724 004767 004462  JSR      PC,SKPINT    ;SET FLAG -- TEMP, WHICH INDICATES IF
2942
2943
2943 007730 005767 170750          TST      TEMP        ;NEXT TEST IS TO BE EXECUTED NEXT.
2944 007734 001124          BNE      TST32       ;CHECK IF -TEMP- IS SET
2945 007736 004567 004306  JSR      R5,PREP     ;TEMP IS SET, GO TO NEXT TEST
2946 007742 000400          400      ;SET UP INSTRUCTION ARGUMENTS
2947 007744 016310          BUF1      ;SOURCE1 LENGTH
2948 007746 000400          400      ;SOURCE1 ADDRESS
2949 007750 016310          BUF1      ;SOURCE2 LENGTH
2950 007752 000377          377      ;SOURCE2 ADDRESS
2951 007754 012767 010042 170706  MOV      #CMC,PCI     ;FILL CHARACTER
2952 007762 012777 014506 170674  MOV      #INTR,@TVECT ;STORE PC OF TEST INSTRUCTION
2953 007770 005077 170672          CLR      @TPSW       ;POINT TTY VECTOR TO INTERRUPT ROUTINE
2954 007774 004767 004462  JSR      PC,TDONE    ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
2955 010000 013777 000554 170654  MOV      @#NULL,@TBUF ;SEND CARRIAGE RETURN

```


MAIN. MACY11 30(1046) 22-JAN-82 08:44 PAGE 57
 CVKAIB.P11 22-JAN-82 08:43 T31

TEST INTERRUPTABILITY OF "CMPC" INSTRUCTION

SEQ 0056

```

3012 010172 104010          ERROR 10          ;*****TEST 31 - ERROR 10*****
3013                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3014 010174                                     67$:
3015 010174 106427 000200      MTPS      #200          ;RESTORE PSW TO PRIORITY 7
3016 010200 016777 170462 170456  MOV      TPSW,@TVECT ;RESTORE TRAP CATCHER
3017
3018
3019
3020
3021 ::*****
3022 :*TEST 32      TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE
3023 :*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
3024 :*ADDRESS OF THE FIRST OCCURRENCE OF THE "CHARACTER"
3025 :*IN THE SOURCE STRING, R0 EQUALS REMAINING LENGTH
3026 :*OF THE SOURCE STRING, AND ALL CONDITION CODES CLEAR.
3027 :*****
3028 :*****
3029 TST32:  SCOPE
3030 JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
3031          10              ;SOURCE LENGTH
3032          BUF1            ;SOURCE ADDRESS
3033          NXM             ;STORE NON-ZERO VALUES TO TEST R2 & R3
3034          NXM             ; UNAFFECTED BY DIS STRING INSTRUCTIONS
3035          4               ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3036 JSR      R5,XPSW         ;STORE EXPECTED PSW VALUE
3037          .WORD          200
3038 JSR      PC,GENR        ;SET UP GENERAL REGISTERS
3039 SCC                      ;SET ALL CONDITION CODES
3040                                ;EXECUTE "LOCC"
3041 LOCC
3042                                ;CHECK RESULTS
3043 JSR      PC,CKCC         ;CHECK PSW, GENERATE CONDITION CODES
3044 BEQ      64$
3045 ERROR 1
3046                                ;*****TEST 32 - ERROR 1*****
3047                                ;PSW ERROR
3048                                ;EXPECTED PSW IS STORED AT "EXPPSW"
3049                                ;ACTUAL PSW IS STORED AT "CCODES"
3050 64$:  CMP      @#SAVR6,SP  ;VERIFY STACK POINTER IS RESTORED
3051 BEQ      65$
3052 MOV      SP,@#BADR6     ;STORE BAD SP VALUE
3053 ERROR 2
3054                                ;*****TEST 32 - ERROR 2*****
3055                                ;STACK POINTER NOT RESTORED BY INSTRUCTION
3056                                ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3057                                ;ERRONEOUS VALUE IS AT "BADR6"
3058 65$:  CMP      FILL,R4    ;CHECK R4 UNCHANGED
3059 BEQ      66$
3060 ERROR 3
3061                                ;*****TEST 32 - ERROR 3*****
3062 66$:  CMP      TABLE,R5 ;CHECK R5 UNCHANGED
3063 BEQ      67$
3064 ERROR 4
3065                                ;*****TEST 32 - ERROR 4*****
3066 67$:  MOV      SRCAD,R3   ;LOCATE "CHARACTER"
3067

```



```

3068 010314 016702 170324      MOV      SRCLN,R2
3069 010320 121367 170330      68$:    CMPB   (R3),FILL
3070 010324 001403                    BEQ     69$
3071 010326 105723                    TSTB   (R3)+      ;POINT TO NEXT BYTE
3072 010330 005302                    DEC     R2         ;DECREMENT BYTE COUNT
3073 010332 001372                    BNE    68$        ;BR, IF NOT FINISHED
3074 010334 020200      69$:    CMP     R2,R0     ;CHECK R0=REMAINING SOURCE LENGTH
3075 010336 001401                    BEQ    70$
3076 010340 104005                    ERROR  5          ;*****TEST 32 - ERROR 5*****
3077                                     ;LOCATE LENGTH ERROR
3078 010342 020301      70$:    CMP     R3,R1     ;CHECK R1=ADDRESS OF CHARACTER
3079 010344 001401                    BEQ    71$
3080 010346 104006                    ERROR  6          ;*****TEST 32 - ERROR 6*****
3081                                     ;LOCATE ADDRESS ERROR
3082 010350      71$:

```

```

:*****
:*TEST 33      TEST 'LOCC' INSTRUCTION WITH SEARCH CHARACTER IN SOURCE, SL .GT. 10000
:*****
:*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
:*ADDRESS OF THE FIRST OCCURANCE OF THE "CHARACTER"
:*IN THE SOURCE STRING, R0 EQUALS REMAINING LENGTH
:*OF THE SOURCE STRING, AND ALL CONDITION CODES CLEAR
:*EXCEPT N=1.
:*****

```

```

3095 010350 000004      TST33:  SCOPE
3096 010352 004567 003672      JSR     R5,PREP   ;SET UP INSTRUCTION ARGUMENTS
3097 010356 100010                    100010 ;SOURCE LENGTH
3098 010360 016310                    BUF1   ;SOURCE ADDRESS
3099 010362 177777                    NXM    ;STORE NON-ZERO VALUES TO TEST R2 & R3
3100 010364 177777                    NXM    ; UNAFFECTED BY DIS STRING INSTRUCTIONS
3101 010366 000004                    4     ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3102 010370 004567 003750      JSR     R5,XPSW   ;STORE EXPECTED PSW VALUE
3103 010374 000210      .WORD  210
3104 010376 004767 003602      JSR     PC,GENR   ;SET UP GENERAL REGISTERS
3105 010402 000277      SCC     ;SET ALL CONDITION CODES
3106                                     ;EXECUTE 'LOCC'
3107 010404 076040      LOCC
3108                                     ;CHECK RESULTS
3109 010406 004767 003756      JSR     PC,CKCC   ;CHECK PSW, GENERATE CONDITION CODES
3110 010412 001401                    BEQ    64$
3111 010414 104001                    ERROR  1          ;*****TEST 33 - ERROR 1*****
3112                                     ;PSW ERROR
3113                                     ;EXPECTED PSW IS STORED AT 'EXPPSW'
3114                                     ;ACTUAL PSW IS STORED AT 'CCODES'
3115 010416      64$:    CMP     @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
3116 010416 023706 000676                    BEQ    65$        ;BR IF OK
3117 010422 001403                    MOV    SP,@#BADR6 ;STORE BAD SP VALUE
3118 010424 010637 000700                    ERROR  2          ;*****TEST 33 - ERROR 2*****
3119 010430 104002                                     ;STACK POINTER NOT RESTORED BY INSTRUCTION
3120                                     ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
3121                                     ;ERRONEOUS VALUE IS AT 'BADR6'
3122
3123 010432      65$:

```



```

3236 010712 000273 +SEV!SEN!SEC ;SET ALL OTHER CONDITION CODES
3237 ;EXECUTE 'LOCC'
3238 010714 076040 LOCC ;CHECK RESULTS
3239 ;CHECK PSW, GENERATE CONDITION CODES
3240 010716 004767 003446 JSR PC,CKCC
3241 010722 001401 BEQ 64$
3242 010724 104001 ERROR 1 ;*****TEST 35 - ERROR 1*****
3243 ;PSW ERROR
3244 ;EXPECTED PSW IS STORED AT 'EXPPSW'
3245 ;ACTUAL PSW IS STORED AT 'CCODES'
3246 010726 64$:
3247 010726 023706 000676 CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
3248 010732 001403 BEQ 65$ ;BR IF OK
3249 010734 010637 000700 MOV SP,@#BADR6 ;STORE BAD SP VALUE
3250 010740 104002 ERROR 2 ;*****TEST 35 - ERROR 2*****
3251 ;STACK POINTER NOT RESTORED BY INSTRUCTION
3252 ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
3253 ;ERRONEOUS VALUE IS AT 'BADR6'
3254 010742 65$:
3255 010742 105700 TSTB R0 ;CHECK R0
3256 010744 001401 BEQ 1$
3257 010746 104003 ERROR 3 ;*****TEST 35 - ERROR 3*****
3258 ;R0 CHANGED
3259 010750 026701 167672 1$: CMP SRCAD,R1 ;CHECK R1
3260 010754 001401 BEQ 40$
3261 010756 104004 ERROR 4 ;*****TEST 35 - ERROR 4*****
3262 ;R1 CHANGED
3263 010760 40$:
3264
3265
3266
3267

```

```

:*****
:*TEST 36 TEST INTERRUPTABILITY OF 'LOCC' INSTRUCTION
:*****
:*THIS TEST INTERRUPTS THE EXECUTION OF 'LOCC' INSTRUCTION,
:*RESUMES EXECUTION AFTER THE INTERRUPT, AND CHECKS THE
:*RESULTS. RESULTS INDICATE THE SEARCH CHARACTER FOUND.
:*****

```

```

3274
3275 010760 000004
3276 010762 004767 003424 TST36: SCOPE
3277 JSR PC,SKPINT ;SET FLAG -- TEMP, WHICH INDICATES IF
3278 ;NEXT TEST IS TO BE EXECUTED NEXT.
3279 010766 005767 167712 TST TEMP ;CHECK IF -TEMP- IS SET
3280 010772 001123 BNE TST37 ;TEMP IS SET, GO TO NEXT TEST
3281 010774 004567 003250 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
3282 011000 000377 377 ;SOURCE LENGTH
3283 011002 016310 BUF1 ;SOURCE ADDRESS
3284 011004 177777 NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
3285 011006 177777 NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
3286 011010 000200 200 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3287 011012 012767 011076 167650 MOV #LOC,PCI ;STORE PC OF TEST INSTRUCTION
3288 011020 012777 014506 167636 MOV #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
3289 011026 005077 167634 CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
3290 011032 004767 003424 JSR PC,TDONE
3291 011036 013777 000554 167616 MOV @#NULL,@TBUF ;SEND CARRIAGE RETURN
3292 011044 004567 003274 JSR R5,XPSW ;STORE EXPECTED PSW VALUE

```


3292	011050	000000			.WORD	00		
3293	011052	106427	000000		MTPS	#0		:SET PSW TO ALLOW INTERRUPTS
3294	011056	052777	000100	167574	BIS	#100,@TCSR		:ENABLE TTY INTERRUPTS
3295	011064	004767	003114		JSR	PC,GENR		:SET UP GENERAL REGISTERS
3296	011070	010637	000676		MOV	SP,@SAVR6		:COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
3297	011074	000277			SCC			:SET ALL CONDITION CODES
3298								
3299	011076	076040			LOC:	LOCC		:EXECUTE 'LOCC'
3300								
3301	011100	106767	167566		MFPS	CCODES		:STORE THE PSW
3302	011104	032777	000100	167546	BIT	#100,@TCSR		:IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
3303	011112	001364			BNE	REPLC		:BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
3304	011114	042767	177400	167550	BIC	#177400,CCODES		:CLEAR ALL UNUSED BITS
3305	011122	023767	000674	167542	CMP	@EXPPSW,CCODES		:CHECK PSW AGAINST EXPECTED VALUE
3306	011130	001401			BEQ	64\$:BR, IF EQUAL
3307	011132	104001			ERROR	1		:*****TEST 36 - ERROR 1*****
3308								:PSW ERROR
3309								:EXPECTED PSW IS STORED AT 'EXPPSW'
3310								:ACTUAL PSW IS STORED AT 'CCODES'
3311	011134				64\$:			
3312	011134	023706	000676		CMP	@SAVR6,SP		:VERIFY STACK POINTER IS RESTORED
3313	011140	001403			BEQ	65\$:BR IF OK
3314	011142	010637	000700		MOV	SP,@BADR6		:STORE BAD SP VALUE
3315	011146	104002			ERROR	2		:*****TEST 36 - ERROR 2*****
3316								:STACK POINTER NOT RESTORED BY INSTRUCTION
3317								:EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
3318								:ERRONEOUS VALUE IS AT 'BADR6'
3319	011150				65\$:			
3320	011150	026704	167500		CMP	FILL,R4		:CHECK R4 UNCHANGED
3321	011154	001401			BEQ	66\$:BR IF OK
3322	011156	104003			ERROR	3		:*****TEST 36 - ERROR 3*****
3323								:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
3324	011160	026705	167472		66\$:			
3325	011164	001401			CMP	TABLE,R5		:CHECK R5 UNCHANGED
3326	011166	104004			BEQ	67\$:BR IF OK
3327					ERROR	4		:*****TEST 36 - ERROR 4*****
3328								:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
3329	011170	016703	167452		67\$:			
3330	011174	016702	167444		MOV	SRCAD,R3		:LOCATE 'CHARACTER'
3331	011200	121367	167450		MOV	SRCLN,R2		
3332	011204	001403			68\$:			
3333	011206	105723			CMPB	(R3),FILL		
3334	011210	005302			BEQ	69\$		
3335	011212	001372			TSTB	(R3)+		:POINT TO NEXT BYTE
3336	011214	020200			DEC	R2		:DECREMENT BYTE COUNT
3337	011216	001401			BNE	68\$:BR, IF NOT FINISHED
3338	011220	104005			69\$:			
3339					CMP	R2,R0		:CHECK R0=REMAINING SOURCE LENGTH
3340	011222	020301			BEQ	70\$		
3341	011224	001401			ERROR	5		:*****TEST 36 - ERROR 5*****
3342	011226	104006						:LOCATE LENGTH ERROR
3343								:CHECK R1=ADDRESS OF CHARACTER
3344	011230				70\$:			
3345								
3346	011230	106427	000200		40\$:			
3347	011234	016777	167426	167422	MTPS	#200		:RESTORE PSW TO PRIORITY 7
					MOV	TPSW,@TVECT		:RESTORE TRAP CATCHER

```
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361 011242 000004
3362 011244 004567 003000
3363 011250 000007
3364 011252 016710
3365 011254 177777
3366 011256 177777
3367 011260 000001
3368 011262 004567 003056
3369 011266 000200
3370 011270 004767 002710
3371 011274 000277
3372
3373 011276 076041
3374
3375 011300 004767 003064
3376 011304 001401
3377 011306 104001
3378
3379
3380
3381 011310
3382 011310 023706 000676
3383 011314 001403
3384 011316 010637 000700
3385 011322 104002
3386
3387
3388
3389 011324
3390 011324 026704 167324
3391 011330 001401
3392 011332 104003
3393
3394 011334 026705 167316
3395 011340 001401
3396 011342 104004
3397
3398 011344
3399 011344 016703 167276
3400 011350 016702 167270
3401 011354 121367 167274
3402 011360 001003
3403 011362 105723

*****
:TEST 37 TEST "SKPC" INSTRUCTION WITH NON-SKIP CHARACTERS IN SOURCE
*****
:PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
:ADDRESS OF THE FIRST OCCURANCE OF A CHARACTER
:OTHER THAN THE SEARCH CHARACTER, R0 EQUALS THE
:REMAINING LENGTH OF THE SOURCE STRING, AND ALL
:CONDITION CODES CLEAR
*****
TST37: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      7              ;SOURCE LENGTH
      BUF2         ;SOURCE ADDRESS
      NXM         ;STORE NON-ZERO VALUES TO TEST R2 & R3
      NXM         ; UNAFFECTED BY DIS STRING INSTRUCTIONS
      1           ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
      JSR      R5,XPSW     ;STORE EXPECTED PSW VALUE
      .WORD    200
      JSR      PC,GENR     ;SET UP GENERAL RESISTERS
      SCC              ;SET ALL CONDITION CODES
                        ;EXECUTE "SKPC"
      SKPC
                        ;CHECK RESULTS
      JSR      PC,CKCC     ;CHECK PSW, GENERATE CONDITION CODES
      BEQ     64$
      ERROR    1          ;*****TEST 37 - ERROR 1*****
                        ;PSW ERROR
                        ;EXPECTED PSW IS STORED AT 'EXPPSW'
                        ;ACTUAL PSW IS STORED AT 'CCODES'
64$:
      CMP     @#SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ     65$
      MOV     SP,@#BADR6  ;BR IF OK
      ERROR  2           ;STORE BAD SP VALUE
                        ;*****TEST 37 - ERROR 2*****
                        ;STACK POINTER NOT RESTORED BY INSTRUCTION
                        ;EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
                        ;ERRONEOUS VALUE IS AT 'BADR6'
65$:
      CMP     FILL,R4     ;CHECK R4 UNCHANGED
      BEQ     66$
      ERROR  3           ;BR IF OK
                        ;*****TEST 37 - ERROR 3*****
                        ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
66$:
      CMP     TABLE,R5  ;CHECK R5 UNCHANGED
      BEQ     67$
      ERROR  4           ;BR IF OK
                        ;*****TEST 37 - ERROR 4*****
                        ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
67$:
      MOV     SRCAD,R3    ;SKIP "CHARACTER"
      MOV     SRCLN,R2
68$:
      CMPB   (R3),FILL
      BNE   69$
      TSTB  (R3)+
                        ;POINT TO NEXT BYTE
```



```
3516 011640 64$: CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
3517 011640 023706 000676 BEQ 65$ ;BR IF OK
3518 011644 001403 MOV SP,@#BADR6 ;STORE BAD SP VALUE
3519 011646 010637 000700 ERROR 2 ;*****TEST 41 - ERROR 2*****
3520 011652 104002 ;STACK POINTER NOT RESTORED BY INSTRUCTION
3521 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3522 ;ERRONEOUS VALUE IS AT "BADR6"
3523
3524 011654 65$: TSTB R0 ;CHECK R0
3525 011654 105700 BEQ 1$
3526 011656 001401 ERROR 3 ;*****TEST 41 - ERROR 3*****
3527 011660 104003 ;R0 CHANGED
3528 ;CHECK R1
3529 011662 026701 166760 1$: CMP SRCAD,R1
3530 011666 001401 BEQ 40$
3531 011670 104004 ERROR 4 ;*****TEST 41 - ERROR 4*****
3532 ;R1 CHANGED
3533 011672 40$:
3534
3535
3536
3537
```

```
::*****
:*TEST 42 TEST INTERRUPTABILITY OF "SKPC" INSTRUCTION
::*****
:*THIS TEST INTERRUPTS THE EXECUTION OF "SKPC"
:*INSTRUCTION, RESUMES EXECUTION AFTER THE INTERRUPT,
:*AND CHECKS THE RESULTS. RESULTS INDICATE THE NON-SKIP
:*CHARACTER FOUND
::*****
::*****
```

```
3546 011672 000004 TST42: SCOPE
3547 011674 004767 002512 JSR PC,SKPINT ;SET FLAG -- TEMP, WHICH INDICATES IF
3548 ;NEXT TEST IS TO BE EXECED NEXT.
3549 011700 005767 167000 TST TEMP ;CHECK IF -TEMP- IS SET
3550 011704 001135 BNE TST43 ;TEMP IS SET, GO TO NEXT TEST
3551 011706 004567 002334 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
3552 011712 000400 400 ;SOURCE LENGTH
3553 011714 016710 BUF2 ;SOURCE ADDRESS
3554 011716 177777 NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
3555 011720 177777 NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
3556 011722 000377 377 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3557 011724 016700 166716 MOV SRCAD,R0 ;GENERATE SOURCE STRING OF ALL
3558 011730 016701 166710 MOV SRCLN,R1 ;SKIP CHARACTERS EXCEPT THE LAST BYTE
3559 011734 005301 DEC R1 ;ADJUST BYTE COUNT=SRCLN-1
3560 011736 116720 166712 CONTG: MOVFB FILL,(R0)+
3561 011742 005301 DEC R1
3562 011744 001374 BNE CONTG
3563 011746 105020 CLRB (R0)+ ;PUT NON-SKIP CHARACTER IN LAST BYTE
3564 011750 012767 012034 166712 MOV #SKP,PCI ;STORE PC OF TEST INSTRUCTION
3565 011756 012777 014506 166700 MOV #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
3566 011764 005077 166676 CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
3567 011770 004767 002466 JSR PC,TDONE
3568 011774 013777 000554 166660 MOV @#NULL,@TBUF ;SEND CARRIAGE RETURN
3569 012002 004567 002336 JSR R5,XPSW ;STORE EXPECTED PSW VALUE
3570 012006 000000 .WORD 00
3571 012010 106427 000000 MTPS #0 ;SET PSW TO ALLOW INTERRUPTS
```

3572	012014	052777	000100	166636		BIS	#100,@TCSR	:ENABLE TTY INTERRUPTS
3573	012022	004767	002156		REPSKP:	JSR	PC,GENR	:SET UP GENERAL REGISTERS
3574	012026	010637	000676			MOV	SP,@SAVR6	:COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
3575	012032	000277				SCC		:SET ALL CONDITION CODES
3576								
3577	012034	076041			SKP:	SKPC		:EXECUTE "SKPC"
3578								
3579								:CHECK RESULTS
3580	012036	106767	166630			MFPS	CCODES	:STORE THE PSW
3581	012042	032777	000100	166610		BIT	#100,@TCSR	:IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
3582	012050	001364				BNE	REPSKP	:BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
3583	012052	042767	177400	166612		BIC	#177400,CCODES	:CLEAR ALL UNUSED BITS
3584	012060	023767	000674	166604		CMP	@EXPPSW,CCODES	:CHECK PSW AGAINST EXPECTED VALUE
3585	012066	001401				BEQ	64\$:BR, IF EQUAL
3586	012070	104001				ERROR	1	:*****TEST 42 - ERROR 1*****
3587								:PSW ERROR
3588								:EXPECTED PSW IS STORED AT "EXPPSW"
3589								:ACTUAL PSW IS STORED AT "CCODES"
3590	012072				64\$:			
3591	012072	023706	000676			CMP	@SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
3592	012076	001403				BEQ	65\$:BR IF OK
3593	012100	010637	000700			MOV	SP,@BADR6	:STORE BAD SP VALUE
3594	012104	104002				ERROR	2	:*****TEST 42 - ERROR 2*****
3595								:STACK POINTER NOT RESTORED BY INSTRUCTION
3596								:EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3597								:ERRONEOUS VALUE IS AT "BADR6"
3598	012106				65\$:			
3599	012106	026704	166542			CMP	FILL,R4	:CHECK R4 UNCHANGED
3600	012112	001401				BEQ	66\$:BR IF OK
3601	012114	104003				ERROR	3	:*****TEST 42 - ERROR 3*****
3602								:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3603	012116	026705	166534		66\$:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
3604	012122	001401				BEQ	67\$:BR IF OK
3605	012124	104004				ERROR	4	:*****TEST 42 - ERROR 4*****
3606								:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3607	012126				67\$:			
3608	012126	016703	166514			MOV	SRCAD,R3	:SKIP "CHARACTER"
3609	012132	016702	166506			MOV	SRCLN,R2	
3610	012136	121367	166512		68\$:	CMPB	(R3),FILL	
3611	012142	001003				BNE	69\$	
3612	012144	105723				TSTB	(R3)+	:POINT TO NEXT BYTE
3613	012146	005302				DEC	R2	:DECREMENT BYTE COUNT
3614	012150	001372				BNE	68\$:BR, IF NOT FINISHED
3615	012152	020200			69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3616	012154	001401				BEQ	70\$	
3617	012156	104005				ERROR	5	:*****TEST 42 - ERROR 5*****
3618								:SKIP LENGTH ERROR
3619	012160	020301			70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3620	012162	001401				BEQ	71\$	
3621	012164	104006				ERROR	6	:*****TEST 42 - ERROR 6*****
3622								:SKIP ADDRESS ERROR
3623	012166				71\$:			
3624								
3625	012166	106427	000200		40\$:	MTPS	#200	:RESTORE PSW TO PRIORITY 7
3626	012172	016777	166470	166464		MOV	TPSW,@TVECT	:RESTORE TRAP CATCHER
3627								


```

3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638 012200 000004
3639 012202 004567 002042
3640 012206 000003
3641 012210 016310
3642 012212 177777
3643 012214 177777
3644 012216 000002
3645 012220 012767 016310 166430
3646 012226 004567 002112
3647 012232 000200
3648 012234 004767 001744
3649 012240 016705 166412
3650 012244 000277
3651
3652 012246 076042
3653
3654 012250 004767 002114
3655 012254 001401
3656 012256 104001
3657
3658
3659
3660 012260
3661 012260 023706 000676
3662 012264 001403
3663 012266 010637 000700
3664 012272 104002
3665
3666
3667
3668 012274
3669 012274 026704 166354
3670 012300 001401
3671 012302 104003
3672
3673 012304 026705 166346
3674 012310 001401
3675 012312 104004
3676
3677 012314
3678 012314 016703 166326
3679 012320 016702 166320
3680 012324 111304
3681 012326 042704 177400
3682 012332 136467 016310 166314
3683 012340 001003

```

```

*****
*TEST 43 TEST "SCANC" WITH A MEMBER CHARACTER IN SOURCE & SL .GT. 0 AT CHARACTER
*****
*PROPER TERMINATION OF TEST IS R1 EQUAL ADDRESS OF SOURCE
*BYTE WHICH WHEN ANDED WITH MASK=1 R0 EQUALS REMAINDER
*OF SOURCE LENGTH, AND ALL CONDITION CODES CLEAR.
*****
TST43: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
3 ;SOURCE LENGTH
BUF1 ;SOURCE ADDRESS
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
2 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
MOV #BUF1,TABLE
JSR R5,XPSW ;STORE EXPECTED PSW VALUE
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
MOV TABLE,R5
SCC ;SET ALL CONDITION CODES
;EXECUTE "SCANC"
SCANC ;CHECK RESULTS
;CHECK PSW, GENERATE CONDITION CODES
JSR PC,CKCC
BEQ 64$
ERROR 1
*****TEST 43 - ERROR 1*****
PSW ERROR
EXPECTED PSW IS STORED AT "EXPPSW"
ACTUAL PSW IS STORED AT "CCODES"
64$:
CMP @#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
BEQ 65$ ;BR IF OK
MOV SP,@#BADR6 ;STORE BAD SP VALUE
ERROR 2
*****TEST 43 - ERROR 2*****
STACK POINTER NOT RESTORED BY INSTRUCTION
EXPECTED VALUE OF SP IS STORED AT "SAVR6"
ERRONEOUS VALUE IS AT "BADR6"
65$:
CMP FILL,R4 ;CHECK R4 UNCHANGED
BEQ 66$ ;BR IF OK
ERROR 3
*****TEST 43 - ERROR 3*****
R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
66$:
CMP TABLE,R5 ;CHECK R5 UNCHANGED
BEQ 67$ ;BR IF OK
ERROR 4
*****TEST 43 - ERROR 4*****
R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$:
MOV SRCAD,R3 ;POINT R3 TO SOURCE STRING
MOV SRCLN,R2 ;STORE SOURCE LENGTH IN R2
MOV (R3),R4 ;USE SOURCE BYTE AS INDEX
BIC #177400,R4 ;CLEAR ANY BYTE SIGN EXTENSION
BITB BUF1(R4),FILL ;"AND" TABLE ENTRY WITH MASK
BNE 69$

```

```

3684 012342 105723      TSTB      (R3)+      :POINT TO NEXT SOURCE BYTE
3685 012344 005302      DEC       R2         :DECREMENT BYTE COUNT
3686 012346 001366      BNE      68$        :BR, IF NOT FINISHED
3687 012350 020200      69$:     CMP       R2,R0   :CHECK R0=REMAINING SOURCE LENGTH
3688 012352 001401      BEQ      70$        :
3689 012354 104005      ERROR    5          :*****TEST 43 - ERROR 5*****
3690                                     :SCAN LENGTH ERROR
3691 012356 020301      70$:     CMP       R3,R1   :CHECK R1=ADDRESS OF CHARACTER
3692 012360 001401      BEQ      71$        :
3693 012362 104006      ERROR    6          :*****TEST 43 - ERROR 6*****
3694                                     :SCAN ADDRESS ERROR
3695 012364      71$:

```

```

:*****
:*TEST 44      TEST "SCANC" FAILS, ALL NON-MEMBER CHARACTERS IN STRING
:*****
:*THIS TEST VERIFIES THAT "SCANC" INDICATES NO MEMBERS FOUND
:*RESULT IS R0=0, R1 EQUALS THE ADDRESS+1 OF LSB IN SOURCE,
:*AND ALL CONDITION CODES CLEAR EXCEPT Z=1.
:*****
:*****

```

```

3700
3701
3702
3703
3704
3705
3706
3707 012364 000004      TST44:   SCOPE
3708 012366 004567 001656      JSR      R5,PREP    :SET UP INSTRUCTION ARGUMENTS
3709 012372 000200      200      :SOURCE LENGTH
3710 012374 016310      BUF1     :SOURCE ADDRESS
3711 012376 177777      NXM      :STORE NON-ZERO VALUES TO TEST R2 & R3
3712 012400 177777      NXM      : UNAFFECTED BY DIS STRING INSTRUCTIONS
3713 012402 000000      0        :SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3714 012404 012767 016310 166244      MOV      #BUF1,TABLE
3715 012412 004567 001726      JSR      R5,XPSW    :STORE EXPECTED PSW VALUE
3716 012416 000204      .WORD   204
3717 012420 004767 001560      JSR      PC,GENR    :SET UP GENERAL REGISTERS
3718 012424 016705 166226      MOV      TABLE,R5
3719 012430 000277      SCC
3720 012432 000244      CLZ      :SET ALL CONDITION CODES, EXCEPT
:CLEAR "Z" CONDITION CODE
3721
3722 012434 076042      SCANC    :EXECUTE "SCANC"
3723
3724                                     :CHECK RESULTS
3725 012436 004767 001726      JSR      PC,CKCC    :CHECK PSW, GENERATE CONDITION CODES
3726 012442 001401      BEQ      64$        :
3727 012444 104001      ERROR    1          :*****TEST 44 - ERROR 1*****
3728                                     :PSW ERROR
3729                                     :EXPECTED PSW IS STORED AT 'EXPPSW'
3730                                     :ACTUAL PSW IS STORED AT 'CCODES'
3731 012446      64$:     CMP      @#SAVR6,SP  :VERIFY STACK POINTER IS RESTORED
3732 012446 023706 000676      BEQ      65$        :BR IF OK
3733 012452 001403      MOV      SP,@#BADR6 :STORE BAD SP VALUE
3734 012454 010637 000700      ERROR    2          :*****TEST 44 - ERROR 2*****
3735 012460 104002      :STACK POINTER NOT RESTORED BY INSTRUCTION
3736                                     :EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3737                                     :ERRONEOUS VALUE IS AT "BADR6"
3738
3739 012462      65$:

```



```

3740 012462 026704 166166      CMP      FILL,R4      ;CHECK R4 UNCHANGED
3741 012466 001401      BEQ      66$         ;BR IF OK
3742 012470 104003      ERROR    3           ;*****TEST 44 - ERROR 3*****
3743                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
3744 012472 026705 166160      66$:    CMP      TABLE,R5 ;CHECK R5 UNCHANGED
3745 012476 001401      BEQ      67$         ;BR IF OK
3746 012500 104004      ERROR    4           ;*****TEST 44 - ERROR 4*****
3747                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION 'TABLE'
3748 012502                                     67$:
3749 012502 016703 166140      MOV      SRCAD,R3    ;POINT R3 TO SOURCE STRING
3750 012506 016702 166132      MOV      SRCLN,R2   ;STORE SOURCE LENGTH IN R2
3751 012512 111304      MOV      (R3),R4    ;USE SOURCE BYTE AS INDEX
3752 012514 042704 177400      BIC      #177400,R4 ;CLEAR ANY BYTE SIGN EXTENSION
3753 012520 136467 016310 166126 BITB     BUF1(R4),FILL ;"AND" TABLE ENTRY WITH MASK
3754 012526 001003      BNE      69$         ;POINT TO NEXT SOURCE BYTE
3755 012530 105723      TSTB     (R3)+      ;DECREMENT BYTE COUNT
3756 012532 005302      DEC      R2         ;BR, IF NOT FINISHED
3757 012534 001366      BNE      68$         ;CHECK R0=REMAINING SOURCE LENGTH
3758 012536 020200      69$:    CMP      R2,R0
3759 012540 001401      BEQ      70$         ;*****TEST 44 - ERROR 5*****
3760 012542 104005      ERROR    5           ;SCAN LENGTH ERROR
3761                                     ;CHECK R1=ADDRESS OF CHARACTER
3762 012544 020301      70$:    CMP      R3,R1
3763 012546 001401      BEQ      71$         ;*****TEST 44 - ERROR 6*****
3764 012550 104006      ERROR    6           ;SCAN ADDRESS ERROR
3765
3766 012552      71$:
3767
3768
3769

```

```

:*****
:*TEST 45      TEST INTERRUPTABILITY OF "SCANC"
:*****
:*THIS TEST INTERRUPTS THE "SCANC" INSTRUCTION, RESUMES
:*EXECUTION AFTER THE INTERRUPT, AND CHECKS RESULTS.
:*RESULTS ARE R0=0, R1 EQUALS ADDRESS+1 OF LSB IN SOURCE,
:*AND ALL CONDITION CODES CLEAR EXCEPT Z=1.
:*****

```

```

3778
3779 012552 000004      TST45:  SCOPE
3780 012554 004767 001632      JSR      PC,SKPINT  ;SET FLAG -- TEMP, WHICH INDICATES IF
3781                                     ;NEXT TEST IS TO BE EXECUTED NEXT.
3782 012560 005767 166120      TST      TEMP      ;CHECK IF -TEMP- IS SET
3783 012564 001135      BNE      TST46     ;TEMP IS SET, GO TO NEXT TEST
3784 012566 004567 001456      JSR      R5,PREP   ;SET UP INSTRUCTION ARGUMENTS
3785 012572 000400      400          ;SOURCE LENGTH
3786 012574 016310      BUF1        ;SOURCE ADDRESS
3787 012576 177777      NXM        ;STORE NON-ZERO VALUES TO TEST R2 & R3
3788 012600 177777      NXM        ; UNAFFECTED BY DIS STRING INSTRUCTIONS
3789 012602 000000      0          ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3790 012604 012767 016310 166044      MOV      #BUF1, TABLE
3791 012612 012767 012704 166050      MOV      #SCN,PCI  ;STORE PC OF TEST INSTRUCTION
3792 012620 012777 014506 166036      MOV      #INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
3793 012626 005077 166034      CLR      @TPSW     ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
3794 012632 004767 001624      JSR      PC,TDONE
3795 012636 013777 000554 166016      MOV      @#NULL,@TBUF ;SEND CARRIAGE RETURN

```

3796	012644	004567	001474		JSR	R5,XPSW	:STORE EXPECTED PSW VALUE
3797	012650	000004			.WORD	04	
3798	012652	106427	000000		MTPS	#0	:SET PSW TO ALLOW INTERRUPTS
3799	012656	052777	000100	165774	BIS	#100,@TCSR	:ENABLE TTY INTERRUPTS
3800	012664	004767	001314		REPSCN: JSR	PC,GENR	:SET UP GENERAL REGISTERS
3801	012670	010637	000676		MOV	SP,@SAVR6	:COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
3802	012674	016705	165756		MOV	TABLE,R5	
3803	012700	000244			CLZ		:CLEAR "Z" CONDITION CODE
3804	012702	000267			+SEV!SEZ!SEC		:SET ALL OTHER CONDITION CODES
3805							
3806	012704	076042			SCN:	SCANC	:EXECUTE "SCANC"
3807							
3808							:CHECK RESULTS
3809	012706	106767	165760		MFPS	CCODES	:STORE THE PSW
3810	012712	032777	000100	165740	BIT	#100,@TCSR	:IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
3811	012720	001361			BNE	REPSCN	:BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
3812	012722	042767	177400	165742	BIC	#177400,CCODES	:CLEAR ALL UNUSED BITS
3813	012730	023767	000674	165754	CMP	@EXPPSW,CCODES	:CHECK PSW AGAINST EXPECTED VALUE
3814	012736	001401			BEQ	64\$:BR, IF EQUAL
3815	012740	104001			ERROR	1	:*****TEST 45 - ERROR 1*****
3816							:PSW ERROR
3817							:EXPECTED PSW IS STORED AT "EXPPSW"
3818							:ACTUAL PSW IS STORED AT "CCODES"
3819	012742				64\$:		
3820	012742	023706	000676		CMP	@SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
3821	012746	001403			BEQ	65\$:BR IF OK
3822	012750	010637	000700		MOV	SP,@BADR6	:STORE BAD SP VALUE
3823	012754	104002			ERROR	2	:*****TEST 45 - ERROR 2*****
3824							:STACK POINTER NOT RESTORED BY INSTRUCTION
3825							:EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3826							:ERRONEOUS VALUE IS AT "BADR6"
3827	012756				65\$:		
3828	012756	026704	165672		CMP	FILL,R4	:CHECK R4 UNCHANGED
3829	012762	001401			BEQ	66\$:BR IF OK
3830	012764	104003			ERROR	3	:*****TEST 45 - ERROR 3*****
3831							:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3832	012766	026705	165664		66\$:	CMP	TABLE,R5
3833	012772	001401			BEQ	67\$:CHECK R5 UNCHANGED
3834	012774	104004			ERROR	4	:BR IF OK
3835							:*****TEST 45 - ERROR 4*****
3836	012776				67\$:		:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3837	012776	016703	165644		MOV	SRCAD,R3	:POINT R3 TO SOURCE STRING
3838	013002	016702	165636		MOV	SRCLN,R2	:STORE SOURCE LENGTH IN R2
3839	013006	111304			MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
3840	013010	042704	177400		BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
3841	013014	136467	016310	165632	BITB	BUF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
3842	013022	001003			BNE	69\$	
3843	013024	105723			TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
3844	013026	005302			DEC	R2	:DECREMENT BYTE COUNT
3845	013030	001366			BNE	68\$:BR, IF NOT FINISHED
3846	013032	020200			69\$:	CMP	R2,R0
3847	013034	001401			BEQ	70\$:CHECK R0=REMAINING SOURCE LENGTH
3848	013036	104005			ERROR	5	:*****TEST 45 - ERROR 5*****
3849							:SCAN LENGTH ERROR
3850	013040	020301			70\$:	CMP	R3,R1
3851	013042	001401			BEQ	71\$:CHECK R1=ADDRESS OF CHARACTER


```

3852 013044 104006          ERROR 6          :*****TEST 45 - ERROR 6*****
3853                                     :SCAN ADDRESS ERROR
3854 013046          71$:
3855
3856 013046 106427 000200 40$: MTPS #200          :RESTORE PSW TO PRIORITY 7
3857 013052 016777 165610 165604 MOV TPSW,@TVECT :RESTORE TRAP CATCHER
3858
3859

```

```

3860 :*****
3861 :*TEST 46 TEST "SPANC" WITH NON-MEMBER CHARACTER IN SOURCE, SL .GT. 0 AT CHARACTER
3862 :*****
3863 :*THIS TEST VERIFIES THAT "SPANC" INDICATES A NON-MEMBER
3864 :*FOUND.
3865 :*THE RESULT IS R1 EQUALS ADDRESS OF NON-MEMBER CHARACTER, R0
3866 :*EQUALS THE REMAINING SOURCE LENGTH, AND ALL
3867 :*CONDITION CODES CLEAR.
3868 :*****
3869 :*****

```

```

3870 013060 000004          TST46: SCOPE
3871 013062 004567 001162 JSR R5,PREP          :SET UP INSTRUCTION ARGUMENTS
3872 013066 000020          :SOURCE LENGTH
3873 013070 016310          :SOURCE ADDRESS
3874 013072 177777          :STORE NON-ZERO VALUES TO TEST R2 & R3
3875 013074 177777          : UNAFFECTED BY DIS STRING INSTRUCTIONS
3876 013076 000007          :SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3877 013100 012767 016310 165550 MOV #BUF1,TABLE
3878 013106 004567 001232 JSR R5,XPSW          :STORE EXPECTED PSW VALUE
3879 013112 000200          .WORD 200
3880 013114 004767 001064 JSR PC,GENR          :SET UP GENERAL REGISTERS
3881 013120 016705 165532 MOV TABLE,R5
3882 013124 000244          CLZ
3883 013126 000267          +SEV!SEZ!SEC
3884
3885 013130 076043          SPANC
3886
3887 013132 004767 001232 JSR PC,CKCC          :CHECK RESULTS
3888 013136 001401          :CHECK PSW, GENERATE CONDITION CODES
3889 013140 104001          BEQ 64$
3890
3891          ERROR 1          :*****TEST 46 - ERROR 1*****
3892          :PSW ERROR
3893          :EXPECTED PSW IS STORED AT 'EXPPSW'
3894          :ACTUAL PSW IS STORED AT 'CCODES'

```

```

3893 013142          64$:
3894 013142 023706 000676 CMP @SAVR6,SP          :VERIFY STACK POINTER IS RESTORED
3895 013146 001403          BEQ 65$              :BR IF OK
3896 013150 010637 000700 MOV SP,@BADR6          :STORE BAD SP VALUE
3897 013154 104002          ERROR 2          :*****TEST 46 - ERROR 2*****
3898          :STACK POINTER NOT RESTORED BY INSTRUCTION
3899          :EXPECTED VALUE OF SP IS STORED AT 'SAVR6'
3900          :ERRONEOUS VALUE IS AT 'BADR6'

```

```

3901 013156          65$:
3902 013156 026704 165472 CMP FILL,R4           :CHECK R4 UNCHANGED
3903 013162 001401          BEQ 66$              :BR IF OK
3904 013164 104003          ERROR 3          :*****TEST 46 - ERROR 3*****
3905          :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION 'FILL'
3906 013166 026705 165464 66$: CMP TABLE,R5          :CHECK R5 UNCHANGED
3907 013172 001401          BEQ 67$              :BR IF OK

```

```

3908 013174 104004          ERROR 4          ;*****TEST 46 - ERROR 4*****
3909                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3910 013176          67$:          MOV SRCAD,R3          ;POINT R3 TO SOURCE STRING
3911 013176 016703 165444      MOV SRCLN,R2          ;STORE SOURCE LENGTH IN R2
3912 013202 016702 165436      MOV (R3),R4          ;USE SOURCE BYTE AS INDEX
3913 013206 111304          68$:      MOV B (R3),R4          ;CLEAR ANY BYTE SIGN EXTENSION
3914 013210 042704 177400      BIC #177400,R4          ;"AND" TABLE ENTRY WITH MASK
3915 013214 136467 016310 165432 BITB BUF1(R4),FILL
3916 013222 001403          BEQ 69$
3917 013224 105723          TSTB (R3)+          ;POINT TO NEXT SOURCE BYTE
3918 013226 005302          DEC R2          ;DECREMENT BYTE COUNT
3919 013230 001366          BNE 68$          ;BR, IF NOT FINISHED
3920 013232 020200          69$:      CMP R2,R0          ;CHECK R0=REMAINING SOURCE LENGTH
3921 013234 001401          BEQ 70$
3922 013236 104005          ERROR 5          ;*****TEST 46 - ERROR 5*****
3923                                     ;SPAN LENGTH ERROR
3924 013240 020301          70$:      CMP R3,R1          ;CHECK R1=ADDRESS OF CHARACTER
3925 013242 001401          BEQ 71$
3926 013244 104006          ERROR 6          ;*****TEST 46 - ERROR 6*****
3927                                     ;SPAN ADDRESS ERROR
3928 013246          71$:
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939

```

```

:*****
:*TEST 47 TEST "SPANC" FAILS ONLY MEMBER CHARACTERS IN SOURCE
:*****
:*PROPER TERMINATION OF THIS TEST IS R0=0, R1 EQUALS
:*ADDRESS+1 OF LSB OF SOURCE, AND ALL CONDITION CODES
:*CLEAR EXCEPT Z=1
:*****

```

```

3940 013246 000004          TST47: SCOPE
3941 013250 004567 000774      JSR R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
3942 013254 000100          100          ;SOURCE LENGTH
3943 013256 016310          BUF1          ;SOURCE ADDRESS
3944 013260 177777          NXM          ;STORE NON-ZERO VALUES TO TEST R2 & R3
3945 013262 177777          NXM          ; UNAFFECTED BY DIS STRING INSTRUCTIONS
3946 013264 000377          377          ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
3947 013266 012767 016310 165362 MOV #BUF1, TABLE
3948 013274 004567 001044      JSR R5,XPSW          ;STORE EXPECTED PSW VALUE
3949 013300 000204          .WORD 204
3950 013302 004767 000676      JSR PC,GENR          ;SET UP GENERAL REGISTERS
3951 013306 016705 165344      MOV TABLE,R5
3952 013312 000244          CLZ          ;CLEAR "Z" CONDITION CODE
3953 013314 000267          +SEV!SEZ!SEC      ;SET ALL OTHER CONDITION CODES
3954                                     ;EXECUTE "SPANC"
3955 013316 076043          SPANC
3956                                     ;CHECK RESULTS
3957 013320 004767 001044      JSR PC,CKCC          ;CHECK PSW, GENERATE CONDITION CODES
3958 013324 001401          BEQ 64$
3959 013326 104001          ERROR 1          ;*****TEST 47 - ERROR 1*****
3960                                     ;PSW ERROR
3961                                     ;EXPECTED PSW IS STORED AT "EXPPSW"
3962                                     ;ACTUAL PSW IS STORED AT "CCODES"
3963 013330          64$:

```



```

4020 013470 001134          BNE      $EOP          ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST
4021 013472          SPCCONT: JSR      R5,PREP        ;SET UP INSTRUCTION ARGUMENTS
4022 013472 004567 000552          400          ;SOURCE LENGTH
4023 013476 000400          BUF1         ;SOURCE ADDRESS
4024 013500 016310          NXM          ;DESTINATION LENGTH
4025 013502 177777          NXM          ;DESTINATION ADDRESS
4026 013504 177777          177         ;FILL CHARACTER
4027 013506 000177          MOV      #BUF1, TABLE ;STORE PC OF TEST INSTRUCTION
4028 013510 012767 016310 165140  MOV      #SPN, PC1     ;POINT TTY VECTOR TO INTERRUPT ROUTINE
4029 013516 012767 013606 165144  MOV      #INTR,@TVECT ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
4030 013524 012777 014506 165132  CLR      @TPSW
4031 013532 005077 165130          JSR      PC,TDONE
4032 013536 004767 000720          MOV      @#NULL,@TBUF ;SEND CARRIAGE RETURN
4033 013542 013777 000554 165112  JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
4034 013550 004567 000570          .WORD    00
4035 013554 000000          MTPS      #0          ;SET PSW TO ALLOW INTERRUPTS
4036 013556 106427 000000          BIS      #100,@TCSR  ;ENABLE TTY INTERRUPTS
4037 013562 052777 000100 165070  JSR      PC,GENR      ;SET UP GENERAL REGISTERS
4038 013570 004767 000410          MOV      SP,@SAVR6   ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
4039 013574 010637 000676          MOV      TABLE,R5
4040 013600 016705 165052          SCC
4041 013604 000277          ;SET ALL CONDITION CODES
4042
4043 013606 076043          SPN:      SPANC      ;EXECUTE "SPANC"
4044
4045          ;CHECK RESULTS
4046 013610 106767 165056          MFPS      CCODES     ;STORE THE PSW
4047 013614 032777 000100 165036  BIT      #100,@TCSR  ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
4048 013622 001362          BNE      REPSPN     ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
4049 013624 042767 177400 165040  BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
4050 013632 023767 000674 165032  CMP      @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
4051 013640 001401          BEQ      64$
4052 013642 104001          ERROR    1          ;BR, IF EQUAL
4053          ;*****TEST 50 - ERROR 1*****
4054          ;PSW ERROR
4055          ;EXPECTED PSW IS STORED AT "EXPPSW"
4056          ;ACTUAL PSW IS STORED AT "CCODES"
4056 013644          64$:
4057 013644 023706 000676          CMP      @SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
4058 013650 001403          BEQ      65$
4059 013652 010637 000700          MOV      SP,@BADR6  ;STORE BAD SP VALUE
4060 013656 104002          ERROR    2          ;*****TEST 50 - ERROR 2*****
4061          ;STACK POINTER NOT RESTORED BY INSTRUCTION
4062          ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
4063          ;ERRONEOUS VALUE IS AT "BADR6"
4064 013660          65$:
4065 013660 026704 164770          CMP      FILL,R4    ;CHECK R4 UNCHANGED
4066 013664 001401          BEQ      66$
4067 013666 104003          ERROR    3          ;BR IF OK
4068          ;*****TEST 50 - ERROR 3*****
4069 013670 026705 164762          66$:  CMP      TABLE,R5  ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
4070 013674 001401          BEQ      67$
4071 013676 104004          ERROR    4          ;CHECK R5 UNCHANGED
4072          ;BR IF OK
4073          ;*****TEST 50 - ERROR 4*****
4074          ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
4074 013700 016703 164742          67$:  MOV      SRCAD,R3   ;POINT R3 TO SOURCE STRING
4075 013704 016702 164734          MOV      SRCLN,R2  ;STORE SOURCE LENGTH IN R2
    
```


MAIN. MACY11 30(1046) 22-JAN-82 08:44 PAGE 77
 CVKAIB.P11 22-JAN-82 08:43 END OF PASS ROUTINE

SEQ 0076

```

4132 014066 032777 010000 164444      BIT      #BIT12,@SWR      ;;RUN WITH TRACE TRAP?
4133 014074 001005                      BNE      1$            ;;BR IF NO
4134 014076 005167 000020                      COM      $TBIT         ;;IS IT TIME FOR TRACE TRAP
4135 014102 100402                      BMI      1$            ;;BR IF NO
4136 014104 052716 000020                      BIS      #20,(SP)     ;;SET TRACE TRAP
4137 014110 012746 014116      1$:      MOV      #SLOOP,-(SP) ;;JUMP TO START OF TEST
4138 014114 000002      SRTRN:  RTI                      ;;RETURN--THIS IS CHANGED TO
4139                                     ;;AN 'RTT' IF 'RTT' IS A LEGAL
4140                                     ;;INSTRUCTION
4141 014116      SLOOP:                      ;;
4142 014116 000137                      JMP      @PC+         ;;RETURN
4143 014120 001330      SRTNAD: .WORD    BEGIN
4144 014122 000000      STBIT:  .WORD    0      ;;'T' BIT STATE INDICATOR
4145 014124 377 377 000      SNULL:  .BYTE    -1,-1,0 ;;NULL CHARACTER STRING
4146 014130 014130                      .EVEN
4147 014130 005015 047105 020104      ENDMSG: .ASCIZ  <15><12>/END PASS/
4148 014136 040520 051523 000
4149 014144 014144                      .EVEN
4150
4151
4152      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 0
4153 014144 011637 000702      TZERO:  MOV      (SP),@#OLDPC ;;GET PC+2 WHERE UNEXPECTED TRAP OCCURRED
4154 014150 104200      ERROR  200 ;;*****ERROR 200*****
4155                                     ;;UNEXPECTED TRAP TO LOCATION 0
4156                                     ;;'OLDPC' CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4157 014152 000000      HALT      ;;PROGRAM MUST BE RESTARTED AT THIS POINT
4158
4159      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 4
4160 014154 011637 000702      TIMTRP: MOV      (SP),@#OLDPC ;;GET PC+2 WHERE UNEXPECTED TIMEOUT TRAP OCCURRED
4161 014160 104204      ERROR  204 ;;*****ERROR 204*****
4162                                     ;;UNEXPECTED TRAP TO LOCATION 4
4163                                     ;;'OLDPC' CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4164 014162 000000      HALT      ;;PROGRAM MUST BE RESTARTED AT THIS POINT
4165
4166      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 10
4167 014164 011637 000702      ILLTRP: MOV      (SP),@#OLDPC ;;GET PC+2 WHERE UNEXPECTED ILLEGAL INSTRUCTION TRAP OCCU
4168 014170 104210      ERROR  210 ;;*****ERROR 210*****
4169                                     ;;UNEXPECTED TRAP TO LOCATION 10
4170                                     ;;'OLDPC' CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4171 014172 000000      HALT      ;;PROGRAM MUST BE RESTARTED AT THIS POINT
4172
4173      ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 20
4174 014174 011637 000702      IOTTRP: MOV      (SP),@#OLDPC ;;GET PC+2 WHERE UNEXPECTED IOT INSTRUCTION TRAP OCCURRED
4175 014200 104220      ERROR  220 ;;*****ERROR 220*****
4176                                     ;;UNEXPECTED TRAP TO LOCATION 20
4177                                     ;;'OLDPC' CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4178 014202 000000      HALT      ;;PROGRAM MUST BE RESTARTED AT THIS POINT
4179
4180
4181      ;SUBROUTINE TO SETUP GENERAL REGISTERS FOR INSTRUCTION EXECUTION
4182 014204 016700 164434      GENR:  MOV      SRCLN,R0 ;;STORE SOURCE LENGTH
4183 014210 016701 164432      MOV      SRCAD,R1    ;;STORE SOURCE ADDRESS
4184 014214 016702 164430      MOV      DSTLN,R2    ;;STORE DESTINATION LENGTH
4185 014220 016703 164426      MOV      DSTAD,R3    ;;STORE DESTINATION ADDRESS
4186 014224 016704 164424      MOV      FILL,R4     ;;STORE FILL CHARACTER
4187 014230 016705 164422      MOV      TABLE,R5  ;

```



```

4188 014234 010637 000676      MOV      SP,@#SAVR6      ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
4189 014240 062737 000002 000676  ADD      #2,@#SAVR6      ;ADJUST SAVED SP BECAUSE OF JSR TO THIS ROUTINE
4190 014246 000207      RTS      PC
4191
4192      ;SUBROUTINE TO PREPARE INSTRUCTION PARAMETERS
4193 014250 012567 164370  PREP:    MOV      (R5)+,SRCLN      ;SET SOURCE LENGTH
4194 014254 012567 164366      MOV      (R5)+,SRCAD      ;SET SOURCE ADDRESS
4195 014260 012567 164364      MOV      (R5)+,DSTLN      ;SET DESTINATION LENGTH
4196 014264 012567 164362      MOV      (R5)+,DSTAD      ;SET DESTINATION ADDRESS
4197 014270 012567 164360      MOV      (R5)+,FILL      ;SET FILL CHARACTER
4198 014274 000205      RTS      R5      ;RETURN
4199
4200      ;SUBROUTINE TO CLEAR DESTINATION & BOUNDARY BYTES
4201      CLDST:  MOV      DSTAD,R0      ;POINT R0 TO DESTINATION AREA
4202 014276 016700 164350      DEC      R0      ;CLEAR LOWER BOUNDARY BYTE OF DEST
4203 014302 005300      CLRB     (R0)+
4204 014304 105020      MOV      DSTLN,R1      ;STORE DEST. BYTE COUNT IN R1
4205 014306 016701 164336  CONTCL:  CLRB     (R0)+      ;CLEAR DESTINATION
4206 014312 105020      DEC      R1      ;DECREMENT DEST. BYTE COUNT
4207 014314 005301      BNE     CONTCL      ;BR, IF NOT FINISHED
4208 014316 001375      CLRB     (R0)+      ;CLEAR UPPER BYTE BOUNDARY OF DEST
4209 014320 105020      RTS      PC      ;RETURN
4210 014322 000207
4211
4212      ;SUBROUTINE TO GENERATE A SOURCE STRING
4213 014324 012700 016310  GENSRC:  MOV      #BUF1,R0      ;POINT R0 TO THE PATTERN STORED
4214 014330 012502      MOV      (R5)+,R2      ;STORE SOURCE BYTE COUNT IN R2
4215 014332 012501      MOV      (R5)+,R1      ;POINT R1 TO THE SOURCE ADDRESS
4216 014334 112021  GENCON:  MOVVB   (R0)+,(R1)+      ;TRANSFER PATTERN TO SOURCE
4217 014336 005302      DEC      R2      ;DECREMENT BYTE COUNT
4218 014340 001375      BNE     GENCON      ;BR, IF NOT FINISHED
4219 014342 000205      RTS      R5      ;RETURN
4220
4221      ;SUBROUTINE TO RECORD EXPECTED PSW
4222      XPSW:  MOV      (R5)+,@#EXPPSW ;STORE EXPECTED PSW VALUE
4223 014344 012537 000674      MFPS    R0
4224 014350 106700      BIT     #TBIT,R0
4225 014352 032700 000020      BEQ     1$
4226 014356 001403      BIS     #TBIT,@#EXPPSW ;OTHERWISE SET T-BIT IN EXPECTED PSW VALUE
4227 014360 052737 000020 000674  1$:     RTS      R5
4228 014366 000205
4229
4230      ;SUBROUTINE TO CHECK THE PSW
4231      ;**NOTE: CONDITION CODES GENERATED ARE TO BE PASSED BACK
4232      ;**      TO THE MAIN PROGRAM, THEREFORE NO CODES SHOULD BE
4233      ;**      ADDED HERE THAT WILL ALTER THE CONDITION CODES.
4234
4235      CKCC:  MFPS    CCODES      ;STORE THE PSW
4236 014370 106767 164276      BIC     #177400,CCODES ;CLEAR ALL UNUSED BITS
4237 014374 042767 177400 164270      CMP     @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
4238 014402 023767 000674 164262      RTS      PC      ;RETURN
4239 014410 000207
4240
4241      ;SUBROUTINE TO SET TEMP IF REQUIRE, WHICH INDICATES GO TO NEXT TEST
4242
4243

```

```
4244 :TEMP = 0 => CONTINUE WITH CURRENT TEST
4245 :TEMP = 1 => GO TO DO NEXT TEST
4246 014412 005067 164266 SKPINT: CLR TEMP :CLEAR FLAG
4247 014416 105777 164116 TSTB @SWR :TEST BIT 7 OF SWR
4248 014422 100414 BMI NEXTST :SKIP TO NEXT TEST IF SET
4249 014424 026767 164120 164226 CMP $TPS,TCR :IS SLU USED FOR INTERRUPTS THE CONSOLE?
4250 014432 001007 BNE CONTIN :BR, IF NOT & PREFORM INTERRUPTABILITY TEST
4251 014434 032767 000001 164144 BIT #BIT0,$ENV :IF YES, CHECK IF ON APT
4252 014442 001403 BEQ CONTIN :BR IF NOT UNDER APT, AND DO THIS TEST
4253 014444 005767 164124 TST $PASS :CHECK IF ON FIRST PASS
4254 014450 001001 BNE NEXTST :IF NOT ON FIRST PASS, BR & SKIP THIS TEST
4255 014452 000207 CONTIN: RTS PC :RETURN AND DO THE CURRENT TEST
4256 014454 005267 164224 NEXTST: INC TEMP :SET FLAG, SKIP CURRENT TEST
4257 014460 000207 RTS PC :RETURN AND DO THE NEXT TEST
4258
4259
4260
4261 :SUBROUTINE TO TEST FOR TRANSMIT DONE FLAG
4262 014462 005037 000704 TDONE: CLR @TEMP :CLEAR A TIMER
4263 014466 105777 164166 1$: TSTB @TCR :IS SLU READY?
4264 014472 100404 BMI RETN :BR IF READY
4265 014474 005237 000704 INC @TEMP :OTHERWISE INCREMENT TIMER
4266 014500 001372 BNE 1$ :BR IF NOT TIMED OUT
4267 014502 104300 ERROR 300 :*****ERROR 300*****
4268 :NEVER GOT TRANSMIT DONE FLAG
4269 014504 000207 RETN: RTS PC :RETURN
4270
4271 :SUBROUTINE TO HANDLE TTY INTERRUPTS IN INSTRUCTION
4272 :INTERRUPTABILITY TESTS
4273
4274 014506 INTR:
4275 014506 021667 164156 CMP (SP),PCI :WAS PC AT INSTRUCTION UNDER TEST?
4276 014512 001003 BNE SEND :BR, IF NO
4277 014514 032704 177400 CKR4: BIT #177400,R4 :IF YES, CHECK UPPER BYTE OF R4
4278 014520 001004 BNE CLRINT :IF ZERO, INSTRUCTION WAS NOT INTERRUPTED-TRY AGAIN
4279 014522 013777 000554 164132 SEND: MOV @SNULL,@TBUF :SEND ANOTHER CHARACTER
4280 014530 000002 RTI :RETURN
4281 014532 042777 000100 164120 CLRINT: BIC #100,@TCR :IF NON-ZERO, CLEAR INTERRUPT ENABLE
4282 014540 000002 RTI :CONTINUE INSTRUCTION
4283
4284
4285 .SBTTL SCOPE HANDLER ROUTINE
4286
4287 :*****
4288 :*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4289 :*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4290 :*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4291 :*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4292 :*SW14=1 LOOP ON TEST
4293 :*SW09=1 LOOP ON ERROR
4294 :*SW08=1 LOOP ON TEST IN SWR<5:0>
4295 :*CALL
4296 :* SCOPE ::SCOPE=IOT
4297
4298 014542 $$COPE:
4299 014542 032777 040000 163770 1$: BIT #BIT14,@SWR ;;LOOP ON PRESENT TEST?
```



```

4300 014550 001065          BNE      $OVER          ;; YES IF SW14=1
4301          :*****START OF CODE FOR THE XOR TESTER*****
4302 014552 000416          $XTSTR: BR      6$          ;; IF RUNNING ON THE 'XOR' TESTER CHANGE
4303          ;; THIS INSTRUCTION TO A 'NOP' (NOP=240)
4304 014554 013746 000004          MOV      @#ERRVEC,-(SP)    ;; SAVE THE CONTENTS OF THE ERROR VECTOR
4305 014560 012737 014600 000004          MOV      #5$,@#ERRVEC    ;; SET FOR TIMEOUT
4306 014566 005737 177060          TST      @#177060        ;; TIME OUT ON XOR?
4307 014572 012637 000004          MOV      (SP)+,@#ERRVEC  ;; RESTORE THE ERROR VECTOR
4308 014576 000434          BR      $$VLAD          ;; GO TO THE NEXT TEST
4309 014600 022626          5$: CMP      (SP)+,(SP)+    ;; CLEAR THE STACK AFTER A TIME OUT
4310 014602 012637 000004          MOV      (SP)+,@#ERRVEC  ;; RESTORE THE ERROR VECTOR
4311 014606 000422          BR      7$          ;; LOOP ON THE PRESENT TEST
4312 014610          6$:*****END OF CODE FOR THE XOR TESTER*****
4313 014610 032777 000400 163722          BIT      #BIT08,@SWR      ;; LOOP ON SPEC. TEST?
4314 014616 001407          BEQ      2$          ;; BR IF NO
4315 014620 017746 163714          MOV      @SWR,-(SP)      ;; SET DESIRED TEST NUM. FROM SWR
4316 014624 042716 000300          BIC      #$$WRMK,(SP)    ;; STRIP AWAY UNDESIRED BITS
4317 014630 122667 163646          CMPB     (SP)+,$TSTNM    ;; ON THE RIGHT TEST?
4318 014634 001433          BEQ      $OVER          ;; BR IF YES
4319 014636 105767 163641          2$: TSTB     $ERFLG        ;; HAS AN ERROR OCCURRED?
4320 014642 001412          BEQ      $$VLAD          ;; BR IF NO
4321 014644 032777 001000 163666          BIT      #BIT09,@SWR      ;; LOOP ON ERROR?
4322 014652 001404          BEQ      4$          ;; BR IF NO
4323 014654 016767 163630 163624          7$: MOV      $LPERR,$LPADR  ;; SET LOOP ADDRESS TO LAST SCOPE
4324 014662 000420          BR      $OVER
4325 014664 105067 163613          4$: CLRB     $ERFLG        ;; ZERO THE ERROR FLAG
4326 014670 105267 163606          $$VLAD: INCB    $TSTNM    ;; COUNT TEST NUMBERS
4327 014674 116767 163602 163670          MOVB    $TSTNM,$TESTN   ;; SET TEST NUMBER IN APT MAILBOX
4328 014702 011667 163600          MOV      (SP),$LPADR    ;; SAVE SCOPE LOOP ADDRESS
4329 014706 011667 163576          MOV      (SP),$LPERR    ;; SAVE ERROR LOOP ADDRESS
4330 014712 005067 163642          CLR      $ESCAPE        ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
4331 014716 112767 000001 163571          MOVB    #1,$ERMAX       ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4332 014724 016777 163552 163610          $OVER: MOV      $TSTNM,@DISPLAY ;; DISPLAY TEST NUMBER
4333 014732 016716 163550          MOV      $LPADR,(SP)    ;; FUDGE RETURN ADDRESS
4334 014736 000002          RTI                    ;; FIXES PS

```

.SBTTL POWER DOWN AND UP ROUTINES

```

4335
4336
4337
4338          :*****
4339          :POWER DOWN ROUTINE
4340 014740 012737 015122 000024          $PWRDN: MOV      #5ILLUP,@#PWRVEC ;; SET FOR FAST UP
4341 014746 012737 000340 000026          MOV      #340,@#PWRVEC+2 ;; PRIO:7
4342 014754 010046          MOV      R0,-(SP)        ;; PUSH R0 ON STACK
4343 014756 010146          MOV      R1,-(SP)        ;; PUSH R1 ON STACK
4344 014760 010246          MOV      R2,-(SP)        ;; PUSH R2 ON STACK
4345 014762 010346          MOV      R3,-(SP)        ;; PUSH R3 ON STACK
4346 014764 010446          MOV      R4,-(SP)        ;; PUSH R4 ON STACK
4347 014766 010546          MOV      R5,-(SP)        ;; PUSH R5 ON STACK
4348 014770 017746 163544          MOV      @SWR,-(SP)      ;; PUSH @SWR ON STACK
4349 014774 010667 000126          MOV      SP,$SAVR6       ;; SAVE SP
4350 015000 012737 015012 000024          MOV      #5PWRUP,@#PWRVEC ;; SET UP VECTOR
4351 015006 000000          HALT
4352 015010 000776          BR      .-2            ;; HANG UP

```

:*****
:POWER UP ROUTINE

4353
4354
4355

```

4356 015012 012737 015122 000024 $PWRUP: MOV    #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWN
4357 015020 016706 000102          MOV    $$AVR5,SP      ;;GET SP
4358 015024 005067 000076          CLR    $$AVR6        ;;WAIT LOOP FOR THE TTY
4359 015030 005267 000072          1$:   INC    $$AVR6        ;;WAIT FOR THE INC
4360 015034 001375          BNE    1$           ;;OF WORD
4361 015036 005067 163440          CLR    $TSTNM
4362 015042 012677 163472          MOV    (SP)+,@SWR   ;;POP STACK INTO @SWR
4363 015046 012605          MOV    (SP)+,R5    ;;POP STACK INTO R5
4364 015050 012604          MOV    (SP)+,R4    ;;POP STACK INTO R4
4365 015052 012603          MOV    (SP)+,R3    ;;POP STACK INTO R3
4366 015054 012602          MOV    (SP)+,R2    ;;POP STACK INTO R2
4367 015056 012601          MOV    (SP)+,R1    ;;POP STACK INTO R1
4368 015060 012600          MOV    (SP)+,R0    ;;POP STACK INTO R0
4369 015062 012737 014740 000024          MOV    #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
4370 015070 012737 000340 000026          MOV    #340,@#PWRVEC+2 ;;PRIO:7
4371 015076 104401          TYPE   $POWER      ;;REPORT THE POWER FAILURE
4372 015100 015130          $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
4373 015102 012716          MOV    (PC)+,(SP)  ;;RESTART AT $SLOOP
4374 015104 014116          $PWRAD: .WORD $SLOOP ;;RESTART ADDRESS
4375 015106 042766 000020 000002          BIC    #20,2(SP)   ;;CLEAR 'T' BIT
4376 015114 005067 177002          CLR    $TBIT      ;;CLEAR THE 'T' BIT FLAG
4377 015120 000002          RTI
4378 015122 000000          $ILLUP: HALT      ;;THE POWER UP SEQUENCE WAS STARTED
4379 015124 000776          BR     .-2        ;; BEFORE THE POWER DOWN WAS COMPLETE
4380 015126 000000          $$AVR6: 0        ;;PUT THE SP HERE
4381 015130 005015 047520 042527          $POWER: .ASCIZ <15><12>'POWER'
4382 015136 000122          .EVEN
4383          .EVEN
4384
4385          .SBTTL ERROR HANDLER ROUTINE
4386
4387
4388          .SBTTL TYPE ROUTINE
4389
4390          ;*****
4391          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
4392          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4393          ;*AND TYPE OUT THE PC OF THE ERROR INSTRUCTION
4394          ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4395          ;*SW15=1      HALT ON ERROR
4396          ;*SW13=1      INHIBIT ERROR TYPEOUTS
4397          ;*SW09=1     LOOP ON ERROR
4398          ;*CALL
4399          ;*      ERROR  N      ;ERROR=EMT AND N=ERROR ITEM NUMBER
4400          ;*
4401          ;:*****
4402
4403          $ERROR:
4404          7$:   INCB   $ERFLG      ;SET THE ERROR FLAG
4405          BEQ    7$           ;DON'T LET THE FLAG GO TO ZERO
4406          MOV    $TSTNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
4407          INC    $ERTTL      ;INC THE ERROR COUNT
4408          MOV    (SP),$ERRPC    ;GET ADDRESS OF ERROR INSTRUCTION
4409          SUB    #2,$ERRPC
4410          MOV    @SERRPC,$ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
4411          MOVB  @SERRPC,$ITEMB

```



```

4412 015200 032777 020000 163332      BIT      #BIT13,@SWR      :SKIP TYPEOUT IF SET
4413 015206 001010                    BNE      20$          :SKIP TYPEOUTS
4414 015210 104401 000563                    TYPE     $CRLF
4415 015214 016746 163276                    MOV      $ERRPC,-(SP)  :SAVE $ERRPC FOR TYPEOUT
4416                                     :ERROR ADDRESS
4417 015220 004767 000444                    JSR      PC,TYPEOUT  :GO TYPE--OCTAL ASCII(ALL DIGITS)
4418 015224 104401 000563                    TYPE     $CRLF
4419 015230                    20$:
4420 015230 122767 000001 163350      CMPB     #APTENV,$ENV  :RUNNING IN APT MODE
4421 015236 001011                    BNE      2$          :NO,SKIP APT ERROR REPORT
4422 015240 005767 163322                    21$:      TST      $MSGTYPE    :FINISHED LAST MESSAGE?
4423 015244 001375                    BNE      21$        :IF NOT, WAIT
4424 015246 116767 163242 163314      MOVB     $ITEMB,$FATAL :REPORT ERROR NUMBER TO APT
4425 015254 005267 163306                    INC      $MSGTYPE    :TELL APT TO TAKE ERROR
4426 015260 000777                    22$:      BR       22$        :APT ERROR LOOP
4427 015262 005777 163252                    2$:      TST      @SWR      :HALT ON ERROR
4428 015266 100001                    BPL      3$          :SKIP IF CONTINUE
4429 015270 000000                    HALT     :HALT ON ERROR!
4430 015272 032777 001000 163240      3$:      BIT      #BIT09,@SWR  :LOOP ON ERROR SWITCH SET?
4431 015300 001402                    BEQ      4$          :BR IF NO
4432 015302 016716 163202                    MOV      $LPERR,(SP)  :FUDGE RETURN FOR LOOPING
4433 015306 005767 163246                    4$:      TST      $ESCAPE   :CHECK FOR AN ESCAPE ADDRESS
4434 015312 001402                    BEQ      5$          :BR IF NONE
4435 015314 016716 163240                    MOV      $ESCAPE,(SP) :FUDGE RETURN ADDRESS FOR ESCAPE
4436 015320                    5$:
4437 015320 022737 014050 000042      CMP      #SENDAD,@#42 :ACT-11 AUTO-ACCEPT?
4438 015326 001001                    BNE      6$          :BRANCH IF NO
4439 015330 000000                    HALT     :YES
4440 015332                    6$:
4441 015332 000002                    RTI      :RETURN

```

.SBTTL TYPE ROUTINE

```

*****
:ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
:THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
:NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
:NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
:NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

```

```

:CALL:
:1) USING A TRAP INSTRUCTION
:* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
:*OR
:* TYPE
:* MESADR
:*

```

```

4459 015334 105767 163217      $TYPE: TSTB     $TPFLG    ;;IS THERE A TERMINAL?
4460 015340 100002                    BPL      1$          ;;BR IF YES
4461 015342 000000                    HALT     ;;HALT HERE IF NO TERMINAL
4462 015344 000430                    BR       3$          ;;LEAVE
4463 015346 010046                    1$:      MOV      RO,-(SP)    ;;SAVE RO
4464 015350 017600 000002      MOV      @2(SP),RO    ;;GET ADDRESS OF ASCIZ STRING
4465 015354 122767 000001 163224      CMPB     #APTENV,$ENV  ;;RUNNING IN APT MODE
4466 015362 001011                    BNE      62$        ;;NO,GO CHECK FOR APT CONSOLE
4467 015364 132767 000100 163215      BITB     #APTPOOL,$ENVM ;;SPOOL MESSAGE TO APT

```



```

4524 015620          10$:          TSTB    @STPS          ;;WAIT UNTIL PRINTER IS READY          :MJD001
4525 015620 105777 162724          BPL     10$
4526 015624 100375          MOVB    2(SP),@STPB          ;;LOAD CHAR TO BE TYPED INTO DATA REG.          :MJD001
4527 015626 116677 000002 162716  CMPB    #CR,2(SP)          ;;IS CHARACTER A CARRIAGE RETURN?
4528 015634 122766 000015 000002          BNE     1$          ;;BRANCH IF NO
4529 015642 001003          CLRB    $CHARCNT          ;;YES--CLEAR CHARACTER COUNT
4530 015644 105067 000014          BR      $TYPEX          ;;EXIT
4531 015650 000406          CMPB    #LF,2(SP)          ;;IS CHARACTER A LINE FEED?
4532 015652 122766 000012 000002 1$:    BEQ     $TYPEX          ;;BRANCH IF YES
4533 015660 001402          INCB    (PC)+          ;;COUNT THE CHARACTER
4534 015662 105227          $CHARCNT: .WORD 0          ;;CHARACTER COUNT STORAGE
4535 015664 000000          $TYPEX: RTS    PC
4536 015666 000207

```

```

4537
4538 *****
4539 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4540 *OCTAL (ASCII) NUMBER AND TYPE IT.
4541 *CALL:

```

```

4542 *      MOV    NUM,-(SP)          ;NUMBER TO BE TYPED
4543 *      JSR    PC,TYPEOUT        ;CALL FOR TYPEOUT

```

```

4544 *****
4545

```

```

4546
4547 015670 112767 000005 000070  TYPOCT: MOVB    #5,$OCNT          ;SET ITERATION COUNT
4548 015676 010446          MOV     R4,-(SP)          ;SAVE R4
4549 015700 010546          MOV     R5,-(SP)          ;SAVE R5
4550 015702 016605 000006          MOV     6(SP),R5          ;PICKUP THE INPUT NUMBER
4551 015706 005004          CLR     R4          ;CLEAR THE OUTPUT WORD
4552 015710 006105 1$:    ROL     R5          ;ROTATE MSB INTO 'C'
4553 015712 000404          BR      3$          ;GO DO MSB
4554 015714 006105 2$:    ROL     R5          ;FORM THIS DIGIT
4555 015716 006105          ROL     R5
4556 015720 006105          ROL     R5
4557 015722 010504          MOV     R5,R4
4558 015724 006104 3$:    ROL     R4          ;GET LSB OF THIS DIGIT
4559 015726 042704 177770          BIC     #177770,R4          ;GET RID OF JUNK
4560 015732 052704 000060          BIS     #'0,R4          ;MAKE THIS DIGIT ASCII
4561 015736 110467 000022          MOVB    R4,8$          ;SAVE FOR TYPING
4562 015742 104401 015764          TYPE    8$          ;GO TYPE THIS DIGIT
4563 015746 105367 000014 7$:    DECB    $OCNT          ;COUNT BY 1
4564 015752 002360          BGE     2$          ;BR IF MORE TO DO
4565 015754 012605 6$:    MOV     (SP)+,R5          ;RESTORE R5
4566 015756 012604          MOV     (SP)+,R4          ;RESTORE R4
4567 015760 012616          MOV     (SP)+,(SP)          ;SET THE STACK FOR RETURNING
4568 015762 000207          RTS     PC          ;RETURN
4569 015764 000          .BYTE 0          ;STORAGE FOR ASCII DIGIT
4570 015765 000          .BYTE 0          ;TERMINATOR FOR TYPE ROUTINE
4571 015766 000000          $OCNT: .WORD 0          ;OCTAL DIGIT COUNTER
4572          .SBTTL  APT COMMUNICATIONS ROUTINE

```

```

4573 *****
4574
4575 015770 112767 000001 000236  $ATY1: MOVB    #1,$FFLG          ;;TO REPORT FATAL ERROR
4576 015776 112767 000001 000226  $ATY3: MOVB    #1,$MFLG          ;;TO TYPE A MESSAGE
4577 016004 000403          BR      $ATYC
4578 016006 112767 000001 000220  $ATY4: MOVB    #1,$FFLG          ;;TO ONLY REPORT FATAL ERROR
4579 016014          $ATYC:

```

```

4580 016014 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
4581 016016 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
4582 016020 105767 000206    TSTB    $MFLG         ;;SHOULD TYPE A MESSAGE?
4583 016024 001450      BEQ     5$            ;;IF NOT: BR
4584 016026 122767 000001 162552    CMPB    #APTENV,$ENV  ;;OPERATING UNDER APT?
4585 016034 001031      BNE     3$            ;;IF NOT: BR
4586 016036 132767 000100 162543    BITB    #APTPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
4587 016044 001425      BEQ     3$            ;;IF NOT: BR
4588 016046 017600 000004      MOV     @4(SP),R0     ;;GET MESSAGE ADDR.
4589 016052 062766 000002 000004    ADD     #2,4(SP)     ;;BUMP RETURN ADDR.
4590 016060 005767 162502 1$:      TST     $MSGTYPE     ;;SEE IF DONE W/ LAST XMISSION?
4591 016064 001375      BNE     1$            ;;IF NOT: WAIT
4592 016066 010067 162510      MOV     R0,$MSGAD    ;;PUT ADDR IN MAILBOX
4593 016072 105720 2$:      TSTB    (R0)+        ;;FIND END OF MESSAGE
4594 016074 001376      BNE     2$
4595 016076 166700 162500      SUB     $MSGAD,R0    ;;SUB START OF MESSAGE
4596 016102 006200      ASR     R0            ;;GET MESSAGE LNTH IN WORDS
4597 016104 010067 162474      MOV     R0,$MSGGLT   ;;PUT LENGTH IN MAILBOX
4598 016110 012767 000004 162450    MOV     #4,$MSGTYPE  ;;TELL APT TO TAKE MSG.
4599 016116 000413      BR      5$
4600 016120 017667 000004 000016 3$:      MOV     @4(SP),4$    ;;PUT MSG ADDR IN JSR LINKAGE
4601 016126 062766 000002 000004    ADD     #2,4(SP)     ;;BUMP RETURN ADDRESS
4602 016134 016746 161636      MOV     177776,-(SP) ;;PUSH 177776 ON STACK
4603 016140 004767 177170      JSR     PC,$TYPE    ;;CALL TYPE MACRO
4604 016144 000000 4$:      .WORD  0
4605 016146 5$:
4606 016146 105767 000062 10$:      TSTB    $FFLG        ;;SHOULD REPORT FATAL ERROR?
4607 016152 001416      BEQ     12$          ;;IF NOT: BR
4608 016154 005767 162426    TST     $ENV         ;;RUNNING UNDER APT?
4609 016160 001413      BEQ     12$          ;;IF NOT: BR
4610 016162 005767 162400 11$:      TST     $MSGTYPE     ;;FINISHED LAST MESSAGE?
4611 016166 001375      BNE     11$          ;;IF NOT: WAIT
4612 016170 017667 000004 162372    MOV     @4(SP),$FATAL ;;GET ERROR #
4613 016176 062766 000002 000004    ADD     #2,4(SP)     ;;BUMP RETURN ADDR.
4614 016204 005267 162356      INC     $MSGTYPE     ;;TELL APT TO TAKE ERROR
4615 016210 105067 000020 12$:      CLRB    $FFLG        ;;CLEAR FATAL FLAG
4616 016214 105067 000013      CLRB    $LFLG        ;;CLEAR LOG FLAG
4617 016220 105067 000006      CLRB    $MFLG        ;;CLEAR MESSAGE FLAG
4618 016224 012601      MOV     (SP)+,R1     ;;POP STACK INTO R1
4619 016226 012600      MOV     (SP)+,R0     ;;POP STACK INTO R0
4620 016230 000207      RTS     PC           ;;RETURN
4621 016232 000      $MFLG: .BYTE 0      ;;MESSG. FLAG
4622 016233 000      $LFLG: .BYTE 0      ;;LOG FLAG
4623 016234 000      $FFLG: .BYTE 0      ;;FATAL FLAG
4624 016236      .EVEN
4625 000200      APTSIZE=200
4626 000001      APTENV=001
4627 000100      APTPOOL=100
4628 000040      APTCSUP=040
4629      .SBTTL TRAP DECODER
4630
4631      ;;*****
4632      ;;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
4633      ;;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4634      ;;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4635      ;;*GO TO THAT ROUTINE.

```



```

4636
4637 016236 010046          $TRAP:  MOV    R0,-(SP)          ;;SAVE R0
4638 016240 016600 000002  MOV    2(SP),R0          ;;GET TRAP ADDRESS
4639 016244 005740          TST    -(R0)             ;;BACKUP BY 2
4640 016246 111000          MOVB   (R0),R0           ;;GET RIGHT BYTE OF TRAP
4641 016250 006300          ASL    R0                ;;POSITION FOR INDEXING
4642 016252 016000 016272  MOV    $TRPAD(R0),R0     ;;INDEX TO TABLE
4643 016256 000200          RTS     R0              ;;GO TO ROUTINE
4644
4645
4646          ;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
4647
4648 016260 011646          $TRAP2: MOV   (SP),-(SP)    ;;MOVE THE PC DOWN
4649 016262 016666 000004 000002  MOV   4(SP),2(SP)        ;;MOVE THE PSW DOWN
4650 016270 000002          RTI                    ;;RESTORE THE PSW
4651
4652          .SBTTL  TRAP TABLE
4653
4654          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
4655          ;*BY THE 'TRAP' INSTRUCTION.
4656
4657          :          ROUTINE
4658          :          -----
4659 016272 016260          $TRPAD: .WORD  $TRAP2          TRAP+1(104401) TTY TYPEOUT ROUTINE
4660 016274 015334          $TYPE  ;;CALL=TYPE
4661
4662
4663
4664 016276 005015 053103 040513  NAME:  .ASCIZ  <15><12>/CVKAIB/
4665 016304 041111 000          .EVEN
4666 016310
4667
4668
4669 016310 000200          BUF1:  .BLKW  200
4670 016710 000200          BUF2:  .BLKW  200
4671 000001          .END

```


CROSS REFERENCE TABLE -- USER SYMBOLS

DISPRE	000174	515#	741											
DSTAD	000652	679#	1350	2331	2474	2643	2718	2791	2878	4185	4196*	4202		
DSTLN	000650	678#	849	1010	1038	1142	1251	1323	1353	1464	1534	1562	1718	1791
		1819	1821	1915	1998	2026	2028	2124	2248	2327	2501	2644	2812	2879
		4184	4195*	4205										
DSWR =	177570	400#	598											
EMTVEC=	000030	489#	713*	714*										
ENDMSG	014130	4114	4147#											
ENDT1	001534	859	863#											
ENDT10	003554	1572	1576#											
ENDT12	004212	1727	1731#											
ENDT13	004444	1831	1835#											
ENDT14	004674	1937	1941#											
ENDT15	005124	2038	2042#											
ENDT16	005364	2146	2150#											
ENDT17	005722	2270	2274#											
ENDT2	001734	950	954#											
ENDT3	002144	1048	1052#											
ENDT4	002374	1154	1158#											
ENDT5	002634	1263	1267#											
ENDT7	003334	1473	1477#											
ERRVEC=	000004	482#	4304	4305*	4307*	4310*								
EXPPSW	000674	688#	1657	2198	2972	3305	3584	3813	4050	4223*	4227*	4238		
FILL	000654	680#	832	850	927	1000	1115	1145	1225	1254	1313	1437	1466	1524
		1691	1720	1781	1899	1918	1988	2108	2127	2232	2251	2336	2408	2463
		2570	2652	2723	2779	2885	3006	3058	3069	3124	3135	3190	3201	3320
		3331	3390	3401	3435	3461	3472	3560	3599	3610	3669	3682	3740	3753
		3828	3841	3902	3915	3972	3985	4065	4078	4186	4197*			
GENCON	014334	4216#	4218											
GENR	014204	792	889	980	1076	1186	1293	1398	1504	1597	1647	1761	1860	1968
		2069	2187	2298	2371	2443	2530	2603	2685	2758	2843	2913	2960	3038
		3104	3169	3234	3295	3370	3440	3504	3573	3648	3717	3800	3880	3950
		4038	4182#											
GENSRC	014324	1181	1288	1393	1499	1963	2064	2437	2752	2837	4213#			
GNS =	***** U	514	4660											
HT =	000011	392#	4480	4538										
ILLTRP	014164	523	4167#											
INTR	014506	1639	2179	2952	3287	3565	3792	4030	4274#					
IOTTRP	014174	4174#												
IOTVEC=	000020	487#	711*	712*										
LF =	000012	393#	4532	4538										
LOC	011076	3286	3299#											
LOCC =	076040	506#	3041	3107	3173	3238	3299							
MC	003776	1638	1651#											
MOVC =	076030	506#	796	892	983	1080	1190	1296	1402	1507	1651			
MOVRC =	076031	506#	1764	1864	1971	2073	2192							
MRC	005510	2178	2192#											
N =	000007	502#	775#	800	801#	808	809#	815	816#	821	822#	825	826#	829
		830#	834	835#	838	839#	845	846#	849	852	853#	860	861#	867#
		896	897#	904	905#	911	912#	916	917#	920	921#	924	925#	929
		930#	933	934#	940	941#	946	947#	951	952#	959#	987	988#	995
		996#	1002	1003#	1006	1007#	1013	1014#	1018	1019#	1022	1023#	1026	1027#
		1033	1034#	1038	1041	1042#	1049	1050#	1056#	1084	1085#	1092	1093#	1099
		1100#	1104	1105#	1108	1109#	1112	1113#	1117	1118#	1121	1122#	1128	1129#
		1133	1136	1137#	1144	1147	1148#	1155	1156#	1160#	1194	1195#	1202	1203#
		1209	1210#	1214	1215#	1218	1219#	1222	1223#	1227	1228#	1231	1232#	1237

CROSS REFERENCE TABLE -- USER SYMBOLS

1238#	1242	1245	1246#	1253	1256	1257#	1264	1265#	1270#	1300	1301#	1308
1309#	1315	1316#	1319	1320#	1326	1327#	1331	1332#	1335	1336#	1339	1340#
1347	1348#	1356	1357#	1366	1367#	1373#	1406	1407#	1414	1415#	1421	1422#
1426	1427#	1430	1431#	1434	1435#	1439	1440#	1443	1444#	1450	1451#	1455
1458	1459#	1468	1469#	1474	1475#	1481#	1511	1512#	1519	1520#	1526	1527#
1530	1531#	1537	1538#	1542	1543#	1546	1547#	1550	1551#	1557	1558#	1562
1565	1566#	1573	1574#	1580#	1602	1603#	1615#	1659	1660#	1668	1669#	1675
1676#	1680	1681#	1684	1685#	1688	1689#	1693	1694#	1697	1698#	1704	1705#
1709	1712	1713#	1722	1723#	1728	1729#	1739#	1768	1769#	1776	1777#	1783
1784#	1787	1788#	1794	1795#	1799	1800#	1803	1804#	1807	1808#	1814	1815#
1821	1824	1825#	1832	1833#	1839#	1868	1869#	1876	1877#	1883	1884#	1888
1889#	1892	1893#	1896	1897#	1901	1902#	1905	1906#	1911	1912#	1917	1920
1921#	1927	1930	1931#	1938	1939#	1944#	1975	1976#	1983	1984#	1990	1991#
1994	1995#	2001	2002#	2006	2007#	2010	2011#	2014	2015#	2021	2022#	2028
2031	2032#	2039	2040#	2046#	2077	2078#	2085	2086#	2092	2093#	2097	2098#
2101	2102#	2105	2106#	2110	2111#	2114	2115#	2120	2121#	2126	2129	2130#
2136	2139	2140#	2147	2148#	2155#	2200	2201#	2209	2210#	2216	2217#	2221
2222#	2225	2226#	2229	2230#	2234	2235#	2238	2239#	2244	2245#	2250	2253
2254#	2260	2263	2264#	2271	2272#	2280#	2306	2307#	2314	2315#	2321	2322#
2325	2326#	2329	2330#	2333	2334#	2338	2339#	2342	2343#	2348#	2378	2379#
2386	2387#	2393	2394#	2397	2398#	2401	2402#	2405	2406#	2410	2411#	2414
2415#	2419#	2450	2451#	2458	2459#	2465	2466#	2469	2470#	2483	2484#	2489
2490#	2497	2498#	2505	2506#	2511#	2538	2539#	2546	2547#	2553	2554#	2557
2558#	2563	2564#	2567	2568#	2572	2573#	2576	2577#	2582#	2611	2612#	2619
2620#	2626	2627#	2633	2634#	2639	2640#	2647	2648#	2654	2655#	2658	2659#
2664#	2693	2694#	2701	2702#	2708	2709#	2712	2713#	2716	2717#	2720	2721#
2725	2726#	2729	2730#	2735#	2766	2767#	2774	2775#	2781	2782#	2785	2786#
2800	2801#	2804	2805#	2810	2811#	2816	2817#	2821#	2851	2852#	2859	2860#
2866	2867#	2872	2873#	2876	2877#	2882	2883#	2887	2888#	2891	2892#	2896#
2918	2919#	2931#	2974	2975#	2982	2983#	2989	2990#	2993	2994#	2999	3000#
3003	3004#	3008	3009#	3012	3013#	3020#	3045	3046#	3053	3054#	3060	3061#
3064	3065#	3076	3077#	3080	3081#	3085#	3111	3112#	3119	3120#	3126	3127#
3130	3131#	3142	3143#	3146	3147#	3152#	3177	3178#	3185	3186#	3192	3193#
3196	3197#	3208	3209#	3212	3213#	3218#	3242	3243#	3250	3251#	3257	3258#
3261	3262#	3267#	3307	3308#	3315	3316#	3322	3323#	3326	3327#	3338	3339#
3342	3343#	3351#	3377	3378#	3385	3386#	3392	3393#	3396	3397#	3408	3409#
3412	3413#	3418#	3448	3449#	3456	3457#	3463	3464#	3467	3468#	3479	3480#
3483	3484#	3488#	3512	3513#	3520	3521#	3527	3528#	3531	3532#	3537#	3586
3587#	3594	3595#	3601	3602#	3605	3606#	3617	3618#	3621	3622#	3630#	3656
3657#	3664	3665#	3671	3672#	3675	3676#	3689	3690#	3693	3694#	3699#	3727
3728#	3735	3736#	3742	3743#	3746	3747#	3760	3761#	3764	3765#	3770#	3815
3816#	3823	3824#	3830	3831#	3834	3835#	3848	3849#	3852	3853#	3860#	3889
3890#	3897	3898#	3904	3905#	3908	3909#	3922	3923#	3926	3927#	3932#	3959
3960#	3967	3968#	3974	3975#	3978	3979#	3992	3993#	3996	3997#	4002#	4052
4053#	4060	4061#	4067	4068#	4071	4072#	4085	4086#	4089	4090#		
NAME	016276	750	4664#									
NEXTST	014454	4248	4254	4256#								
NXM =	177777	503#	785	879	2292	2294	2362	2596	2678	3033	3034	3099
		3165	3228	3229	3230	3283	3284	3365	3366	3430	3431	3498
		3554	3555	3642	3643	3711	3712	3787	3788	3874	3875	3944
		4026										
OLDPC	000702	691#	4153*	4160*	4167*	4174*						
ONEBYT	000714	697#										
ONES	000712	696#	876									
PATGEN	001300	749	762#									
PCI	000670	686#	1638*	2178*	2951*	3286*	3564*	3791*	4029*	4275		

\$APTHD	000400	556	562#						
\$ASTAT=	***** U	4606	4621						
\$ATYC	016014	4577	4579#						
\$ATY1	015770	4575#							
\$ATY3	015776	4470	4576#						
\$ATY4	016006	4578#							
\$AUTOB	000534	595#							
\$BASE	000642	657#	752						
\$BDADR	000522	590#							
\$BDDAT	000526	592#							
\$CHARC	015664	4487*	4497*	4504	4530*	4535#			
\$CKSWR=	***** U	4663							
\$CLR.T	014040	4118	4121#						
\$CMTAG	000500	578#	705						
\$CM3 =	000000	608#							
\$CPUOP	000614	631#							
\$CRLF	000563	610#	4414	4418	4486	4538			
\$DEVCT	000576	622#							
\$DOAGN	014060	4110	4116	4123	4129#				
\$ENDAD	014050	543	748	4125#	4437				
\$ENDCT	014012	719	4112#						
\$ENULL	014124	4145#							
\$ENV	000606	627#	4017	4251	4420	4465	4584	4608	
\$ENVM	000607	628#	744	4467	4472	4586			
\$EOP	013762	4014	4020	4103#					
\$EOPCT	014004	719*	4109#	4113					
\$ERFLG	000503	581#	4290	4319	4325*	4335	4405*		
\$ERMAX	000515	587#	721*	4331*	4335				
\$ERROR	015140	713	4404#						
\$ERRPC	000516	588#	4409*	4410*	4411	4415			
\$ERRTB	000644	674#							
\$ERTTL	000512	585#	4408*						
\$ESCAP	000560	608#	720*	4330*	4433	4435			
\$ETABL	000606	626#							
\$ETEND	000644	568	658#						
\$FATAL	000570	619#	700*	4424*	4612*				
\$FFLG	016234	4575*	4578*	4606	4615*	4623#			
\$FILLC	000556	606#	4490	4538					
\$FILLS	000555	605#	4538						
\$GDADR	000520	589#							
\$GDDAT	000524	591#							
\$GET42	014022	4115#							
\$GTSWR=	***** U	4662							
\$HIBTS	000400	563#							
\$ICNT	000504	582#							
\$ILLUP	015122	4340	4356	4378#					
\$INTAG	000535	596#							
\$ITEMB	000514	586#	4411*	4424					
\$LF	000564	611#	4538						
\$LFLG	016233	4616*	4622#						
\$LOOP	014116	4137	4141#	4374					
\$LPADR	000506	583#	737*	4323*	4328*	4333	4335	4335	
\$LPERR	000510	584#	738*	4323	4329*	4335	4432		
\$MADR1	000620	644#							
\$MADR2	000624	648#							
\$MADR3	000630	651#							

CATCH	357#	4153	4160	4167	4174														
COMMEN	1#	494#																	
CPREP	360#	2290	2360	2430	2522	2592	2674	2746	2830	2945									
EHLT	365#	800	808	815	821	825	829	834	838	845	852	860	896	904	911				
	916	920	924	929	933	940	946	951	987	995	1002	1006	1013	1018	1022				
	1026	1033	1041	1049	1084	1092	1099	1104	1108	1112	1117	1121	1128	1136	1147				
	1155	1194	1202	1209	1214	1218	1222	1227	1231	1237	1245	1256	1264	1300	1308				
	1315	1319	1326	1331	1335	1339	1347	1356	1366	1406	1414	1421	1426	1430	1434				
	1439	1443	1450	1458	1468	1474	1511	1519	1526	1530	1537	1542	1546	1550	1557				
	1565	1573	1602	1659	1668	1675	1680	1684	1688	1693	1697	1704	1712	1722	1728				
	1768	1776	1783	1787	1794	1799	1803	1807	1814	1824	1832	1868	1876	1883	1888				
	1892	1896	1901	1905	1911	1920	1930	1938	1975	1983	1990	1994	2001	2006	2010				
	2014	2021	2031	2039	2077	2085	2092	2097	2101	2105	2110	2114	2120	2129	2139				
	2147	2200	2209	2216	2221	2225	2229	2234	2238	2244	2253	2263	2271	2306	2314				
	2321	2325	2329	2333	2338	2342	2378	2386	2393	2397	2401	2405	2410	2414	2450				
	2458	2465	2469	2483	2489	2497	2505	2538	2546	2553	2557	2563	2567	2572	2576				
	2611	2619	2626	2633	2639	2647	2654	2658	2693	2701	2708	2712	2716	2720	2725				
	2729	2766	2774	2781	2785	2800	2804	2810	2816	2851	2859	2866	2872	2876	2882				
	2887	2891	2918	2974	2982	2989	2993	2999	3003	3008	3012	3045	3053	3060	3064				
	3076	3080	3111	3119	3126	3130	3142	3146	3177	3185	3192	3196	3208	3212	3242				
	3250	3257	3261	3307	3315	3322	3326	3338	3342	3377	3385	3392	3396	3408	3412				
	3448	3456	3463	3467	3479	3483	3512	3520	3527	3531	3586	3594	3601	3605	3617				
	3621	3656	3664	3671	3675	3689	3693	3727	3735	3742	3746	3760	3764	3815	3823				
	3830	3834	3848	3852	3889	3897	3904	3908	3922	3926	3959	3967	3974	3978	3992				
	3996	4052	4060	4067	4071	4085	4089	4089											
ENDCOM	1#	494#																	
ENDPAS	353#	4114																	
ERR	366#	800	808	815	821	825	829	834	838	845	852	860	896	904	911				
	916	920	924	929	933	940	946	951	987	995	1002	1006	1013	1018	1022				
	1026	1033	1041	1049	1084	1092	1099	1104	1108	1112	1117	1121	1128	1136	1147				
	1155	1194	1202	1209	1214	1218	1222	1227	1231	1237	1245	1256	1264	1300	1308				
	1315	1319	1326	1331	1335	1339	1347	1356	1366	1406	1414	1421	1426	1430	1434				
	1439	1443	1450	1458	1468	1474	1511	1519	1526	1530	1537	1542	1546	1550	1557				
	1565	1573	1602	1659	1668	1675	1680	1684	1688	1693	1697	1704	1712	1722	1728				
	1768	1776	1783	1787	1794	1799	1803	1807	1814	1824	1832	1868	1876	1883	1888				
	1892	1896	1901	1905	1911	1920	1930	1938	1975	1983	1990	1994	2001	2006	2010				
	2014	2021	2031	2039	2077	2085	2092	2097	2101	2105	2110	2114	2120	2129	2139				
	2147	2200	2209	2216	2221	2225	2229	2234	2238	2244	2253	2263	2271	2306	2314				
	2321	2325	2329	2333	2338	2342	2378	2386	2393	2397	2401	2405	2410	2414	2450				
	2458	2465	2469	2483	2489	2497	2505	2538	2546	2553	2557	2563	2567	2572	2576				
	2611	2619	2626	2633	2639	2647	2654	2658	2693	2701	2708	2712	2716	2720	2725				
	2729	2766	2774	2781	2785	2800	2804	2810	2816	2851	2859	2866	2872	2876	2882				
	2887	2891	2918	2974	2982	2989	2993	2999	3003	3008	3012	3045	3053	3060	3064				
	3076	3080	3111	3119	3126	3130	3142	3146	3177	3185	3192	3196	3208	3212	3242				
	3250	3257	3261	3307	3315	3322	3326	3338	3342	3377	3385	3392	3396	3408	3412				
	3448	3456	3463	3467	3479	3483	3512	3520	3527	3531	3586	3594	3601	3605	3617				
	3621	3656	3664	3671	3675	3689	3693	3727	3735	3742	3746	3760	3764	3815	3823				
	3830	3834	3848	3852	3889	3897	3904	3908	3922	3926	3959	3967	3974	3978	3992				
	3996	4052	4060	4067	4071	4085	4089	4089											
ERROR	388#	800	808	815	821	825	829	834	838	845	852	860	896	904	911				
	916	920	924	929	933	940	946	951	987	995	1002	1006	1013	1018	1022				
	1026	1033	1041	1049	1084	1092	1099	1104	1108	1112	1117	1121	1128	1136	1147				
	1155	1194	1202	1209	1214	1218	1222	1227	1231	1237	1245	1256	1264	1300	1308				
	1315	1319	1326	1331	1335	1339	1347	1356	1366	1406	1414	1421	1426	1430	1434				
	1439	1443	1450	1458	1468	1474	1511	1519	1526	1530	1537	1542	1546	1550	1557				
	1565	1573	1602	1659	1668	1675	1680	1684	1688	1693	1697	1704	1712	1722	1728				

.MAIN. MACY11 30(1046) 22-JAN-82 08:44 PAGE 104
CVKAIB.P11 22-JAN-82 08:43 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0101

.1170 1#

. ABS. 017310 000

ERRORS DETECTED: 0

CVKAIB.BIN, CVKAIB.LST/CRF/SOL/NL:TOC=SYSMAC.SML, CVKAIB.P11
RUN-TIME: 18 20 1 SECONDS
RUN-TIME RATIO: 104/40=2.5
CORE USED: 41K (82 PAGES)