

LSI-11

MOVE/STRING INSTR TEST
MD-11-DVKAI-A

EP DVKAI A DL A
COPYRIGHT 1977
FICHE 1 OF 1

MAR 1977
digital
MADE IN USA

B01

EDF:100KABPSE0411 00010000
DVKAI:11 P11 22-DEC-76 10:47

MAC022427(1006) 22-DEC-76 10:49 PAGE 00010000

00010000

770224
SEQ 0002

.REM !

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVKAI-A-D
PRODUCT NAME: DIS MOVE & STRING INSTRUCTION TEST
DATE CREATED: FEBRUARY 1977
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: D.J. CASALETTO

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION

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

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

MD-11-DVKAI*	[THIS DIAGNOSTIC]
MD-11-DVKAJ*	DIS DECIMAL INSTRUCTION TEST
MD-11-DVKAB*	LSI-11 EIS INSTRUCTION TEST

WHERE "*" IS THE LATEST REVISION

1.2 SYSTEM REQUIREMENTS

1.2.1 EQUIPMENT

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

1.2.2 STORAGE

THE PROGRAM USES MEMORY FROM 000000 TO 17434

1.2.3 PRELIMINARY PROGRAMS

IT IS ASSUMED THAT THE FOLLOWING DIAGNOSTICS HAVE BEEN RUN:

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

WHERE "*" IS THE LATEST REVISION

OPERATING INSTRUCTIONS

LOADING PROCEDURES

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES

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.

OPERATING PROCEDURES

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 TIMEOUT
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.

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*
```

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.

ERROR REPORTING

IF A ROUTINE FAILS AND THE INHIBIT ERROR TIMEOUT (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
           3           <--ODT ENTERED

```

WHERE "XXXXXX" IS AN OCTAL VALUE

SUBROUTINE ABSTRACTS

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).

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.

ERROR

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

\$POWER

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

NPREP

4.6 GENR

4.7 XPSW

4.9 INTR

THIS ROUTINE IS USED TO DETECT WHEN THE TEST INSTRUCTION HAS BEEN INTERRUPTED AND TO CONTINUE THE INTERRUPT STREAM UNTIL THE INSTRUCTION IS INTERRUPTED.

5.0 MISCELLANEOUS
-----5.1 STACK POINTER

STACK POINTER IS INITIALLY SET TO 600.

5.2 PASS COUNT
-----A 16 BIT LOCATION "\$PASS" IS USED TO KEEP THE PASS COUNT. IT IS
CLEARED BY STARTING AT 200.5.3 TEST NUMBER
-----A 16 BIT LOCATION "\$TSTNM" IS USED TO KEEP TRACK OF THE SUBTEST
NUMBER. THIS NUMBER IS ALSO PLACED IN THE APT E-TABLE AT "\$TESTN"
WHEN UNDER APT.5.4 POWER FAIL
-----THE DIAGNOSTIC CAN BE POWER FAILED WITH NO ERRORS. AFTER POWERING
DOWN AND THEN UP AGAIN, THE PROGRAM WILL RESTART FROM TEST# 1
(I.E., RESTARTS THE PASS THAT WAS INTERRUPTED) AFTER TYPING "POWER"
TO THE CONSOLE. HOWEVER IF THE PROGRAM IS STORED IN MOS MEMORY
THAT CAN NOT HOLD DATA WITH POWER DOWN, THEN THE PROGRAM WILL NOT
RECOVER FROM A POWER FAIL.5.5 EVENT INTERRUPTS
-----THIS DIAGNOSTIC CAN BE RUN WITH THE REAL TIME CLOCK ACTIVE
(INTERRUPT = 100). LOCATION 100 POINTS TO LOCATION 102
WHICH CONTAINS AN "RTI" INSTRUCTION. THUS ON CLOCK INTERRUPTS,
AN RTI IS EXECUTED TO HANDLE IT.

J01

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 9
DYKIA.P11 22-DEC-76 10:47

SEQ 0010

```
.ENABLE ABS
```

```

.LIST      ME
.NLIST     MC,MD,CND

```


.SBTTL BASIC DEFINITIONS

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

STACK= 1100

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

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

;*MISCELLANEOUS DEFINITIONS

HT= 11 ;;CODE FOR HORIZONTAL TAB

LF= 12 ;;CODE FOR LINE FEED

CR= 15 ;;CODE FOR CARRIAGE RETURN

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

PS= 177776 ;;PROCESSOR STATUS WORD

.EQUIV PS,PSW

STKLMT= 177774 ;;STACK LIMIT REGISTER

PIRQ= 177772 ;;PROGRAM INTERRUPT REQUEST REGISTER

DSWR= 177570 ;;HARDWARE SWITCH REGISTER

DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS

R0= %0 ;;GENERAL REGISTER

R1= %1 ;;GENERAL REGISTER

R2= %2 ;;GENERAL REGISTER

R3= %3 ;;GENERAL REGISTER

R4= %4 ;;GENERAL REGISTER

R5= %5 ;;GENERAL REGISTER

R6= %6 ;;GENERAL REGISTER

R7= %7 ;;GENERAL REGISTER

SP= %6 ;;STACK POINTER

PC= %7 ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS

PR0= 0 ;;PRIORITY LEVEL 0

PR1= 40 ;;PRIORITY LEVEL 1

PR2= 100 ;;PRIORITY LEVEL 2

PR3= 140 ;;PRIORITY LEVEL 3

PR4= 200 ;;PRIORITY LEVEL 4

PR5= 240 ;;PRIORITY LEVEL 5

PR6= 300 ;;PRIORITY LEVEL 6

PR7= 340 ;;PRIORITY LEVEL 7

;*SWITCH REGISTER" SWITCH DEFINITIONS

SW15= 100000

SW14= 40000

SW13= 20000

SW12= 10000

SW11= 4000

SW10= 2000

SW09= 1000

SW08= 400

SW07= 200

SW06= 100

SW05= 40

SW04= 20

001100

000011

000012

000015

000200

177776

177774

177772

177570

177570

000000

000001

000002

000003

000004

000005

000006

000007

000006

000007

000000

000040

000100

000140

000200

000240

000300

000340

100000

040000

020000

010000

004000

002000

001000

000400

000200

000100

000040

000020

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 11
 DVKAIA.F11 22-DEC-76 10:47 BASIC DEFINITIONS

SEQ 0012

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

000010
000004
000002
000001

SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09,BIT9
.EQUIV BIT08,BIT8
.EQUIV BIT07,BIT7
.EQUIV BIT06,BIT6
.EQUIV BIT05,BIT5
.EQUIV BIT04,BIT4
.EQUIV BIT03,BIT3
.EQUIV BIT02,BIT2
.EQUIV BIT01,BIT1
.EQUIV BIT00,BIT0

.*BASIC "CPU" TRAP VECTOR ADDRESSES

000004
000010
000014
000014
000014
000020
000024
000030
000034
000060
000064
000240

ERRVEC= 4 ; TIME OUT AND OTHER ERRORS
 RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS
 TBITVEC= 14 ; "T" BIT
 TRTVEC= 14 ; TRACE TRAP
 BPTVEC= 14 ; BREAKPOINT TRAP (BPT)
 IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
 PWRVEC= 24 ; POWER FAIL
 EMTVEC= 30 ; EMULATOR TRAP (EMT) **ERROR**
 TRAPVEC= 34 ; "TRAP" TRAP
 TKVEC= 60 ; TTY KEYBOARD VECTOR
 TPVEC= 64 ; TTY PRINTER VECTOR
 PIRQVEC= 240 ; PROGRAM INTERRUPT REQUEST VECTOR

MO1

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 12
DVKAIA.P11 22-DEC-76 10:47 BASIC DEFINITIONS

SEQ 0013

467	000200	APTSIZE=	200
468	000001	APTENV=	001
469	000100	APTSP00L=	100
470	000040	APTC SUP=	040
471	171400	\$SWR=	171400
472	000300	\$SWRMK=	300
473	000020	TBIT=	20
474	000001	\$TN=	1
475	000001	N=	1
476	177777	NXM=	177777
477	177560	ABASE=	177560
478	000060	AVECT1=	60
479	000000	. =	0

NO1

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 13
 DVKAIA.P11 22-DEC-76 10:47 BASIC DEFINITIONS

SEQ 0014

```

480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507

      .SBTTL TRAP CATCHER

      =0
      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

      =174
      DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
      SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER

      =0
      TZERO
      340
      TIMTRP
      340
      ILLTRP
      340

      ;SET LOCATIONS 0.4,6 TO ERROR REPORT

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

      =200
      JMP START
      ;STARTING ADDRESS OF PROGRAM
  
```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 14
 DVKATA.P11 22-DEC-76 10:47 TRAP CATCHER

SEQ 0015

```

508      000400
509      .SBTTL  =400
510      ACT11 HOOKS
511
512      ;*****
513      ;HOOKS REQUIRED BY ACT11
514      $SVPC=.      ;SAVE PC
515      =46
516      $ENDAD      ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOF
517      =52
518      .WORD 0      ;;2)SET LOC.52 TO ZERO
519      = $SVPC      ;; RESTORE PC
520      .SBTTL  APT PARAMETER BLOCK
521
522      ;*****
523      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
524      ;*****
525      $X=.      ;;SAVE CURRENT LOCATION
526      =24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
527      200      ;;FOR APT START UP
528      =44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
529      $APTHDR  ;;POINT TO APT HEADER BLOCK
530      =. $X      ;;RESET LOCATION COUNTER
531      ;*****
532      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
533      ;INTERFACE SPEC.
534
535      $APTHD:
536      $HI8TS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
537      $MBADR: .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
538      $TSTM:  .WORD 30      ;;RUN TIM OF LONGEST TEST
539      $PASTM: .WORD 40      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
540      $UNITM: .WORD      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
541      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
542

```

C02

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 15
DVKAIA.P11 22-DEC-76 10:47 APT PARAMETER BLOCK

SEQ 0016

543


```

544 .SBTTL COMMON TAGS
545
546 ;*****
547 ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
548 ;*USED IN THE PROGRAM.
549
550 000500      .=500
551 000500      $CMTAG:      .;START OF COMMON TAGS
552 000500      .WORD      0
553 000502      $TSTNM:     .BYTE      0      .;CONTAINS THE TEST NUMBER
554 000503      $ERFLG:     .BYTE      0      .;CONTAINS ERROR FLAG
555 000504      $ICNT:      .WORD      0      .;CONTAINS SUBTEST ITERATION COUNT
556 000506      $LPADR:     .WORD      0      .;CONTAINS SCOPE LOOP ADDRESS
557 000510      $LPERR:     .WORD      0      .;CONTAINS SCOPE RETURN FOR ERRORS
558 000512      $ERTTL:     .WORD      0      .;CONTAINS TOTAL ERRORS DETECTED
559 000514      $ITEMB:     .BYTE      0      .;CONTAINS ITEM CONTROL BYTE
560 000515      $ERMAX:     .BYTE      1      .;CONTAINS MAX. ERRORS PER TEST
561 000516      $ERRPC:     .WORD      0      .;CONTAINS PC OF LAST ERROR INSTRUCTION
562 000520      $GDADR:     .WORD      0      .;CONTAINS ADDRESS OF 'GOOD' DATA
563 000522      $BDADR:     .WORD      0      .;CONTAINS ADDRESS OF 'BAD' DATA
564 000524      $GDDAT:     .WORD      0      .;CONTAINS 'GOOD' DATA
565 000526      $BDDAT:     .WORD      0      .;CONTAINS 'BAD' DATA
566 000530      .WORD      0      .;RESERVED--NOT TO BE USED
567 000532      .WORD      0
568 000534      $AUTOB:     .BYTE      0      .;AUTOMATIC MODE INDICATOR
569 000535      $INTAG:     .BYTE      0      .;INTERRUPT MODE INDICATOR
570 000536      .WORD      0
571 000540      $SWR:       .WORD      DSWR      .;ADDRESS OF SWITCH REGISTER
572 000542      $DISPLAY:  .WORD      DDISP     .;ADDRESS OF DISPLAY REGISTER
573 000544      $TKS:       177560      .;TTY KBD STATUS
574 000546      $TKB:       177562      .;TTY KBD BUFFER
575 000550      $TPS:       177564      .;TTY PRINTER STATUS REG. ADDRESS
576 000552      $TPB:       177566      .;TTY PRINTER BUFFER REG. ADDRESS
577 000554      $NULL:     .BYTE      0      .;CONTAINS NULL CHARACTER FOR FILLS
578 000555      $FILLS:     .BYTE      2      .;CONTAINS # OF FILLER CHARACTERS REQUIRED
579 000556      $FILLC:     .BYTE     12      .;INSERT FILL CHARS. AFTER A "LINE FEED"
580 000557      $TPFLG:     .BYTE      0      .;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
581 000560      $ESCAPE:    0      .;ESCAPE ON ERROR ADDRESS
582 000562      $QUES:      .ASCII  /?/      .;QUESTION MARK
583 000563      $CRLF:      .ASCII  <15>     .;CARRIAGE RETURN
584 000564      $LF:        .ASCIIZ  <12>     .;LINE FEED
585
586 ;*****
587 .SBTTL APT MAILBOX-ETABLE
588
589 ;*****
590 .EVEN
591 000566      $MAIL:      .;APT MAILBOX
592 000566      $MSGTY:     .WORD      AMSGTY   .;MESSAGE TYPE CODE
593 000570      $FATAL:     .WORD      AFATAL   .;FATAL ERROR NUMBER
594 000572      $TESTN:     .WORD      ATESTN   .;TEST NUMBER
595 000574      $PASS:      .WORD      APASS     .;PASS COUNT
596 000576      $DEVCT:     .WORD      ADEVCT   .;DEVICE COUNT
597 000600      $UNIT:      .WORD      AUNIT    .;I/O UNIT NUMBER
598 000602      $MSGAD:     .WORD      AMSGAD   .;MESSAGE ADDRESS
599 000604      $MSGLG:     .WORD      AMSGLG   .;MESSAGE LENGTH
600 000606      $ETABLE:    .;APT ENVIRONMENT TABLE

```

E02

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 17
 DVKAIA.P11 22-DEC-76 10:47 APT MAILBOX-ETABLE

SEQ 0012

600	000606	000	\$ENV:	.BYTE	AENV	::ENVIRONMENT BYTE
601	000607	000	\$ENVM:	.BYTE	AENVM	::ENVIRONMENT MODE BITS
602	000610	000000	\$SWREG:	.WORD	ASWREG	::APT SWITCH REGISTER
603	000612	000000	\$USWR:	.WORD	AUSWR	::USER SWITCHES
604	000614	000000	\$CPUOP:	.WORD	ACPUOP	::CPU TYPE, OPTIONS
605			.*			BITS 15-11=CPU TYPE
606			.*			11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
607			.*			11/70=06, PDQ=07, Q=10
608			.*			BIT 10=REAL TIME CLOCK
609			.*			BIT 9=FLOATING POINT PROCESSOR
610			.*			BIT 8=MEMORY MANAGEMENT
611	000616	000	\$MAMS1:	.BYTE	AMAMS1	::HIGH ADDRESS, M.S. BYTE
612	000617	000	\$MTYP1:	.BYTE	AMTYP1	::MEM. TYPE, BLK#1
613			.*			MEM. TYPE BYTE -- (HIGH BYTE)
614			.*			900 NSEC CORE=001
615			.*			300 NSEC BIPOLAR=002
616			.*			500 NSEC MOS=003
617	000620	000000	\$MADR1:	.WORD	AMADR1	::HIGH ADDRESS, BLK#1
618			.*			MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
619	000622	000	\$MAMS2:	.BYTE	AMAMS2	::HIGH ADDRESS, M.S. BYTE
620	000623	000	\$MTYP2:	.BYTE	AMTYP2	::MEM. TYPE, BLK#2
621	000624	000000	\$MADR2:	.WORD	AMADR2	::MEM. LAST ADDRESS, BLK#2
622	000626	000	\$MAMS3:	.BYTE	AMAMS3	::HIGH ADDRESS, M.S. BYTE
623	000627	000	\$MTYP3:	.BYTE	AMTYP3	::MEM. TYPE, BLK#3
624	000630	000000	\$MADR3:	.WORD	AMADR3	::MEM. LAST ADDRESS, BLK#3
625	000632	000	\$MAMS4:	.BYTE	AMAMS4	::HIGH ADDRESS, M.S. BYTE
626	000633	000	\$MTYP4:	.BYTE	AMTYP4	::MEM. TYPE, BLK#4
627	000634	000000	\$MADR4:	.WORD	AMADR4	::MEM. LAST ADDRESS, BLK#4
628	000636	000060	\$VECT1:	.WORD	AVECT1	::INTERRUPT VECTOR#1, BUS PRIORITY#1
629	000640	000000	\$VECT2:	.WORD	AVECT2	::INTERRUPT VECTOR#2, BUS PRIORITY#2
630	000642	177560	\$BASE:	.WORD	ABASE	::BASE ADDRESS OF EQUIPMENT UNDER TEST
631	000644		\$ETEND:			
632			.MEXIT			

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 18
 DVKAIA.P11 22-DEC-76 10:47 ERROR POINTER TABLE

SEQ 0019

633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671

.SBTTL ERROR POINTER TABLE

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

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

\$ERRTB:

SRCLN: .WORD 0
 SRCAD: .WORD
 DSTLN: .WORD
 DSTAD: .WORD
 FILL: .WORD
 TABLE: .WORD 177777
 TCSR: 177564
 TBUF: 177566
 TVECT: 64
 TPSW: 66
 PCI: 0
 CCODES: 0
 EXPPSW: 0
 SAVR6: 0
 BADR6: 0
 OLDPC: 0
 TEMP: 0
 TEMP1: 0
 TEMP2: 0

;;SOURCE LENGTH
 ;;SOURCE ADDRESS
 ;;DESTINATION LENGTH
 ;;DESTINATION ADDRESS
 ;;FILL CHARACTER
 ;;TRANSLATION TABLE ADDRESS
 ;;TCSR ADDRESS OF SLU USED FOR INTERRUPTS
 ;;TBUF ADDRESS
 ;;TRANSMIT INTERRUPT VECTOR
 ;;AND PSW LOCATION
 ;;ADDRESS OF TEST INSTRUCTION TO INTERRUPT
 ;;CONDITION CODES AFTER TEST INSTRUCTION EXECUTION
 ;;EXPECTED CONDITION CODES
 ;;STACK POINTER VALUE BEFORE TEST INSTRUCTION EXECUTION
 ;;BAD STACK POINTER VALUE
 ;;PC WHERE UNEXPECTED TRAP OR INTERRUPT OCCURRED

ONES: .WORD -1
 ONEBYT: 377

000644

000644 000000
 000646 000000
 000650 000000
 000652 000000
 000654 000000
 000656 177777
 000660 177564
 000662 177566
 000664 000064
 000666 000066
 000670 000000
 000672 000000
 000674 000000
 000676 000000
 000700 000000
 000702 000000
 000704 000000
 000706 000000
 000710 000000
 000712 177777
 000714 000377

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 19
 DVKAIA.P11 22-DEC-76 10:47 ERROR POINTER TABLE

SEQ 0020

672											
673	000716	005067	177646			START:	CLR	\$FATAL		;CLEAR ERROR NO.	
674	000722	005067	177640				CLR	\$MSGTYP		;CLEAR MESSAGE TYPE	
675	000726	005067	177640				CLR	\$TESTN		;CLEAR TEST NO.	
676						.SBTTL	INITIALIZE THE COMMON TAGS				
677										;CLEAR THE COMMON TAGS (\$CMTAG) AREA	
678	000732	012706	000500				MOV	#\$CMTAG,R6		;FIRST LOCATION TO BE CLEARED	
679	000736	005026					CLR	(R6)+		;CLEAR MEMORY LOCATION	
680	000740	022706	000540				CMP	#\$SWR,R6 ;DONE?			
681	000744	001374					BNE	-6		;LOOP BACK IF NO	
682	000746	012706	000500				MOV	#\$500,SP		;SETUP THE STACK POINTER	
683										;INITIALIZE A FEW VECTORS	
684	000752	012737	015412	000020			MOV	#\$SCOPE,0#\$IOTVEC		;IOT VECTOR FOR SCOPE ROUTINE	
685	000760	012737	000340	000022			MOV	#\$340,0#\$IOTVEC+2		;LEVEL 7	
686	000766	012737	016010	000030			MOV	#\$ERROR,0#\$EMTVEC		;EMT VECTOR FOR ERROR ROUTINE	
687	000774	012737	000340	000032			MOV	#\$340,0#\$EMTVEC+2		;LEVEL 7	
688	001002	012737	016354	000034			MOV	#\$TRAP,0#\$TRAPVEC		;TRAP VECTOR FOR TRAP CALLS	
689	001010	012737	000340	000036			MOV	#\$340,0#\$TRAPVEC+2		;LEVEL 7	
690	001016	012737	015610	000024			MOV	#\$PWARN,0#\$PWAVEC		;POWER FAILURE VECTOR	
691	001024	012737	000340	000026			MOV	#\$340,0#\$PWAVEC+2		;LEVEL 7	
692	001032	016767	013714	013704			MOV	\$ENDCT,\$EOPCT		;SETUP END-OF-PROGRAM COUNTER	
693	001040	005067	177514				CLR	\$ESCAPE		;CLEAR THE ESCAPE ON ERROR ADDRESS	
694	001044	112767	000001	177443			MOVB	#\$1,\$ERMAX		;ALLOW ONE ERROR PER TEST	
695										;INITIALIZE THE "T-BIT" TRAP VECTOR. THEN LOAD LOCATION	
696										;THE "END-OF-PASS" (\$EOP) ROUTINE, WITH A "RTI" OR "RTT"	
697	001052	012737	015054	000014			MOV	#\$SRTN,0#\$TBITVEC		;SET "T" BIT VECTOR TO SRTN	
698	001060	012737	000340	000016			MOV	#\$340,0#\$TBITVEC+2		;LEVEL 7	
699	001066	012767	000002	013760			MOV	#\$RTI,\$SRTN		;SET SRTN TO A RTI	
700	001074	013767	000010	177602			MOV	0#\$RESVEC,TEMP		;SAVE ILLEGAL INSTRUCTION TRAP VECTOR	
701	001102	012737	001130	000010			MOV	0#\$2,\$RESVEC		;TRY TO DO A RTT	
702	001110	005046					CLR	-(SP)		;DUMMY PS	
703	001112	012746	001120				MOV	#\$1,\$-(SP)		;AND PC	
704	001116	000006					RTT			;TRY THE RTT	
705	001120	012767	000006	013726	1\$:		MOV	#\$RTT,\$SRTN		;RTT IS LEGAL--SET SRTN TO A RTT	
706	001126	000402					BR	3\$			
707	001130	062706	000010		2\$:		ADD	#\$10,SP		;RTT ILLEGAL--CLEAN OFF THE STACK	
708	001134	016737	177544	000010	3\$:		MOV	TEMP,0#\$RESVEC		;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR	
709	001142	005067	013714				CLR	\$TBIT		;CLEAR "T" BIT SWITCH	
710	001146	012767	001146	177332			MOV	#\$,\$LPADR		;INITIALIZE THE LOOP ADDRESS FOR SCOPE	
711	001154	012767	001154	177326			MOV	#\$,\$LPERR		;SETUP THE ERROR LOOP ADDRESS	
712										;SETUP FOR A SOFTWARE SWITCH REGISTER.	
713	001162	012767	000176	177350			MOV	#\$SWREG,\$SWR		;POINT TO SOFTWARE SWR	
714	001170	012767	000174	177344			MOV	#\$DISPREG,\$DISPLAY			
715											
716	001176	005067	177372				CLR	\$PASS		;CLEAR PASS COUNT	
717	001202	132767	000200	177377			BITB	#\$APTSIZE,\$ENVN		;TEST USER SIZE UNDER APT	
718	001210	001403					BEQ	4\$		;YES,USE NON-APT SWITCH	
719	001212	012767	000610	177320			MOV	#\$SWREG,\$SWR		;NO,USE APT SWITCH REGISTER	
720	001220				4\$:						
721	001220	026737	013564	000042			CMP	\$ENDAD,0#\$42		;UNDER ACT11 AUTO-ACCEPT?	
722	001226	001424					BEQ	PATGEN		;BR IF YES, & SKIP PROGRAM ID TYPEOUT	
723	001230	104401	016414				TYPE,	NAME			
724										;SET UP ADDRESSES OF SLU TO USE FOR INTERRUPTABILITY TES	
725	001234	013700	000642				MOV	0#\$BASE,RO		;GET ADDRESS OF THE SLU	
726	001240	062700	000004				ADD	#\$4,RO		;ADJUST TO TCSR ADDRESS	
727	001244	010037	000660				MOV	RO,0#\$TCSR		;STORE TCSR ADDRESS	

H02

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 20
 DVKAIA.P11 22-DEC-76 10:47 INITIALIZE THE COMMON TAGS

SEQ 0021

729	001250	005720		TST	(R0)+	:ADJUST TO TBUF ADDRESS
729	001252	010037	000662	MOV	R0,2#TBUF	:STORE TBUF ADDRESS
730	001256	013700	000636	MOV	2#\$VECT1,R0	:GET SLU INTERRUPT VECTOR
731	001262	062700	000004	ADD	#4,R0	:ADJUST TO TRANSMIT INTERRUPT VECTOR
732	001266	010037	000664	MOV	R0,2#TVECT	:STORE TRANSMIT INTERRUPT VECTOR
733	001272	005720		TST	(R0)+	:ADJUST TO TRANSMIT INTERRUPT PSW
734	001274	010037	000666	MOV	R0,2#TPSW	:STORE TRANSMIT INTERRUPT PSW LOCATION
735	001300	012701	016434	PATGEN: MOV	#BUF1,R1	:GENERATE TEST PATTERN IN BUF1
736	001304	012700	000001	MOV	#1,R0	: THE PATTERN IS A BINARY COUNT ALTERNATED
737	001310	110021		1\$: MOVB	R0,(R1)+	: WITH ITS COMPLEMENT
738	001312	105100		COMB	R0	: 001,376,002,375,003,374,.....,177,200,200,177
739	001314	110021		MOVB	R0,(R1)+	
740	001316	105100		COMB	R0	
741	001320	105200		INCB	R0	:INCREMENT PATTERN
742	001322	020027	000200	CMP	R0,#200	:IS PATTERN FINISHED?
743	001326	003770		BLE	1\$	:BR IF NOT
744						
745	001330	106427	000200	BEGIN: MTPS	#200	:SET PRIORITY TO 7

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 21
 DVKAIA.P11 22-DEC-76 10:47 INITIALIZE THE COMMON TAGS

SEQ 0022

```

746
747
748
749
750
751
752
753
754
755 001334 000004
756 001336 004567 013650
757 001342 000000
758 001344 177777
759 001346 000005
760 001350 017035
761 001352 000377
762 001354 004767 013660
763 001360 004567 013722
764 001364 000211
765 001366 004767 013554
766 001372 000251
767 001374 000266
768
769 001376 076030
770
771 001400 106767 177266
772 001404 042767 177400 177260
773 001412 023767 000674 177252
774 001420 001401
775 001422 104001
776
777
778
779 001424
780 001424 023706 000676
781 001430 001403
782 001432 010637 000700
783 001436 104002
784
785
786
787 001440
788 001440 005700
789 001442 001401
790 001444 104003
791
792
793 001446
794 001446 005701
795 001450 001401
796 001452 104004
797
798 001454 005702
799 001456 001401
800 001460 104005
801

;*****
;TEST 1 TEST "MOVC" INSTRUCTION WITH ZERO SOURCE LENGTH
;SUCCESSIONAL COMPLETION OF THIS TEST RESULTS IN
;"FILL" ("377") BYTES WRITTEN THROUGHOUT THE DESTINATION
;FIELD AND CONDITION CODES-->N,C=1 & Z,V=0
;*****
+ST1: SCOPE
      JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
      0 ;SOURCE LENGTH
      NXM ;SOURCE ADDRESS
      5 ;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
      +SEZ!SEV ;SET CONDITION CODES Z & V
      MOVC ;EXECUTE "MOVE CHARACTER" INSTRUCTION

;CHECK RESULTS
;STORE THE PSW
;CLEAR ALL UNUSED BITS
;CHECK PSW AGAINST EXPECTED VALUE
;BR, IF EQUAL
;*****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

```


J02

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 22
 DVKAIA.P11 22-DEC-76 10:47 T1

TEST "MOVC" INSTRUCTION WITH ZERO SOURCE LENGTH

SEQ 0023

802	001462	005703		67\$:	TST	R3		:TEST R3
803	001464	001401			BEQ	68\$		:BR, IF ZERO
804	001466	104006			ERROR	6		:*****TEST 1 - ERROR 5*****
805								:R3 SHOULD BE ZERO
806	001470			68\$:				
807	001470	026704	177160		CMP	FILL,R4		:CHECK R4 UNCHANGED
808	001474	001401			BEQ	69\$		:BR IF OK
809	001476	104007			ERROR	7		:*****TEST 1 - ERROR 7*****
810								:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
811	001500	026705	177152	69\$:	CMP	TABLE,R5		:CHECK R5 UNCHANGED
812	001504	001401			BEQ	70\$		:BR IF OK
813	001506	104010			ERROR	10		:*****TEST 1 - ERROR 10*****
814								:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
815	001510			70\$:				
816								:VERIFY DESTINATION CONTENTS
817	001510	012700	017034		MOV	#BUF2,R0		:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
818	001514	105720			TSTB	(R0)+		:TEST CONTENTS OF BOUNDARY
819	001516	001401			BEQ	71\$		:BR, IF STILL ZERO
820	001520	104011			ERROR	11		:*****TEST 1 - ERROR 11*****
821								:LOWER BOUNDARY OF DESTINATION CHANGED
822								: SHOULD STILL EQUAL ZERO
823	001522			71\$:				
824	001522	016701	177122		MOV	DSTLN,R1		:STORE TRANSFER BYTE COUNT IN R1
825	001526	122067	177122	T1E12:	CMPB	(R0)+,FILL		:CHECK CHARACTERS IN DESTINATION
826	001532	001401			BEQ	+4		:BR IF OK
827	001534	104012			ERROR	12		:*****TEST 1 - ERROR 12*****
828								:COMPARE ERROR IN DESTINATION
829								:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
830								
831	001536	005301			DEC	R1		:DECREMENT BYTE COUNT
832	001540	001372			BNE	T1E12		:BR, IF NOT FINISHED CHECKING
833	001542	105720			TSTB	(R0)+		:TEST CONTENTS OF DEST. UPPER BOUNDARY
834	001544	001401			BEQ	ENDT1		:BR, IF STILL ZERO
835	001546	104013			ERROR	13		:*****TEST 1 - ERROR 13*****
836								:UPPER BOUNDARY OF DEST. CHANGED
837								: SHOULD STILL EQUAL ZERO
838	001550			ENDT1:				

K02

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 23
 DVKAIA.P11 22-DEC-76 10:47 T1

TEST "MOVC" INSTRUCTION WITH ZERO SOURCE LENGTH

SEQ 0024

```

839
840
841
842
843
844
845
846
847
848
849
850 001550 000004
851 001552 012767 000712 177066
852 001560 004567 013426
853 001564 000001
854 001566 177777
855 001570 000000
856 001572 017035
857 001574 000377
858
859 001576 012700 017034
860 001602 005020
861 001604 005020
862 001606 004567 013474
863 001612 000200
864 001614 004767 013326
865 001620 000277
866
867 001622 076030
868
869 001624 106767 177042
870 001630 042767 177400 177034
871 001636 023767 000674 177026
872 001644 001401
873 001646 104001
874
875
876
877 001650
878 001650 023706 000676
879 001654 001403
880 001656 010637 000700
881 001662 104002
882
883
884
885 001664
886 001664 020067 176754
887 001670 001401
888 001672 104003
889
890 001674
891 001674 005701
892 001676 001401
893 001700 104004
894

```

```

*****
*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
*****
TST2:  SCOPE
      MOV  #ONES, SRCAD      ;SET SOURCE ADDRESS
      JSR  R5, PREP          ;SET UP INSTRUCTION ARGUMENTS
      1      ;SOURCE LENGTH
      NXM      ;SOURCE ADDRESS
      0      ;DESTINATION LENGTH
      BUF2+1    ;DESTINATION ADDRESS
      377      ;FILL CHARACTER
      ;CLEAR DESTINATION
      MOV  #BUF2, R0          ;POINT R0 TO LOWER BYTE BOUNDARY
      CLR  (R0)+              ;CLEAR BOUNDARY & DEST. BYTES
      CLR  (R0)+              ;CLEAR UPPER BYTE BOUNDARY
      JSR  R5, XPSW          ;STORE EXPECTED PSW VALUE
      .WORD 200
      JSR  PC, GENR          ;SET UP GENERAL REGISTERS
      SCC      ;SET ALL CONDITION CODES
      ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      ;CHECK RESULTS
      MFPS  CCODES          ;STORE THE PSW
      BIC  #177400, CCODES  ;CLEAR ALL UNUSED BITS
      CMP  @#EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ  64$              ;BR, IF EQUAL
      ERROR 1
      *****TEST 2 - 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 2 - ERROR 2*****
      ;STACK POINTER NOT RESTORED BY INSTRUCTION
      ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ;ERRONEOUS VALUE IS AT "BADR6"
65$:  CMP  R0, SRCLN         ;CHECK R0 EQUAL TO SOURCE LENGTH
      BEQ  1$              ;BR, IF EQUAL
      ERROR 3
      *****TEST 2 - ERROR 3*****
      ;R0 SHOULD EQUAL SOURCE LENGTH
      ;CHECK OTHER GENERAL REGISTERS
1$:   TST  R1                ;TEST R1
      BEQ  66$              ;BR, IF ZERO
      ERROR 4
      *****TEST 2 - ERROR 4*****
      ;R1 SHOULD BE ZERO

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 24
 DVKAIA.P11 22-DEC-76 10:47 T2

TEST "MOVC" INSTRUCTION WITH ZERO DESTINATION LENGTH

SEQ 0025

895	001702	005702		66\$:	TST R2		;TEST R2
896	001704	001401			BEQ 67\$		;BR IF ZERO
897	001706	104005			ERROR 5		;*****TEST 2 - ERROR 5*****
898							;R2 SHOULD BE ZERO
899	001710	005703		67\$:	TST R3		;TEST R3
900	001712	001401			BEQ 68\$		;BR, IF ZERO
901	001714	104006			ERROR 6		;*****TEST 2 - ERROR 6*****
902							;R3 SHOULD BE ZERO
903	001716			68\$:			
904	001716	026704	176732		CMP FILL,R4		;CHECK R4 UNCHANGED
905	001722	001401			BEQ 69\$		;BR IF OK
906	001724	104007			ERROR 7		;*****TEST 2 - ERROR 7*****
907							;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
908	001726	026705	176724	69\$:	CMP TABLE,R5		;CHECK R5 UNCHANGED
909	001732	001401			BEQ 70\$		;BR IF OK
910	001734	104010			ERROR 10		;*****TEST 2 - ERROR 10*****
911							;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
912	001736			70\$:			
913							;VERIFY DESTINATION UNCHANGED
914	001736	012700	017034		MOV #BUF2,R0		;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
915	001742	105720			TSTB (R0)+		;TEST CONTENTS OF BOUNDARY
916	001744	001401			BEQ 71\$		;BR, IF STILL ZERO
917	001746	104011			ERROR 11		;*****TEST 2 - ERROR 11*****
918							;LOWER BOUNDARY OF DESTINATION CHANGED
919							; SHOULD STILL EQUAL ZERO
920	001750			71\$:			
921	001750	105720			TSTB (R0)+		;TEST CONTENTS OF DESTINATION BYTE
922	001752	001401			BEQ 2\$		;BR, IF STILL ZERO
923	001754	104012			ERROR 12		;*****TEST 2 - ERROR 12*****
924							;DESTINATION WAS CHANGED
925	001756			2\$:			
926	001756	105720			TSTB (R0)+		;TEST CONTENTS OF DEST. UPPER BOUNDARY
927	001760	001401			BEQ ENDT2		;BR, IF STILL ZERO
928	001762	104013			ERROR 13		;*****TEST 2 - ERROR 13*****
929							;UPPER BOUNDARY OF DEST. CHANGED
930							; SHOULD STILL EQUAL ZERO
931	001764			ENDT2:			
932							

933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988

001764 000004
001766 004567 013220
001772 000020
001774 016434
001776 000011
002000 017035
002002 000377
002004 004767 013230
002010 004567 013272
002014 000200
002016 004767 013124
002022 000277
002024 076030
002026 106767 176640
002032 042767 177400 176632
002040 023767 000674 176624
002046 001401
002050 104001
002052
002052 023706 000676
002056 001403
002060 010637 000700
002064 104002
002066
002066 026704 176562
002072 001401
002074 104003
002076 026705 176554
002102 001401
002104 104004
002106
002106 016705 176532

```

*****
*TEST 3      TEST "MOVC" INSTRUCTION WITH SRCAD .LT. DSTAD, SL .GT. DL
*****
*PROPER TERMINATION FOR THIS INSTRUCTION TEST
*IS A TRUNCATED SOURCE STORED IN THE DESTINATION
*(LEAST SIGNIFICANT BYTES NOT MOVED), RD EQUALS THE
*NUMBER OF UNMOVED SOURCE BYTES (SRCLN-DSTLN),
*R1-->R3 EQUAL TO ZERO, AND ALL CONDITION
*CODES CLEAR
*****
TST3:  SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      20      ;SOURCE LENGTH
      BUF1     ;SOURCE ADDRESS
      11      ;DESTINATION LENGTH
      BUF2+1   ;DESTINATION ADDRESS
      377     ;FILL CHARACTER
      JSR      PC,CLDST     ;CLEAR DESTINATION
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD    200
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC      ;SET ALL CONDITION CODES
      MOVC     ;EXECUTE "MOVE CHARACTER" INSTRUCTION

      MFPS     CCODES      ;CHECK RESULTS
      BIC      #177400,CCODES ;STORE THE PSW
      CMP      @#EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
      BEQ      64$         ;CHECK PSW AGAINST EXPECTED VALUE
      ERROR    1           ;BR, IF EQUAL
      *****TEST-2 - 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 3 - 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 3 - ERROR 3*****
      R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
      CHECK R5 UNCHANGED
      BEQ      67$         ;BR IF OK
      ERROR    4           ;*****TEST 3 - ERROR 4*****
      R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

67$:  MOV      SRCLN,R5     ;CHECK RD=SOURCE LENGTH MINUS

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 26
 DVKAIA.P11 22-DEC-76 10:47 T3

TEST "MOVC" INSTRUCTION WITH SRCAD .LT. DSTAD, SL .GT. DL

SEQ 0027

989	002112	166705	176532	SUB	DSTLN,R5	;DESTINATION LENGTH
990	002116	020005		CMP	R0,R5	
991	002120	001401		BEQ	2\$	;BR IF OK
992	002122	104005		ERROR	5	;*****TEST 3 - ERROR 5*****
993						;R0 NOT EQUAL TO (SRCLN-DSTLN)
994	002124			2\$: TST	R1	;CHECK OTHER GENERAL REGISTERS
995	002124	005701		BEQ	68\$	;TEST R1
996	002126	001401		ERROR	6	;BR, IF ZERO
997	002130	104005				;*****TEST 3 - ERROR 6*****
998						;R1 SHOULD BE ZERO
999	002132	005702		68\$: TST	R2	;TEST R2
1000	002134	001401		BEQ	69\$	;BR IF ZERO
1001	002136	104007		ERROR	7	;*****TEST 3 - ERROR 7*****
1002						;R2 SHOULD BE ZERO
1003	002140	05703		69\$: TST	R3	;TEST R3
1004	002142	001401		BEQ	70\$	;BR, IF ZERO
1005	002144	104010		ERROR	10	;*****TEST 3 - ERROR 10*****
1006						;R3 SHOULD BE ZERO
1007	002146			70\$:		
1008						;VERIFY DESTINATION AREA
1009	002146	012700	017034	MOV	#BUF2,R0	;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1010	002152	105720		TSTB	(R0)+	;TEST CONTENTS OF BOUNDARY
1011	002154	001401		BEQ	71\$	;BR, IF STILL ZERO
1012	002156	104011		ERROR	11	;*****TEST 3 - ERROR 11*****
1013						;LOWER BOUNDARY OF DESTINATION CHANGED
1014						; SHOULD STILL EQUAL ZERO
1015	002160			71\$:		
1016	002160	016702	176462	MOV	SRCAD,R2	;POINT R2 TO SOURCE STRING
1017	002164	016701	176460	MOV	DSTLN,R1	;STORE TRANSFER BYTE COUNT IN R1
1018	002170	122022		T3E12: CMPB	(R0)+,(R2)+	;CHECK CHARACTERS IN DESTINATION
1019	002172	001401		BEQ	+4	;BR IF OK
1020	002174	104012		ERROR	12	;*****TEST 3 - ERROR 12*****
1021						;COMPARE ERROR IN DESTINATION
1022						;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1023						
1024	002176	005301		DEC	R1	;DECREMENT BYTE COUNT
1025	002200	001373		BNE	T3E12	;BR, IF NOT FINISHED CHECKING
1026	002202	105720		TSTB	(R0)+	;TEST CONTENTS OF DEST. UPPER BOUNDARY
1027	002204	001401		BEQ	ENDT3	;BR, IF STILL ZERO
1028	002206	104013		ERROR	13	;*****TEST 3 - ERROR 13*****
1029						;UPPER BOUNDARY OF DEST. CHANGED
1030						; SHOULD STILL EQUAL ZERO
1031	002210			ENDT3:		

```

1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045 002210 000004
1046 002212 004567 012774
1047 002216 000374
1048 002220 016434
1049 002222 000376
1050 002224 017035
1051 002226 000377
1052 002230 004767 013004
1053 002234 004567 013046
1054 002240 000211
1055 002242 004767 012700
1056 002246 000251
1057 002250 000266
1058
1059 002252 076030
1060
1061 002254 106767 176412
1062 002260 042767 177400 176404
1063 002266 023767 000674 176376
1064 002274 001401
1065 002276 104001
1066
1067
1068
1069 002300
1070 002300 023706 000676
1071 002304 001403
1072 002306 010637 000700
1073 002312 104002
1074
1075
1076
1077 002314
1078 002314 005700
1079 002316 001401
1080 002320 104003
1081
1082 002322
1083 002322 005701
1084 002324 001401
1085 002326 104004
1086
1087 002330 005702

;*****
;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,
;R0-->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
;STORE THE PSW
;CLEAR ALL UNUSED BITS
;CHECK PSW AGAINST EXPECTED VALUE
;BR, IF EQUAL
;*****TEST 4 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "EXPPSW"
;ACTUAL PSW IS STORED AT "CCODES"

64$: CMP Q#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
      BEQ 65$ ;BR IF OK
      MOV SP,Q#BADR6 ;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
;TEST R1
;BR, IF ZERO
;*****TEST 4 - ERROR 4*****
;R1 SHOULD BE ZERO
;TEST R2

1$: TST R1
      BEQ 66$
      ERROR 4

66$: TST R2

```


1088	002332	001401			BEQ	67\$		;BR IF ZERO
1089	002334	104005			ERROR	5		*****TEST 4 - ERROR 5*****
1090								;R2 SHOULD BE ZERO
1091	002336	005703		67\$:	TST	R3		;TEST R3
1092	002340	001401			BEQ	68\$		;BR, IF ZERO
1093	002342	104006			ERROR	6		*****TEST 4 - ERROR 6*****
1094								;R3 SHOULD BE ZERO
1095	002344			68\$:				
1096	002344	026704	176304		CMP	FILL,R4		;CHECK R4 UNCHANGED
1097	002350	001401			BEQ	69\$		;BR IF OK
1098	002352	104007			ERROR	7		*****TEST 4 - ERROR 7*****
1099								;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1100	002354	026705	176276	69\$:	CMP	TABLE,R5		;CHECK R5 UNCHANGED
1101	002360	001401			BEQ	70\$		;BR IF OK
1102	002362	104010			ERROR	10		*****TEST 4 - ERROR 10*****
1103								;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1104	002364			70\$:				
1105								;VERIFY DESTINATION
1106	002364	012700	017034		MOV	#BUF2,R0		;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1107	002370	105720			TSTB	(R0)+		;TEST CONTENTS OF BOUNDARY
1108	002372	001401			BEQ	71\$		;BR, IF STILL ZERO
1109	002374	104011			ERROR	11		*****TEST 4 - ERROR 11*****
1110								;LOWER BOUNDARY OF DESTINATION CHANGED
1111								; SHOULD STILL EQUAL ZERO
1112	002376			71\$:				
1113	002376	016702	176244	2\$:	MOV	SRCAD,R2		;POINT R2 TO SOURCE ADDRESS
1114	002402	016701	176236		MOV	SRCLN,R1		;STORE TRANSFER BYTE COUNT IN R1
1115	002406	122022		T4E12:	CMPB	(R0)+,(R2)+		;CHECK CHARACTERS IN DESTINATION
1116	002410	001401			BEQ	.+4		;BR IF OK
1117	002412	104012			ERROR	12		*****TEST 4 - ERROR 12*****
1118								;COMPARE ERROR IN DESTINATION
1119								;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1120								
1121	002414	005301			DEC	R1		;DECREMENT BYTE COUNT
1122	002416	001373			BNE	T4E12		;BR, IF NOT FINISHED CHECKING
1123	002420	016705	176224		MOV	DSTLN,R5		;CALCULATE THE NUMBER OF "FILL"
1124	002424	166705	176214		SUB	SRCLN,R5		;CHARACTERS THAT SHOULD APPEAR IN DEST.
1125	002430	010501			MOV	R5,R1	;STORE	TRANSFER BYTE COUNT IN R1
1126	002432	122067	176216	T4E13:	CMPB	(R0)+,FILL		;CHECK CHARACTERS IN DESTINATION
1127	002436	001401			BEQ	.+4		;BR IF OK
1128	002440	104013			ERROR	13		*****TEST 4 - ERROR 13*****
1129								;COMPARE ERROR IN DESTINATION
1130								;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1131								
1132	002442	005301			DEC	R1		;DECREMENT BYTE COUNT
1133	002444	001372			BNE	T4E13		;BR, IF NOT FINISHED CHECKING
1134	002446	105720			TSTB	(R0)+		;TEST CONTENTS OF DEST. UPPER BOUNDARY
1135	002450	001401			BEQ	ENDT4		;BR, IF STILL ZERO
1136	002452	104014			ERROR	14		*****TEST 4 - ERROR 14*****
1137								;UPPER BOUNDARY OF DEST. CHANGED
1138								; SHOULD STILL EQUAL ZERO
1139	002454			ENDT4:				

```

1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154 002454 000004
1155 002456 004567 012530
1156 002462 000010
1157 002464 017035
1158 002466 000015
1159 002470 017035
1160 002472 000377
1161 002474 004767 012540
1162 002500 004567 012562
1163 002504 000010
1164 002506 017035
1165 002510 004567 012572
1166 002514 000211
1167 002516 004767 012424
1168 002522 000245
1169 002524 000272
1170
1171 002526 076030
1172
1173 002530 106767 176136
1174 002534 042767 177400 176130
1175 002542 023767 000674 176122
1176 002550 001401
1177 002552 104001
1178
1179
1180
1181 002554
1182 002554 023706 000676
1183 002560 001403
1184 002562 010637 000700
1185 002566 104002
1186
1187
1188
1189 002570
1190 002570 005700
1191 002572 001401
1192 002574 104003
1193
1194 002576
1195 002576 005701

```

```

*****
*TEST 5      TEST "MOVC" INSTRUCTION WITH SRCAD=DSTAD, SL .LT. DL
*****
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS
*NO CHARACTERS TRANSFERED TO THE DESTINATION,
*RO-->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
      MFPS     CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$         ;BR, IF EQUAL
      ERROR    1           ;*****TEST 5 - ERROR 1*****
                              ;PSW ERROR
                              ;EXPECTED PSW IS STORED AT "EXPPSW"
                              ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      @#SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$         ;BR IF OK
      MOV      SP,@#BADR6  ;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$:
      TST      R0           ;CHECK R0 FOR ZERO
      BEQ      1$         ;BR, IF EQUAL
      ERROR    3           ;*****TEST 5 - ERROR 3*****
                              ;R0 SHOULD BE ZERO
                              ;CHECK OTHER GENERAL REGISTERS
      1$:
      TST      R1          ;TEST R1

```


.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 30
 DVKAIA.P11 22-DEC-76 10:47 TS

TEST "MOVC" INSTRUCTION WITH SRCAD=DSTAD, SL .LT. DL

SEQ 0031

1196	002600	001401		BEQ	66\$		;BR, IF ZERO
1197	002602	104004		ERROR	4		*****TEST 5 - ERROR 4*****
1198							;R1 SHOULD BE ZERO
1199	002604	005702	66\$:	TST	R2		;TEST R2
1200	002606	001401		BEQ	67\$		;BR IF ZERO
1201	002610	104005		ERROR	5		*****TEST 5 - ERROR 5*****
1202							;R2 SHOULD BE ZERO
1203	002612	005703	67\$:	TST	R3		;TEST R3
1204	002614	001401		BEQ	68\$		;BR, IF ZERO
1205	002616	104006		ERROR	6		*****TEST 5 - ERROR 6*****
1206							;R3 SHOULD BE ZERO
1207	002620		68\$:				
1208	002620	026704	176030	CMP	FILL,R4		;CHECK R4 UNCHANGED
1209	002624	001401		BEQ	69\$		;BR IF OK
1210	002626	104007		ERROR	7		*****TEST 5 - ERROR 7*****
1211							;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1212	002630	026705	176022	69\$:	CMP	TABLE,R5	;CHECK R5 UNCHANGED
1213	002634	001401		BEQ	70\$		;BR IF OK
1214	002636	104010		ERROR	10		*****TEST 5 - ERROR 10*****
1215							;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1216	002640		70\$:				
1217	002640	012700	017034	MOV	#BUF2,R0		;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1218	002644	105720		TSTB	(R0)+		;TEST CONTENTS OF BOUNDARY
1219	002646	001401		BEQ	71\$		;BR, IF STILL ZERO
1220	002650	104011		ERROR	11		*****TEST 5 - ERROR 11*****
1221							;LOWER BOUNDARY OF DESTINATION CHANGED
1222							; SHOULD STILL EQUAL ZERO
1223	002652		71\$:				
1224	002652	012702	017035	MOV	#BUF2+1,R2		
1225	002656	016701	175762	MOV	SRCLN,R1		;STORE TRANSFER BYTE COUNT IN R1
1226	002662	122022		TSE12:	CMPB	(R0)+,(R2)+	;CHECK CHARACTERS IN DESTINATION
1227	002664	001401		BEQ	+4		;BR IF OK
1228	002666	104012		ERROR	12		*****TEST 5 - ERROR 12*****
1229							;COMPARE ERROR IN DESTINATION
1230							;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1231							
1232	002670	005301		DEC	R1		;DECREMENT BYTE COUNT
1233	002672	001373		BNE	TSE12		;BR, IF NOT FINISHED CHECKING
1234	002674	016702	175750	MOV	DSTLN,R2		;CALCULATE FILL LENGTH
1235	002700	166702	175740	SUB	SRCLN,R2		
1236	002704	010201		MOV	R2,R1		;STORE TRANSFER BYTE COUNT IN R1
1237	002706	122067	175742	TSE13:	CMPB	(R0)+,FILL	;CHECK CHARACTERS IN DESTINATION
1238	002712	001401		BEQ	+4		;BR IF OK
1239	002714	104013		ERROR	13		*****TEST 5 - ERROR 13*****
1240							;COMPARE ERROR IN DESTINATION
1241							;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1242							
1243	002716	005301		DEC	R1		;DECREMENT BYTE COUNT
1244	002720	001372		BNE	TSE13		;BR, IF NOT FINISHED CHECKING
1245	002722	105720		TSTB	(R0)+		;TEST CONTENTS OF DEST. UPPER BOUNDARY
1246	002724	001401		BEQ	ENDT5		;BR, IF STILL ZERO
1247	002726	104014		ERROR	14		*****TEST 5 - ERROR 14*****
1248							;UPPER BOUNDARY OF DEST. CHANGED
1249							; SHOULD STILL EQUAL ZERO
1250	002730			ENDT5:			

```

1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264 002730 000004
1265 002732 004567 012254
1266 002736 000020
1267 002740 017034
1268 002742 000007
1269 002744 017037
1270 002746 000377
1271 002750 004567 012312
1272 002754 000020
1273 002756 017034
1274 002760 004567 012322
1275 002764 000200
1276 002766 004767 012154
1277
1278
1279 002772 076030
1280
1281 002774 106767 175672
1282 003000 042767 177400 175664
1283 003006 023767 000674 175656
1284 003014 001401
1285 003016 104001
1286
1287
1288
1289 003020
1290 003020 023706 000676
1291 003024 001403
1292 003026 010637 000700
1293 003032 104002
1294
1295
1296
1297 003034
1298 003034 026704 175614
1299 003040 001401
1300 003042 104003
1301
1302 003044 026705 175606
1303 003050 001401
1304 003052 104004
1305
1306 003054

*****
*TEST 6 TEST "MOVC" WITH SRCAD .LT. DSTAD, SRCLN .GT. DSTLN
*****
*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN
*THE DESTINATION AREA IS CONTAINED WITHIN THE SOURCE AREA.
*THEREFORE, SRCAD<DSTAD AND SRCLN>DSTLN
*THE RESULT IS A TRUNCATED (LSB OF SOURCE NOT MOVED) SOURCE
*IN THE DESTINATION, RD EQUALS (SRCLN-DSTLN), AND ALL CONDITION
*CODES CLEAR
*****
TS6: SCOPE
      JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
          20 ;SOURCE LENGTH
          BUF2 ;SOURCE ADDRESS
          7 ;DESTINATION LENGTH
          BUF2+3 ;DESTINATION ADDRESS
          377 ;FILL CHARACTER
      JSR R5,GENSRC ;GENERATE SOURCE STRING
          .WORD 20
          .WORD BUF2
      JSR R5,XPSW ;STORE EXPECTED PSW VALUE
          .WORD 200
      JSR PC,GENR ;SET UP GENERAL REGISTERS
          ;SET ALL CONDITION CODES
          ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      MOVC
          ;CHECK RESULTS
          MFPS CCODES ;STORE THE PSW
          BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
          CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
          BEQ 64$ ;BR, IF EQUAL
          ;*****TEST 6 - 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
          ;*****TEST 6 - 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
          ;*****TEST 6 - ERROR 3*****
          ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
          ;CHECK R5 UNCHANGED
          BEQ 67$ ;BR IF OK
          ;*****TEST 6 - ERROR 4*****
          ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
          ERROR 2
          ERROR 3
          ERROR 4
      67$:

```


1307	003054	016705	175564	MOV	SRCLN,R5	:CHECK R0=SOURCE LENGTH MINUS
1308	003060	166705	175564	SUB	DSTLN,R5	:DESTINATION LENGTH
1309	003064	020005		CMP	R0,R5	
1310	003066	001401		BEG	2\$	
1311	003070	104005		ERROR	5	:*****TEST 6 - ERROR 5*****
1312						:R0 NOT EQUAL TO (SRCLN-DSTLN)
1313	003072		2\$:	TST	R1	:TEST R1
1314	003072	005701		BEG	68\$	:BR, IF ZERO
1315	003074	001401		ERROR	6	:*****TEST 6 - ERROR 6*****
1316	003076	104006				:R1 SHOULD BE ZERO
1317				TST	R2	:TEST R2
1318	003100	005702	68\$:	BEG	69\$	:BR IF ZERO
1319	003102	001401		ERROR	7	:*****TEST 6 - ERROR 7*****
1320	003104	104007				:R2 SHOULD BE ZERO
1321				TST	R3	:TEST R3
1322	003106	005703	69\$:	BEG	70\$	:BR, IF ZERO
1323	003110	001401		ERROR	10	:*****TEST 6 - ERROR 10*****
1324	003112	104010				:R3 SHOULD BE ZERO
1325						
1326	003114		70\$:			:VERIFY DESTINATION & NON-OVERLAP SOURCE
1327						:CHECK BEGINNING OF OLD SOURCE AREAS
1328	003114	012700	016434	MOV	#BUF1,R0	
1329	003120	012701	017034	MOV	#BUF2,R1	
1330	003124	122021		CMPB	(R0)+,(R1)+	
1331	003126	001401	3\$:	BEG	4\$	
1332	003130	104011		ERROR	11	:*****TEST 6 - ERROR 11*****
1333						:NON-OVERLAP SOURCE AREA CHANGED
1334						:R0 CONTAINS PC+1 OF CHANGED SOURCE CHARACTER
1335	003132	020167	175514	CMP	R1,DSTAD	:REACHED START OF DESTINATION?
1336	003136	002772		BLT	3\$	:BR, IF NO
1337	003140	012702	016434	MOV	#BUF1,R2	:CHECK DESTINATION AREA
1338	003144	016703	175500	MOV	DSTLN,R3	
1339	003150	122221		CMPB	(R2)+,(R1)+	
1340	003152	001401	5\$:	BEG	6\$	
1341	003154	104012		ERROR	12	:*****TEST 6 - ERROR 12*****
1342						:COMPARE ERROR IN DESTINATION
1343						:R1 CONTAINS PC+1 OF BAD DESTINATION CHARACTER
1344	003156	005200	6\$:	INC	R0	:UPDATE POINTER TO OLD SOURCE
1345	003160	005303		DEC	R3	:DECREMENT DEST. BYTE COUNT
1346	003162	001372		BNE	5\$	:BR, IF NOT FINISHED
1347	003164	016703	175456	MOV	SRCAD,R3	:CHECK LOWER PORTION OF NON-OVERLAP SOURCE
1348	003170	066703	175450	ADD	SRCLN,R3	
1349	003174	122021	7\$:	CMPB	(R0)+,(R1)+	
1350	003176	001401		BEG	10\$	
1351	003200	104013		ERROR	13	:*****TEST 6 - ERROR 13*****
1352						:NON-OVERLAP SOURCE AREA CHANGED
1353						:R0 CONTAINS PC+1 OF CHANGED SOURCE CHARACTER
1354	003202	020103	10\$:	CMP	R1,R3	:FINISHED CHECK?
1355	003204	002773		BLT	7\$	:BR, IF NO

1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411

003206 000004
003210 004567 011776
003214 000007
003216 017037
003220 000017
003222 017035
003224 000377
003226 004767 012006
003232 004567 012030
003236 000007
003240 017037
003242 004567 012040
003246 000211
003250 004767 011672
003254 000251
003256 000266
003260 076030
003262 106767 175404
003266 042767 177400 175376
003274 023767 000674 175370
003302 001401
003304 104001
003306
003306 023706 000676
003312 001403
003314 010637 000700
003320 104002
003322
003322 005700
003324 001401
003326 104003
003330
003330 005701

```

*****
*TEST 7      TEST "MOVC" WITH DSTAD .GT. SRCAD, SRCLN .LT. DSTLN
*****
*THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN
*THE SOURCE AREA IS CONTAINED WITHIN THE DESTINATION AREA.
*THEREFORE, SRCAD>DSTAD AND DSTLN>SRCLN.
*THE RESULT IS A FULL TRANSFER OF THE SOURCE TO THE
*DESTINATION AND "FILL" CHARACTERS IN THE LSB OF THE
*DESTINATION; R0-->R3 EQUAL ZERO, AND CONDITION
*CODES N,C=1 AND Z,V=0
*****
TST7:  SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      7          ;SOURCE LENGTH
      BUF2+3      ;SOURCE ADDRESS
      17          ;DESTINATION LENGTH
      BUF2+1      ;DESTINATION ADDRESS
      377         ;FILL CHARACTER
      JSR      PC,CLDST     ;CLEAR DESTINATION
      JSR      R5,GENSRC    ;GENERATE A SOURCE STRING
      .WORD    7
      .WORD    BUF2+3
      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 CHARACTER" INSTRUCTION
      MOVC
                              ;CHECK RESULTS
      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      2#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$
      ERROR    1
      *****TEST 7 - ERROR 1*****
      PSW ERROR
      EXPECTED PSW IS STORED AT "EXPPSW"
      ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      2#SAVR6,SP    ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$
      MOV      SP,2#BADR6    ;STORE BAD SP VALUE
      ERROR    2
      *****TEST 7 - ERROR 2*****
      STACK POINTER NOT RESTORED BY INSTRUCTION
      EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ERRONEOUS VALUE IS AT "BADR6"
      65$:
      TST      R0            ;CHECK R0 FOR ZERO
      BEQ      1$
      ERROR    3
      *****TEST 7 - ERROR 3*****
      R0 SHOULD BE ZERO
      1$:
      TST      R1            ;TEST R1

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 34
 DVKATA.P11 22-DEC-76 10:47 T7

TEST "MOVC" WITH DSTAD .GT. SRCAD, SRCLN .LT. DSTLN

SEQ 0035

1412	003332	001401		BEQ	66\$		:BR, IF ZERO
1413	003334	104004		ERROR	4		:*****TEST 7 - ERROR 4*****
1414							:R1 SHOULD BE ZERO
1415	003336	005702		66\$: TST	R2		:TEST R2
1416	003340	001401		BEQ	67\$		:BR IF ZERO
1417	003342	104005		ERROR	5		:*****TEST 7 - ERROR 5*****
1418							:R2 SHOULD BE ZERO
1419	003344	005703		67\$: TST	R3		:TEST R3
1420	003346	001401		BEQ	68\$		:BR, IF ZERO
1421	003350	104006		ERROR	6		:*****TEST 7 - ERROR 6*****
1422							:R3 SHOULD BE ZERO
1423	003352			68\$: CMP	FILL,R4		:CHECK R4 UNCHANGED
1424	003352	026704	175276	BEQ	69\$		:BR IF OK
1425	003356	001401		ERROR	7		:*****TEST 7 - ERROR 7*****
1426	003360	104007					:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1427							:CHECK R5 UNCHANGED
1428	003362	026705	175270	69\$: CMP	TABLE,R5		:BR IF OK
1429	003366	001401		BEQ	70\$		:*****TEST 7 - ERROR 10*****
1430	003370	104010		ERROR	10		:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1431							
1432	003372			70\$: MOV	#BUF2,R0		:VERIFY DESTINATION
1433							:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1434	003372	012700	017034	TSTB	(R0)+		:TEST CONTENTS OF BOUNDARY
1435	003376	105720		BEQ	71\$		:BR, IF STILL ZERO
1436	003400	001401		ERROR	11		:*****TEST 7 - ERROR 11*****
1437	003402	104011					:LOWER BOUNDARY OF DESTINATION CHANGED
1438							: SHOULD STILL EQUAL ZERO
1439							
1440	003404			71\$: MOV	#BUF1,R2		:POINT R2 TO ORIGINAL SOURCE BYTES
1441	003404	012702	016434	2\$: MOV	SRCLN,R1		:STORE TRANSFER BYTE COUNT IN R1
1442	003410	016701	175230	77E12: CMPB	(R0)+,(R2)+		:CHECK CHARACTERS IN DESTINATION
1443	003414	122022		BEQ	.+4		:BR IF OK
1444	003416	001401		ERROR	12		:*****TEST 7 - ERROR 12*****
1445	003420	104012					:COMPARE ERROR IN DESTINATION
1446							:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1447							
1448							
1449	003422	005301		DEC	R1		:DECREMENT BYTE COUNT
1450	003424	001373		BNE	77E12		:BR, IF NOT FINISHED CHECKING
1451	003426	016705	175216	MOV	DSTLN,R5		:CHECK FILL
1452	003432	166705	175206	SUB	SRCLN,R5		
1453	003436	122067	175212	3\$: CMPB	(R0)+,FILL		
1454	003442	001401		BEQ	4\$		
1455	003444	104013		ERROR	13		:*****TEST 7 - ERROR 13*****
1456							: "FILL" ERROR IN DESTINATION
1457	003446	105305		4\$: DECB	R5		
1458	003450	001372		BNE	3\$		
1459	003452	105720		TSTB	(R0)+		:TEST CONTENTS OF DEST. UPPER BOUNDARY
1460	003454	001401		BEQ	ENDT7		:BR, IF STILL ZERO
1461	003456	104014		ERROR	14		:*****TEST 7 - ERROR 14*****
1462							:UPPER BOUNDARY OF DEST. CHANGED
1463							: SHOULD STILL EQUAL ZERO
1464	003460			ENDT7:			

```

1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478 003460 000004
1479 003462 004567 011524
1480 003466 000020
1481 003470 017054
1482 003472 000007
1483 003474 017035
1484 003476 000377
1485 003500 004767 011534
1486 003504 004567 011556
1487 003510 000020
1488 003512 017054
1489 003514 004567 011566
1490 003520 000200
1491 003522 004767 011420
1492 003526 000277
1493
1494 003530 076030
1495
1496 003532 106767 175134
1497 003536 042767 177400 175126
1498 003544 023767 000674 175120
1499 003552 001401
1500 003554 104001
1501
1502
1503
1504 003556
1505 003556 023706 000676
1506 003562 001403
1507 003564 010637 000700
1508 003570 104002
1509
1510
1511
1512 003572
1513 003572 026704 175056
1514 003576 001401
1515 003600 104003
1516
1517 003602 026705 175050
1518 003606 001401
1519 003610 104004
1520

```

```

*****
*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
      .WORD    20
      .WORD    BUF2+20
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD    200
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC       ;SET ALL CONDITION CODES
      MOVC      ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
      ERROR    1            ;*****TEST 10 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED AT "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 10 - 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 10 - 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 10 - ERROR 4*****
      ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```


K03

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 36
DVKAIA.P11 22-DEC-76 10:47 T10

TEST "MOVC" WITH DSTAD .LT. SRCAD & SRCLN .GT. DSTLN, (TRUNCATION)

SEQ 0037

```

1521 003612 67$: MOV SRCLN,R5 ;CHECK R0=SOURCE LENGTH MINUS
1522 003612 016705 175026 SUB DSTLN,R5 ; DESTINATION LENGTH
1523 003616 166705 175026 CMP R0,R5
1524 003622 020005 BEQ 68$
1525 003624 001401 ERROR 5 ;*****TEST 10 - ERROR 5*****
1526 003626 104005 ;R0 NOT EQUAL TO (SRCLN-DSTLN)
1527 ;CHECK OTHER GENERAL REGISTERS
1528 003630 68$: TST R1 ;TEST R1
1529 003630 005701 BEQ 69$ ;BR, IF ZERO
1530 003632 001401 ERROR 6 ;*****TEST 10 - ERROR 5*****
1531 003634 104006 ;R1 SHOULD BE ZERO
1532 ;TEST R2
1533 003636 005702 69$: TST R2 ;TEST R2
1534 003640 001401 BEQ 70$ ;BR IF ZERO
1535 003642 104007 ERROR 7 ;*****TEST 10 - ERROR 7*****
1536 ;R2 SHOULD BE ZERO
1537 003644 005703 70$: TST R3 ;TEST R3
1538 003646 001401 BEQ 71$ ;BR, IF ZERO
1539 003650 104010 ERROR 10 ;*****TEST 10 - ERROR 10*****
1540 ;R3 SHOULD BE ZERO
1541 003652 71$: ;VERIFY DESTINATION AREA
1542 ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1543 003652 012700 017034 MOV #BUF2,R0
1544 003656 105720 TSTB (R0)+
1545 003660 001401 BEQ 72$ ;TEST CONTENTS OF BOUNDARY
1546 003662 104011 ERROR 11 ;BR, IF STILL ZERO
1547 ;*****TEST 10 - ERROR 11*****
1548 ;LOWER BOUNDARY OF DESTINATION CHANGED
1549 ; SHOULD STILL EQUAL ZERO
1550 003664 72$: MOV SRCAD,R2
1551 003670 016701 174756 MOV DSTLN,R1
1552 003674 122022 T10E12: CMPB (R0)+,(R2)+
1553 003676 001401 BEQ +4 ;STORE TRANSFER BYTE COUNT IN R1
1554 003700 104012 ERROR 12 ;CHECK CHARACTERS IN DESTINATION
1555 ;BR IF OK
1556 ;*****TEST 10 - ERROR 12*****
1557 ;COMPARE ERROR IN DESTINATION
1558 ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1559 003702 005301 DEC R1 ;DECREMENT BYTE COUNT
1560 003704 001373 BNE T10E12 ;BR, IF NOT FINISHED CHECKING
1561 003706 105720 TSTB (R0)+ ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1562 003710 001401 BEQ ENDT10 ;BR, IF STILL ZERO
1563 003712 104013 ERROR 13 ;*****TEST 10 - ERROR 13*****
1564 ;UPPER BOUNDARY OF DEST. CHANGED
1565 003714 ENDT10: ; SHOULD STILL EQUAL ZERO
1566

```

```

1567
1568
1569
1570
1571
1572
1573
1574
1575
1576 003714 000004
1577 003716 004567 011270
1578 003722 016434
1579 003724 000010
1580 003726 017034
1581 003730 000010
1582 003732 000377
1583 003734 012737 076032 003762
1584 003742 013767 000010 174736
1585 003750 012737 003772 000010
1586 003755 004767 011164
1587
1588 003762 076032
1589
1590 003764 016700 177772
1591 003770 104001
1592
1593
1594
1595 003772 012626
1596 003774 005267 177762
1597 004000 022767 076040 177754
1598 004006 001363
1599 004010 016737 174672 000010
1600

```

```

;*****
;TEST 11      TEST THAT BAD "MOVE" OPCODES TRAP
;*****
;THIS TEST VERIFIES THAT OPCODES 076032-->076037 TRAP TO
;LOCATION 10
;*****
;*****
TST11:  SCOPE
        JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
        BUF1     ;SOURCE LENGTH
        10       ;SOURCE ADDRESS
        BUF2     ;DESTINATION LENGTH
        10       ;DESTINATION ADDRESS
        377      ;FILL CHARACTER
        MOV      #076032, @BD11 ;STORE THE FIRST BAD MOVE OPCODE
        MOV      @#10, TEMP1    ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR
        MOV      #T11CONT, @#10 ;POINT ILLEGAL INSTRUCTION VECTOR TO CONTINUE TEST
REP11:  JSR      PC,GENR        ;SET UP GENERAL REGISTERS

BD11:   .WORD    076032        ;EXECUTE BAD MOVE INSTRUCTION

        MOV      BD11, R0      ;STORE BAD OPCODE THAT DID NOT TRAP
        ERROR    1            ;*****TEST 11 - ERROR 1*****
        ;BAD MOVE OPCODE DID NOT TRAP
        ;R0 CONTAINS THE BAD OPCODE

T11CONT:MOV      (SP)+, (SP)+  ;RESTORE THE STACK POINTER AFTER THE TRAP
        INC      BD11         ;INCREMENT INSTRUCTION OPCODE
        CMP      #076040, BD11 ;FINISHED WITH BAD MOVE OPCODES?
        BNE     REP11         ;BR IF NOT
        MOV      TEMP1, @#10  ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR

```



```

1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615 004016 000004
1616 004020 105777 174514
1617 004024 100570
1618 004026 026767 174516 174624
1619 004034 001007
1620 004036 032767 000001 174542
1621 004044 001403
1622 004046 005767 174522
1623 004052 001155
1624 004054
1625 004054 004567 011132
1626 004060 000374
1627 004062 016434
1628 004064 000376
1629 004066 017035
1630 004070 000377
1631 004072 004767 011142
1632 004076 012767 004160 174564
1633 004104 012777 015356 174552
1634 004112 005077 174550
1635 004116 004767 011210
1636 004122 013777 000554 174532
1637 004130 004567 011152
1638 004134 000011
1639 004136 106427 000000
1640 004142 052777 000100 174510
1641 004150 004767 010772
1642 004154 000251
1643 004156 000266
1644
1645 004160 076030
1646
1647 004162 106767 174504
1648 004166 032777 000100 174464
1649 004174 001365
1650 004176 042767 177400 174466
1651 004204 023767 000674 174460
1652 004212 001401
1653 004214 104001
1654
1655
1656

;*****
;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
TSTB @SWR ;TEST BIT 7 OF SWR
BMI TST13 ;SKIP TO NEXT TEST IF SET
CMP $TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T12CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT @BIT0, $ENV ;IF YES, CHECK IF ON APT
BEQ T12CONT ;BR IF NOT UNDER APT, AND DO THIS TEST
TST $PASS ;CHECK IF ON FIRST PASS
BNE TST13 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T12CONT: JSR R5, PREP ;SET UP INSTRUCTION ARGUMENTS
;SOURCE LENGTH
;SOURCE ADDRESS
;DESTINATION LENGTH
;DESTINATION ADDRESS
;FILL CHARACTER
;CLEAR DESTINATION
JSR PC, CLDST ;STORE PC OF TEST INSTRUCTION
MOV @MC, PC ;POINT TTY VECTOR TO INTERRUPT ROUTINE
MOV @INTR, @TVECT ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
CLR @TPSW
JSR PC, TDONE
MOV @R, @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
JSR PC, GENR ;SET UP GENERAL REGISTERS
+CLN!CLC
+SEV!SEZ

REPMC: JSR PC, GENR

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"

```

1657	004216			64\$:					
1658									
1659	004216	023706	000676		CMP	2#SAVR6, SP			;CHECK RESULTS
1660	004222	001403			BEQ	65\$			;VERIFY STACK POINTER IS RESTORED
1661	004224	010637	000700		MOV	SP, 2#BADR6			;BR IF OK
1662	004230	104002			ERROR	2			;STORE BAD SP VALUE
1663									;*****TEST 12 - ERROR 2*****
1664									;STACK POINTER NOT RESTORED BY INSTRUCTION
1665									;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
1666	004232			65\$:					;ERRONEOUS VALUE IS AT "BADR6"
1667	004232	005700			TST	R0			;CHECK R0=ZERO
1668	004234	001401			BEQ	1\$			
1669	004236	104003			ERROR	3			;*****TEST 12 - ERROR 3*****
1670									;R0 SHOULD BE ZERO
1671	004240			1\$:					;CHECK OTHER GENERAL REGISTERS
1672	004240	005701			TST	R1			;TEST R1
1673	004242	001401			BEQ	66\$			;BR, IF ZERO
1674	004244	104004			ERROR	4			;*****TEST 12 - ERROR 4*****
1675									;R1 SHOULD BE ZERO
1676	004246	005702		66\$:	TST	R2			;TEST R2
1677	004250	001401			BEQ	67\$			;BR IF ZERO
1678	004252	104005			ERROR	5			;*****TEST 12 - ERROR 5*****
1679									;R2 SHOULD BE ZERO
1680	004254	005703		67\$:	TST	R3			;TEST R3
1681	004256	001401			BEQ	68\$			;BR, IF ZERO
1682	004260	104006			ERROR	6			;*****TEST 12 - ERROR 6*****
1683									;R3 SHOULD BE ZERO
1684	004262			68\$:					
1685	004262	026704	174366		CMP	FILL, R4			;CHECK R4 UNCHANGED
1686	004266	001401			BEQ	69\$			;BR IF OK
1687	004270	104007			ERROR	7			;*****TEST 12 - ERROR 7*****
1688									;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1689	004272	026705	174360	69\$:	CMP	TABLE, R5			;CHECK R5 UNCHANGED
1690	004276	001401			BEQ	70\$			;BR IF OK
1691	004300	104010			ERROR	10			;*****TEST 12 - ERROR 10*****
1692									;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1693	004302			70\$:					
1694	004302	012700	017034		MOV	#BUF2, R0			;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1695	004306	012700	017034		MOV	#BUF2, R0			;TEST CONTENTS OF BOUNDARY
1696	004312	105720			TSTB	(R0)+			;BR, IF STILL ZERO
1697	004314	001401			BEQ	71\$			;*****TEST 12 - ERROR 11*****
1698	004316	104011			ERROR	11			;LOWER BOUNDARY OF DESTINATION CHANGED
1699									; SHOULD STILL EQUAL ZERO
1700									
1701	004320			71\$:					
1702	004320	016702	174322	4\$:	MOV	SRCAD, R2			;POINT R2 TO THE SOURCE ADDRESS
1703	004324	016701	174314		MOV	SRCLN, R1			;STORE TRANSFER BYTE COUNT IN R1
1704	004330	122022		T12E12:	CMPB	(R0)+, (R2)+			;CHECK CHARACTERS IN DESTINATION
1705	004332	001401			BEQ	+4			;BR IF OK
1706	004334	104012			ERROR	12			;*****TEST 12 - ERROR 12*****
1707									;COMPARE ERROR IN DESTINATION
1708									;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1709									
1710	004336	005301			DEC	R1			;DECREMENT BYTE COUNT
1711	004340	001373			BNE	T12E12			;BR, IF NOT FINISHED CHECKING
1712	004342	016701	174302		MOV	DSTLN, R1			;CALCULATE THE NUMBER OF "FILL"

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 40
 DVKAIA.P11 22-DEC-76 10:47 T12

TEST INTERRUPTABILITY OF "MOVC" INSTRUCTION

SEQ 0041

1713	004346	166701	174272		SUB	SRCLN,R1	:CHARACTERS THAT SHOULD APPEAR IN DEST.
1714	004352	122067	174276	5\$:	CMPB	(R0)+,FILL	:CHECK LSB'S OF DEST. FOR FILLS
1715	004356	001401			BEQ	6\$	:BR, IF EQUAL
1716	004360	104013			ERROR	13	:*****TEST 12 - ERROR 13*****
1717							: "FILL" ERROR IN DESTINATION
1718	004362	005301		6\$:	DEC	R1	:DECREMENT FILL COUNT
1719	004364	001372			BNE	5\$	:BR, IF NOT FINISHED
1720	004366	105720			TSTB	(R0)+	:TEST CONTENTS OF DEST. UPPER BOUNDARY
1721	004370	001401			BEQ	ENDT12	:BR, IF STILL ZERO
1722	004372	104014			ERROR	14	:*****TEST 12 - ERROR 14*****
1723							:UPPER BOUNDARY OF DEST. CHANGED
1724							: SHOULD STILL EQUAL ZERO
1725	004374			ENDT12:			
1726							
1727							
1728	004374	106427	000200	40\$:	MTPS	#200	:RESTORE PSW TO PRIORITY 7
1729	004400	016777	174262 174256		MOV	TPSW,QTVECT	:RESTORE TRAP CATCHER

```

1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744 004406 000004
1745
1746 004410 004567 010576
1747 004414 000020
1748 004416 016434
1749 004420 000011
1750 004422 017035
1751 004424 000377
1752 004426 004767 010606
1753 004432 004567 010650
1754 004436 000200
1755 004440 004767 010502
1756 004444 000277
1757
1758 004446 076031
1759
1760 004450 106767 174216
1761 004454 042767 177400 174210
1762 004462 023767 000674 174202
1763 004470 001401
1764 004472 104001
1765
1766
1767
1768 004474
1769 004474 023706 000676
1770 004500 001403
1771 004502 010637 000700
1772 004506 104002
1773
1774
1775
1776 004510
1777 004510 026704 174140
1778 004514 001401
1779 004516 104003
1780
1781 004520 026705 174132
1782 004524 001401
1783 004526 104004
1784
1785 004530

```

```

*****
TEST 13 TEST "MOVC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), SL .GT. DL
*****
*PROPER TERMINATION FOR THIS INSTRUCTION TEST IS A
*TRUNCATED SOURCE REVERSE (HIGH ADDRESS) JUSTIFIED
*(WITH MOST SIGNIFICANT BYTES NOT MOVED) IN THE
*DESTINATION. R0 EQUALS THE NUMBER OF UNMOVED SOURCE
*BYTES, R1-->R3 EQUAL ZERO AND ALL CONDITION CODES
*CLEAR
*****
TST13: SCOPE

```

```

JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
20 ;SOURCE LENGTH
BUF1 ;SOURCE ADDRESS
11 ;DESTINATION LENGTH
BUF2+1 ;DESTINATION ADDRESS
377 ;FILL CHARACTER
JSR PC,CLOST ;CLEAR DESTINATION
JSR R5,XPSW ;STORE EXPECTED PSW VALUE
.WORD 200
JSR PC,GENR ;SET UP GENERAL REGISTERS
SCC ;SET ALL CONDITION CODES
;EXECUTE "MOVE REVERSE CHARACTER"

```

```

;CHECK RESULTS
;STORE THE PSW
;CLEAR ALL UNUSED BITS
;CHECK PSW AGAINST EXPECTED VALUE
;BR, IF EQUAL
*****TEST 13 - 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 13 - 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 13 - ERROR 3*****
;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
;CHECK R5 UNCHANGED
;BR IF OK
66$:
CMP TABLE,R5 ;CHECK R5 UNCHANGED
BEQ 67$ ;BR IF OK
ERROR 4 ;*****TEST 13 - ERROR 4*****
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```

```

67$:

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 42
 DVKAIA.P11 22-DEC-76 10:47 T13

TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), SL .GT. DL

SEQ 0043

1786	004530	016705	174110	MOV	SRCLN,R5	;CHECK R0=SOURCE LENGTH MINUS
1787	004534	166705	174110	SUB	DSTLN,R5	; DESTINATION LENGTH
1788	004540	020005		CMP	R0,R5	
1789	004542	001401		BEQ	68\$	
1790	004544	104005		ERROR	5	*****TEST 13 - ERROR 5*****
1791						;R0 NOT EQUAL TO (SRCLN-DSTLN)
1792	004546					;CHECK OTHER GENERAL REGISTERS
1793	004546	005701		TST	R1	;TEST R1
1794	004550	001401		BEQ	69\$	;BR, IF ZERO
1795	004552	104006		ERROR	6	*****TEST 13 - ERROR 6*****
1796						;R1 SHOULD BE ZERO
1797	004554	005702		TST	R2	;TEST R2
1798	004556	001401		BEQ	70\$	;BR IF ZERO
1799	004560	104007		ERROR	7	*****TEST 13 - ERROR 7*****
1800						;R2 SHOULD BE ZERO
1801	004562	005703		TST	R3	;TEST R3
1802	004564	001401		BEQ	71\$	;BR, IF ZERO
1803	004566	104010		ERROR	10	*****TEST 13 - ERROR 10*****
1804						;R3 SHOULD BE ZERO
1805	004570					
1806						;VERIFY DESTINATION AREA
1807	004570	012700	017034	MOV	#BUF2,R0	;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1808	004574	105720		TSTB	(R0)+	;TEST CONTENTS OF BOUNDARY
1809	004576	001401		BEQ	72\$	;BR, IF STILL ZERO
1810	004600	104011		ERROR	11	*****TEST 13 - ERROR 11*****
1811						;LOWER BOUNDARY OF DESTINATION CHANGED
1812						; SHOULD STILL EQUAL ZERO
1813	004602					
1814	004602	016702	174036	MOV	SRCLN,R2	;CALCULATE ADDRESS OF MSB OF THE
1815	004606	166702	174036	SUB	DSTLN,R2	; SOURCE TO BE MOVED
1816	004612	066702	174030	ADD	SRCAD,R2	
1817	004616	016701	174026	MOV	DSTLN,R1	;STORE TRANSFER BYTE COUNT IN R1
1818	004622	122022		T13E12: CMPB	(R0)+,(R2)+	;CHECK CHARACTERS IN DESTINATION
1819	004624	001401		BEQ	+4	;BR IF OK
1820	004626	104012		ERROR	12	*****TEST 13 - ERROR 12*****
1821						;COMPARE ERROR IN DESTINATION
1822						;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1823						
1824	004630	005301		DEC	R1	;DECREMENT BYTE COUNT
1825	004632	001373		BNE	T13E12	;BR, IF NOT FINISHED CHECKING
1826	004634	105720		TSTB	(R0)+	;TEST CONTENTS OF DEST. UPPER BOUNDARY
1827	004636	001401		BEQ	ENDT13	;BR, IF STILL ZERO
1828	004640	104013		ERROR	13	*****TEST 13 - ERROR 13*****
1829						;UPPER BOUNDARY OF DEST. CHANGED
1830						; SHOULD STILL EQUAL ZERO
1831	004642			ENDT13:		

1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887

004642 000004
004644 004567 010342
004650 000011
004652 016434
004654 000020
004656 017035
004660 000377
004662 004767 010352
004666 004567 010414
004672 000211
004674 004767 010246
004700 000251
004702 000266
004704 076031
004706 10671 173760
004712 042767 177400 173752
004720 023767 000674 173744
004726 001401
004730 104001
004732
004732 023706 000676
004736 001403
004740 010637 000700
004744 104002
004746
004746 005700
004750 001401
004752 104003
004754
004754 005701
004756 001401
004760 104004

```

*****
*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
                                ;SOURCE LENGTH
                                ;SOURCE ADDRESS
                                ;DESTINATION LENGTH
                                ;DESTINATION ADDRESS
                                ;FILL CHARACTER
        JSR      PC,CLDST     ;CLEAR DESTINATION
        JSR      5,XPSW       ;STORE EXPECTED PSW VALUE
        .WORD    211
        JSR      PC,GENR      ;SET UP GENERAL REGISTERS
                                ;CLEAR CONDITION CODES N&C
                                ;SET CONDITION CODES V&Z
        +CLN:CLC              ;EXECUTE "MOVE REVERSE CHARACTER"
        +SEV:SEZ
        MOVRC
                                ;CHECK RESULTS
                                ;STORE THE PSW
        MFPS      CCODES      ;CLEAR ALL UNUSED BITS
        BIC      #177400,CCODES
        CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        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
                                ;*****TEST 14 - ERROR 3*****
                                ;R0 SHOULD BE ZERO
                                ;CHECK OTHER GENERAL REGISTERS
        1$:
        TST      R1
        BEQ      66$
        ERROR    4
                                ;*****TEST 14 - ERROR 4*****
                                ;R1 SHOULD BE ZERO

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 44
 DVKAIA.P11 22-DEC-76 10:47 T14

TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), DL .GT. SL

SEQ 0045

1888	004762	005702		66\$:	TST R2		:TEST R2
1889	004764	001401			BEQ 67\$		:BR IF ZERO
1890	004766	104005			ERROR 5		:*****TEST 14 - ERROR 5*****
1891							:R2 SHOULD BE ZERO
1892	004770	005703		67\$:	TST R3		:TEST R3
1893	004772	001401			BEQ 68\$		:BR, IF ZERO
1894	004774	104006			ERROR 6		:*****TEST 14 - ERROR 6*****
1895							:R3 SHOULD BE ZERO
1896	004776			68\$:			
1897	004776	026704	173652		CMP FILL,R4		:CHECK R4 UNCHANGED
1898	005002	001401			BEQ 69\$		:BR IF OK
1899	005004	104007			ERROR 7		:*****TEST 14 - ERROR 7*****
1900							:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1901	005006	026705	173644	69\$:	CMP TABLE,R5		:CHECK R5 UNCHANGED
1902	005012	001401			BEQ 70\$		:BR IF OK
1903	005014	104010			ERROR 10		:*****TEST 14 - ERROR 10*****
1904							:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1905	005016			70\$:			
1906	005016	012700	017034		MOV #BUF2,R0		:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1907	005022	105720			TSTB (R0)+		:TEST CONTENTS OF BOUNDARY
1908	005024	001401			BEQ 71\$		:BR, IF STILL ZERO
1909	005026	104011			ERROR 11		:*****TEST 14 - ERROR 11*****
1910							:LOWER BOUNDARY OF DESTINATION CHANGED
1911							: SHOULD STILL EQUAL ZERO
1912	005030			71\$:			
1913	005030	016702	173614		MOV DSTLN,R2		:CALCULATE THE NUMBER OF "FILL"
1914	005034	166702	173604		SUB SRCLN,R2		:CHARACTERS THAT SHOULD APPEAR IN DEST.
1915	005040	010201			MOV R2,R1	:STORE	:TRANSFER BYTE COUNT IN R1
1916	005042	122067	173606	T14E12:	CMPB (R0)+,FILL		:CHECK CHARACTERS IN DESTINATION
1917	005046	001401			BEQ +4		:BR IF OK
1918	005050	104012			ERROR 12		:*****TEST 14 - ERROR 12*****
1919							:COMPARE ERROR IN DESTINATION
1920							:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1921							
1922	005052	005301			DEC R1		:DECREMENT BYTE COUNT
1923	005054	001372			BNE T14E12		:BR, IF NOT FINISHED CHECKING
1924	005056	016702	173564		MOV SRCAD,R2		:POINT R2 TO SOURCE STRING
1925	005062	016701	173556		MOV SRCLN,R1		:STORE TRANSFER BYTE COUNT IN R1
1926	005066	122022		T14E13:	CMPB (R0)+,(R2)+		:CHECK CHARACTERS IN DESTINATION
1927	005070	001401			BEQ +4		:BR IF OK
1928	005072	104013			ERROR 13		:*****TEST 14 - ERROR 13*****
1929							:COMPARE ERROR IN DESTINATION
1930							:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1931							
1932	005074	005301			DEC R1		:DECREMENT BYTE COUNT
1933	005076	001373			BNE T14E13		:BR, IF NOT FINISHED CHECKING
1934	005100	105720			TSTB (R0)+		:TEST CONTENTS OF DEST. UPPER BOUNDARY
1935	005102	001401			BEQ ENDT14		:BR, IF STILL ZERO
1936	005104	104014			ERROR 14		:*****TEST 14 - ERROR 14*****
1937							:UPPER BOUNDARY OF DEST. CHANGED
1938							: SHOULD STILL EQUAL ZERO
1939	005106			ENDT14:			

G04

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 45
DVKAIA.F11 22-DEC-76 10:47 T14

TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), DL .GT. SL

SEQ 0046

1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995

005106	000004	
005110	004567	010076
005114	000020	
005116	017054	
005120	000007	
005122	017035	
005124	000377	
005126	004767	010106
005132	004567	010130
005136	000020	
005140	017054	
005142	004567	010140
005146	000200	
005150	004767	007772
005154	000277	
005156	076031	
005160	106767	173506
005164	042767	177400 173500
005172	023767	000674 173472
005200	001401	
005202	104001	
005204		
005204	023706	000676
005210	001403	
005212	010637	000700
005216	104002	
005220		
005220	026704	173430
005224	001401	
005226	104003	
005230	026705	173422
005234	001401	
005236	104004	

```

*****
*TEST 15      TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB), SL .GT. DL, (TRUNCATED)
*****
*THIS TEST CHECKS "MOVE REVERSE" WHEN THE LEAST SIGNIFICANT
*BYTE OF THE SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION LSB
*AND THE TRANSFER IS TRUNCATED.
*THE RESULT IS THE MOST SIGNIFICANT BYTES ARE TRUNCATED, R0 = THE
*NUMBER OF BYTES TRUNCATED, R1-->R3 EQUAL ZERO, AND ALL
*CONDITION CODES CLEAR.
*****
+ST15:  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 A SOURCE STRING
        .WORD      20
        .WORD      BUF2+20
        JSR      R5,XPSW         ;STORE EXPECTED PSW VALUE
        .WORD      200
        JSR      PC,GENR        ;SET UP GENERAL REGISTERS
        SCC          ;SET ALL CONDITION CODES

        MOVRC          ;EXECUTE "MOVE REVERSE CHARACTER" INSTRUCTION

        MFPS          CCODES      ;STORE THE PSW
        BIC          #177400,CCODES ;CLEAR ALL UNUSED BITS
        CMP          @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ          64$         ;BR, IF EQUAL
        ERROR        1          ;*****TEST 15 - 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 15 - ERROR 2*****
        STACK POINTER NOT RESTORED BY INSTRUCTION
        EXPECTED VALUE OF SP IS STORED AT "SAVR6"
        ERRONEOUS VALUE IS AT "BADR6"

        65$:
        CML          FILL,R4     ;CHECK R4 UNCHANGED
        BEQ          66$         ;BR IF OK
        ERROR        3          ;*****TEST 15 - ERROR 3*****
        R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
        CHECK R5 UNCHANGED
        BEQ          67$         ;BR IF OK
        ERROR        4          ;*****TEST 15 - ERROR 4*****
        R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```


1996	005240			67%:	MOV	SRCLN,R5	:CHECK R0=SOURCE LENGTH MINUS
1997	005240	016705	173400		SUB	DSTLN,R5	: DESTINATION LENGTH
1998	005244	166705	173400		CMP	R0,R5	
1999	005250	020005			BEQ	68%	
2000	005252	001401			ERROR	5	:*****TEST 15 - ERROR 5*****
2001	005254	104005					:R0 NOT EQUAL TO (SRCLN-DSTLN)
2002							:CHECK OTHER GENERAL REGISTERS
2003	005256			68%:	TST	R1	:TEST R1
2004	005256	005701			BEQ	69%	:BR, IF ZERO
2005	005260	001401			ERROR	6	:*****TEST 15 - ERROR 6*****
2006	005262	104006					:R1 SHOULD BE ZERO
2007							:TEST R2
2008	005264	005702		69%:	TST	R2	:BR IF ZERO
2009	005266	001401			BEQ	70%	:*****TEST 15 - ERROR 7*****
2010	005270	104007			ERROR	7	:R2 SHOULD BE ZERO
2011							:TEST R3
2012	005272	005703		70%:	TST	R3	:BR, IF ZERO
2013	005274	001401			BEQ	71%	:*****TEST 15 - ERROR 10*****
2014	005276	104010			ERROR	10	:R3 SHOULD BE ZERO
2015							
2016	005300			71%:			:VERIFY DESTINATION AREA
2017							:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2018	005300	012700	017034		MOV	#BUF2,R0	:TEST CONTENTS OF BOUNDARY
2019	005304	105720			TSTB	(R0)+	:BR, IF STILL ZERO
2020	005306	001401			BEQ	72%	:*****TEST 15 - ERROR 11*****
2021	005310	104011			ERROR	11	:LOWER BOUNDARY OF DESTINATION CHANGED
2022							: SHOULD STILL EQUAL ZERO
2023							
2024	005312			72%:	MOV	SRCLN,R2	:CALCULATE ADDRESS OF MSB OF
2025	005312	016702	173326		SUB	DSTLN,R2	: THE SOURCE TO BE MOVED
2026	005316	166702	173326		ADD	SRCAD,R2	
2027	005322	066702	173320		MOV	DSTLN,R1	:STORE TRANSFER BYTE COUNT IN R1
2028	005326	016701	173316		CMPB	(R0)+,(R2)+	:CHECK CHARACTERS IN DESTINATION
2029	005332	122022		T15E12:	BEQ	+4	:BR IF OK
2030	005334	001401			ERROR	12	:*****TEST 15 - ERROR 12*****
2031	005336	104012					:COMPARE ERROR IN DESTINATION
2032							:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2033							
2034							
2035	005340	005301			DEC	R1	:DECREMENT BYTE COUNT
2036	005342	001373			BNE	T15E12	:BR, IF NOT FINISHED CHECKING
2037	005344	105720			TSTB	(R0)+	:TEST CONTENTS OF DEST. UPPER BOUNDARY
2038	005346	001401			BEQ	ENDT15	:BR, IF STILL ZERO
2039	005350	104013			ERROR	13	:*****TEST 15 - ERROR 13*****
2040							:UPPER BOUNDARY OF DEST. CHANGED
2041							: SHOULD STILL EQUAL ZERO
2042	005352			ENDT15:			
2043							

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 47
 DVKATA.P11 22-DEC-76 10:47 T15 TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB), SL .GT. DL. (TRUNCATED) SEQ 0048

```

2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056 005352 000004
2057 005354 004567 007632
2058 005360 000010
2059 005362 017054
2060 005364 000015
2061 005366 017035
2062 005370 000377
2063 005372 004767 007642
2064 005376 004567 007664
2065 005402 000010
2066 005404 017054
2067 005406 004567 007674
2068 005412 000211
2069 005414 004767 007526
2070 005420 000251
2071 005422 000266
2072
2073 005424 076031
2074
2075 005426 106767 173240
2076 005432 042767 177400 173232
2077 005440 023767 000674 173224
2078 005446 001401
2079 005450 104001
2080
2081
2082
2083 005452
2084 005452 023706 000676
2085 005456 001403
2086 005460 010637 000700
2087 005464 104002
2088
2089
2090
2091 005466
2092 005466 005700
2093 005470 001401
2094 005472 104003
2095
2096 005474
2097 005474 005701
2098 005476 001401
2099 005500 104004

```

```

*****
*TEST 16      TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB) & SRCLN .GT. DSTLN. (FILL)
*****
*THIS TEST CHECKS "MOVE REVERSE" WHEN THE LEAST SIGNIFICANT
*BYTE OF THE SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION LSB
*AND ALL BYTES ARE TRANSFERED.
*THE RESULT IS THE MSB OF THE DESTINATION ARE FILLS, R0-->R3 EQUAL
*ZERO, AND CONDITION CODES N,C=1 & Z,V=0.
*****
TST16: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      10      ;SOURCE LENGTH
      BUF2+20 ;SOURCE ADDRESS
      15      ;DESTINATION LENGTH
      BUF2+1  ;DESTINATION ADDRESS
      377     ;FILL CHARACTER
      JSR      PC,CLDST     ;CLEAR DESTINATION AREA
      JSR      R5,GENSRC    ;GENERATE SOURCE STRING
      .WORD    10
      .WORD    BUF2+20
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD    211
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      +CLN!CLC
      +SEV!SEZ

      MOVRC                ;EXECUTE "MOVE REVERSE CHARACTER"

      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$         ;BR, IF EQUAL
      *****TEST 16 - 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 16 - ERROR 2*****
      STACK POINTER NOT RESTORED BY INSTRUCTION
      EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ERRONEOUS VALUE IS AT "BADR6"

65$:
      TST      R0
      BEQ      3$
      ERROR    3
      *****TEST 16 - ERROR 3*****
      R0 SHOULD BE ZERO

3$:
      TST      R1
      BEQ      66$
      ERROR    4
      *****TEST 16 - ERROR 4*****

```


0100						:R1 SHOULD BE ZERO
0101	005502	005702		66\$: TST R2		:TEST R2
0102	005504	001401		BEG 67\$		:BR IF ZERO
0103	005506	104005		ERROR 5		:*****TEST 16 - ERROR 5*****
0104						:R2 SHOULD BE ZERO
0105	005510	005703		67\$: TST R3		:TEST R3
0106	005512	001401		BEG 68\$		:BR, IF ZERO
0107	005514	104006		ERROR 6		:*****TEST 16 - ERROR 6*****
0108						:R3 SHOULD BE ZERO
0109	005516			68\$: CMP FILL,R4		:CHECK R4 UNCHANGED
0110	005516	026704	173132	BEG 69\$		:BR IF OK
0111	005522	001401		ERROR 7		:*****TEST 16 - ERROR 7*****
0112	005524	104007				:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
0113						:CHECK R5 UNCHANGED
0114	005526	026705	173124	69\$: CMP TABLE,R5		:BR IF OK
0115	005532	001401		BEG 70\$		:*****TEST 16 - ERROR 10*****
0116	005534	104010		ERROR 10		:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
0117						
0118	005536			70\$: MOV #BUF2,R0		:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
0119	005536	012700	017034	TSTB (R0)+		:TEST CONTENTS OF BOUNDARY
0120	005542	105720		BEG 71\$		:BR, IF STILL ZERO
0121	005544	001401		ERROR 11		:*****TEST 16 - ERROR 11*****
0122	005546	104011				:LOWER BOUNDARY OF DESTINATION CHANGED
0123						: SHOULD STILL EQUAL ZERO
0124						
0125	005550			71\$: MOV DSTLN,R2		:CALCULATE "FILL" COUNT
0126	005550	016702	173074	SUB SRCLN,R2		
0127	005554	166702	173064	MOV R2,R1 ;STORE	TRANSFER BYTE COUNT IN R1	
0128	005560	010201		CMPB (R0)+,FILL		:CHECK CHARACTERS IN DESTINATION
0129	005562	122067	173066	BEG +4		:BR IF OK
0130	005566	001401		ERROR 12		:*****TEST 16 - ERROR 12*****
0131	005570	104012				:COMPARE ERROR IN DESTINATION
0132						:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
0133						
0134						
0135	005572	005301		DEC R1		:DECREMENT BYTE COUNT
0136	005574	001372		BNE T16E12		:BR, IF NOT FINISHED CHECKING
0137	005576	016702	173044	MOV SRCAD,R2		
0138	005602	016701	173036	MOV SRCLN,R1		:STORE TRANSFER BYTE COUNT IN R1
0139	005606	122022		CMPB (R0)+,(R2)+		:CHECK CHARACTERS IN DESTINATION
0140	005610	001401		BEG +4		:BR IF OK
0141	005612	104013		ERROR 13		:*****TEST 16 - ERROR 13*****
0142						:COMPARE ERROR IN DESTINATION
0143						:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
0144						
0145	005614	005301		DEC R1		:DECREMENT BYTE COUNT
0146	005616	001373		BNE T16E13		:BR, IF NOT FINISHED CHECKING
0147	005620	105720		TSTB (R0)+		:TEST CONTENTS OF DEST. UPPER BOUNDARY
0148	005622	001401		BEG ENDT16		:BR, IF STILL ZERO
0149	005624	104014		ERROR 14		:*****TEST 16 - ERROR 14*****
0150						:UPPER BOUNDARY OF DEST. CHANGED
0151						: SHOULD STILL EQUAL ZERO
0152						
0153	005626			ENDT16:		

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

PC	OPCODE	OP1	OP2	OP3	OP4	COMMENT
2210						;ACTUAL PSW IS STORED AT "CCODES"
2211	006032					
2212						
2213	006032	023706	000676			;CHECK RESULTS
2214	006036	001403				;VERIFY STACK POINTER IS RESTORED
2215	006040	010637	000700			;BR IF OK
2216	006044	104002				;STORE BAD SP VALUE
2217						*****TEST 17 - ERROR 2*****
2218						;STACK POINTER NOT RESTORED BY INSTRUCTION
2219						;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2220						;ERRONEOUS VALUE IS AT "BADR6"
2221	006046					
2222	006046	005700				;CHECK R0=ZERO
2223	006050	001401				
2224	006052	104003				*****TEST 17 - ERROR 3*****
2225						;R0 SHOULD BE ZERO
2226	006054					;CHECK OTHER GENERAL REGISTERS
2227	006054	005701				;TEST R1
2228	006056	001401				;BR, IF ZERO
2229	006060	104004				*****TEST 17 - ERROR 4*****
2230						;R1 SHOULD BE ZERO
2231	006062	005702				;TEST R2
2232	006064	001401				;BR IF ZERO
2233	006066	104005				*****TEST 17 - ERROR 5*****
2234						;R2 SHOULD BE ZERO
2235	006070	005703				;TEST R3
2236	006072	001401				;BR, IF ZERO
2237	006074	104006				*****TEST 17 - ERROR 6*****
2238						;R3 SHOULD BE ZERO
2239	006076	026704	172552			;CHECK R4 UNCHANGED
2240	006102	001401				;BR IF OK
2241	006104	104007				*****TEST 17 - ERROR 7*****
2242						;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2243	006106	026705	172544			;CHECK R5 UNCHANGED
2244	006112	001401				;BR IF OK
2245	006114	104010				*****TEST 17 - ERROR 10*****
2246						;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2247	006116					
2248	006116	012700	017034			;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2249	006122	105720				;TEST CONTENTS OF BOUNDARY
2250	006124	001401				;BR, IF STILL ZERO
2251	006126	104011				*****TEST 17 - ERROR 11*****
2252						;LOWER BOUNDARY OF DESTINATION CHANGED
2253						; SHOULD STILL EQUAL ZERO
2254	006130					
2255	006130	016702	172514			;CALCULATE THE NUMBER OF FILL
2256	006134	166702	172504			; CHARACTERS THAT SHOULD APPEAR IN DEST.
2257	006140	010201				;TRANSFER BYTE COUNT IN R1
2258	006142	122067	172506			;CHECK CHARACTERS IN DESTINATION
2259	006146	001401				;BR IF OK
2260	006150	104012				*****TEST 17 - ERROR 12*****
2261						;COMPARE ERROR IN DESTINATION
2262						;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2263						
2264	006152	005301				;DECREMENT BYTE COUNT
2265	006154	001372				;BR, IF NOT FINISHED CHECKING

M04

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 51
 DVKAIA.P11 22-DEC-76 10:47 T17

TEST INTERRUPTABILITY OF "MOVRC" INSTRUCTION

SEQ 0052

2266	006156	016702	172464		MOV	SRCAD,R2		:POINT R2 TO SOURCE STRING
2267	006162	016701	172456		MOV	SRCLN,R1		:STORE TRANSFER BYTE COUNT IN R1
2268	006166	122022		TE13:	CMPB	(R0)+,(R2)+		:CHECK CHARACTERS IN DESTINATION
2269	006170	001401			BEQ	.+4		:BR IF OK
2270	006172	104013			ERROR	13		:*****TEST 17 - ERROR 13*****
2271								:COMPARE ERROR IN DESTINATION
2272								:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2273								
2274	006174	005301			DEC	R1		:DECREMENT BYTE COUNT
2275	006176	001373			BNE	TE13		:BR IF NOT FINISHED CHECKING
2276	006200	105720			TSTB	(R0)+		:TEST CONTENTS OF DEST. UPPER BOUNDARY
2277	006202	001401			BEQ	ENDT17		:BR IF STILL ZERO
2278	006204	104014			ERROR	14		:*****TEST 17 - ERROR 14*****
2279								:UPPER BOUNDARY OF DEST. CHANGED
2280								: SHOULD STILL EQUAL ZERO
2281	006206			ENDT17:				
2282	006206	106427	000200	40\$:	MTPS	#200		:RESTORE PSW TO PRIORITY 7
2283	006212	016777	172450	172444	MOV	TPSW,ATVECT		:RESTORE TRAP CATCHER


```

2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296 006220 000004
2297 006222 004567 006764
2298 006226 000000
2299 006230 177777
2300 006232 000000
2301 006234 177777
2302 006236 000377
2303 006240 004567 007042
2304 006244 000204
2305 006246 004767 006674
2306 006252 000277
2307 006254 000244
2308
2309 006256 076044
2310
2311 006260 106767 172406
2312 006264 042767 177400 172400
2313 006272 023767 000674 172372
2314 006300 001401
2315 006302 104001
2316
2317
2318
2319 006304
2320 006304 023706 000676
2321 006310 001403
2322 006312 010637 000700
2323 006316 104002
2324
2325
2326
2327 006320
2328 006320 026700 172320
2329 006324 001401
2330 006326 104003
2331
2332 006330 026701 172312
2333 006334 001401
2334 006336 104004
2335
2336 006340 026702 172304
2337 006344 001401
2338 006346 104005
2339

```

```

*****
*TEST 20      TEST "CMPC" WITH SOURCE1 & SOURCE2 LENGTHS = 0
*****
*THIS TEST VERIFIES THAT "COMPARE CHARACTER" INDICATES EQUAL
*STRINGS WITH SOURCE LENGTHS EQUAL TO ZERO.
*THE RESULT IS R0-->R3 ARE UNCHANGED, AND ALL CONDITION
*CODES CLEAR EXCEPT Z=1.
*****
↑ST20:  SCOPE
        JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
                                ;SOURCE1 LENGTH
                                ;SOURCE1 ADDRESS
                                ;SOURCE2 LENGTH
                                ;SOURCE2 ADDRESS
                                ;FILL CHARACTER
        JSR      R5,XPSW
        .WORD    204
        JSR      PC,GENR      ;SET UP GENERAL REGISTERS
        SCC                      ;SET ALL CONDITION CODES, EXCEPT
        CLZ                      ;CLEAR Z
                                ;EXECUTE "CMPC" INSTRUCTION
        CMPC
        MFPS      CCODES      ;STORE THE PSW
        BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
        CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$          ;BR, IF EQUAL
        ERROR    1            ;*****TEST 20 - ERROR 1*****
                                ;PSW ERROR
                                ;EXPECTED PSW IS STORED AT "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$:
        CMP      SRC1N,R0      ;CHECK R0 UNCHANGED
        BEQ      1$
        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

```

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 53
 DVKAIA.P11 22-DEC-76 10:47 T20

TEST "CMPC" WITH SOURCE1 & SOURCE2 LENGTHS = 0

SEQ 0054

2340	006350	026703	172276	3\$:	CMP	DSTAD,R3	;CHECK R3 UNCHANGED
2341	006354	001401			BEQ	4\$	
2342	006356	104006			ERROR	6	*****TEST 20 - ERROR 6*****
2343							;SRC2 ADDRESS ERROR
2344	006360			4\$:			
2345	006360	026704	172270		CMP	FILL,R4	;CHECK R4 UNCHANGED
2346	006364	001401			BEQ	66\$	;BR IF OK
2347	006366	104007			ERROR	7	*****TEST 20 - ERROR 7*****
2348							;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2349	006370	026705	172262	66\$:	CMP	TABLE,R5	;CHECK R5 UNCHANGED
2350	006374	001401			BEQ	67\$	;BR IF OK
2351	006376	104010			ERROR	10	*****TEST 20 - ERROR 10*****
2352							;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2353	006400			67\$:			


```

2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367 006400 000004
2368
2369 006402 004567 006604 JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
2370 006406 000000 0 ;SOURCE1 LENGTH
2371 006410 177777 NXM ;SOURCE1 ADDRESS
2372 006412 000004 4 ;SOURCE2 LENGTH
2373 006414 017034 BUF2 ;SOURCE2 ADDRESS
2374 006416 000377 377 ;FILL CHARACTER
2375 006420 012700 017034 MOV #BUF2,R0 ;GENERATE SOURCE2 STRING STARTING
2376 006424 012720 177777 MOV #177777,(R0)+ ;WITH TWO FILL BYTES
2377 006430 005020 (R0)+ ;THEN TWO NON-FILL BYTES
2378 006432 004567 006650 JSR R5,XPSW ;STORE EXPECTED PSW VALUE
2379 006436 000210 .WORD 210
2380 006440 004767 006502 JSR PC,GENR ;SET UP GENERAL REGISTERS
2381 006444 000277 SCC ;SET ALL CONDITION CODES
2382
2383 006446 076044 CMPC ;EXECUTE "CMPC" INSTRUCTION
2384
2385 006450 106767 172216 MFPS CCODES ;STORE THE PSW
2386 006454 042767 177400 172210 BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
2387 006462 023767 000674 172202 CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
2388 006470 001401 64$ BR, IF EQUAL
2389 006472 104001 ERROR 1 ;*****TEST 21 - ERROR 1*****
2390
2391
2392
2393 006474 64$:
2394 006474 023706 000676 CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2395 006500 001403 65$ BR IF OK
2396 006502 010637 000700 MOV SP,#BADR6 ;STORE BAD SP VALUE
2397 006506 104002 ERROR 2 ;*****TEST 21 - ERROR 2*****
2398
2399
2400
2401 006510 55$:
2402 006510 005700 TST R0 ;CHECK R0=0
2403 006512 001401 1$ BEQ 1$
2404 006514 104003 ERROR 3 ;*****TEST 21 - ERROR 3*****
2405
2406 006516 026701 172124 1$: CMP SRCAD,R1 ;SRC1 LENGTH ERROR
2407 006522 001401 2$ BEQ 2$ ;CHECK R1 UNCHANGED
2408 006524 104004 ERROR 4 ;*****TEST 21 - ERROR 4*****
2409
2409

```

D05

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 55
 DVKATA.P11 22-DEC-76 10:47 T21

TEST "CMPC" WITH SOURCE1 LENGTH=0 & A NON-FILL CHARACTER IN SRC2

SEQ 0056

2410	006526	022702	000002	2\$:	CMP	#2,R2	;CHECK R2=2
2411	006532	001401			BEG	3\$	
2412	006534	104005			ERROR	5	*****TEST 21 - ERROR 5*****
2413							;SRC2 LENGTH ERROR
2414	006536	022703	017036	3\$:	CMP	#BUF2+2,R3	;CHECK R3=BUF2+2
2415	006542	001401			BEG	4\$	
2416	006544	104006			ERROR	6	*****TEST 21 - ERROR 6*****
2417							;SRC2 ADDRESS ERROR
2418	006546			4\$:			
2419	006546	026704	172102		CMP	FILL,R4	;CHECK R4 UNCHANGED
2420	006552	001401			BEG	66\$	;BR IF OK
2421	006554	104007			ERROR	7	*****TEST 21 - ERROR 7*****
2422							;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2423	006556	026705	172074	66\$:	CMP	TABLE,R5	;CHECK R5 UNCHANGED
2424	006562	001401			BEG	67\$	;BR IF OK
2425	006564	104010			ERROR	10	*****TEST 21 - ERROR 10*****
2426							;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2427	006566			67\$:			

2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483

006566 000004
006570 004567 006416
006574 000007
006576 016434
006600 000012
006602 017034
006604 000377
006606 004767 006426
006612 004567 006450
006616 000012
006620 017034
006622 105037 017041
006626 004567 006454
006632 000210
006634 004767 006306
006640 000277

006642 076044

006644 106767 172022
006650 042767 177400 172014
006656 023767 000674 172006
006664 001401
006666 104001

006670
006670 023706 000676
006674 001403
006676 010637 000700
006702 104002

006704
006704 026704 171744
006710 001401
006712 104003

006714 026704 171736
006720 001401
006722 104004

```

*****
*TEST 22      TEST "CMPC" WITH S1L .LT. S2L AND SRC1.GT.SRC2, NON-COMPARE IN SOURCE
*****
*THIS TEST VERIFIES THAT "CMPC" RETURNS THE CORRECT SUBSTRING IDENTIFIERS
*WHEN SOURCE1 STRING AND SOURCE2 STRING DO NOT COMPARE.
*THE RESULT IS R0 EQUALS SRC1 SUBSTRING LENGTH, R1 EQUALS
*SRC1 SUBSTRING POINTER, R2 EQUALS SRC2 SUBSTRING LENGTH, R3 EQUALS
*SRC2 SUBSTRING POINTER, AND ALL CONDITION CODES CLEAR EXCEPT N=1.
*****
*ST22:  SCOPE
        JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
        7          ;SOURCE1 LENGTH
        BUF1       ;SOURCE1 ADDRESS
        12         ;SOURCE2 LENGTH
        BUF2       ;SOURCE2 ADDRESS
        377        ;FILL CHARACTER
        JSR      PC,CLODST       ;CLEAR SOURCE2 AREA
        JSR      R5,GENSRC       ;GENERATE SOURCE2 STRING
        .WORD     12
        .WORD     BUF2
        CLRB      2*BUF2+5       ;CREATE A NON-COMPARE CHARACTER IN SRC2 STRING
        JSR      R5,XPSW         ;STORE EXPECTED PSW VALUE
        .WORD     210
        JSR      PC,GENR        ;SET UP GENERAL REGISTERS
        SCC                ;SET ALL CONDITION CODES
        CMPC                ;EXECUTE "CMPC" INSTRUCTION
        MFPS      CCODES        ;STORE THE PSW
        BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
        CMP      2*EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$            ;BR, IF EQUAL
        ERROR    1              ;*****TEST 22 - ERROR 1*****
        PSW ERROR
        EXPECTED PSW IS STORED AT "EXPPSW"
        ACTUAL PSW IS STORED AT "CCODES"
        64$:
        CMP      2*SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
        BEQ      65$            ;BR IF OK
        MOV      SP,2*BADR6     ;STORE BAD SP VALUE
        ERROR    2              ;*****TEST 22 - 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 22 - 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 22 - ERROR 4*****
        R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 57
 DVKATA.P11 22-DEC-76 10:47 T22

TEST "CMPC" WITH S1L .LT. S2L AND SRC1.GT.SRC2, NON-COMPARE IN SOURCE

SEQ 0032

2484	006724			67%:			
2485	006724	005067	171754	1%:	CLR	TEMP	
2486	006730	016704	171712		MOV	SRCAD,R4	:POINT R4 TO SOURCE1 ADDRESS
2487	006734	016705	171712		MOV	DSTAD,R5	:POINT R5 TO SOURCE2 ADDRESS
2488	006740	121415		2%:	CMPB	(R4),(R5)	:COMPARE SOURCES
2489	006742	001004			BNE	3%	:BR IF NOT EQUAL
2490	006744	005267	171734		INC	TEMP	
2491	006750	122425			CMPB	(R4)+,(R5)+	:POINT STRINGS TO NEXT BYTES
2492	006752	000772			BR	2%	
2493	006754			3%:			:CHECK ADDRESS & LENGTH DESCRIPTORS
2494	006754	020401			CMP	R4,R1	:CHECK R1, SOURCE1 ADDRESS
2495	006756	001401			BEQ	4%	:BR IF OK
2496	006760	104005			ERROR	5	:*****TEST 22 - ERROR 5*****
2497							:SOURCE1 ADDRESS ERROR
2498							:R4 CONTAINS THE EXPECTED ADDRESS POINTER
2499							:R1 CONTAINS THE BAD VALUE
2500	006762	020503		4%:	CMP	R5,R3	:CHECK R3, SOURCE2 ADDRESS
2501	006764	001401			BEQ	5%	:BR IF OK
2502	006766	104006			ERROR	6	:*****TEST 22 - ERROR 6*****
2503							:SOURCE2 ADDRESS ERROR
2504							:R5 CONTAINS THE EXPECTED ADDRESS POINTER
2505							:R3 CONTAINS THE BAD VALUE
2506	006770	016704	171650	5%:	MOV	SRCLEN,R4	:CALCULATE LENGTH OF SRC1 REMAINDER
2507	006774	166704	171704		SUB	TEMP,R4	
2508	007000	020400			CMP	R4,R0	:CHECK R0, SOURCE1 LENGTH
2509	007002	001401			BEQ	6%	:BR IF OK
2510	007004	104007			ERROR	7	:*****TEST 22 - ERROR 7*****
2511							:SOURCE1 LENGTH ERROR
2512							:R4 CONTAINS THE EXPECTED LENGTH DESCRIPTOR
2513							:R0 CONTAINS THE BAD VALUE
2514	007006	016704	171636	6%:	MOV	DSTLEN,R4	:CALCULATE LENGTH OF SRC2 REMAINDER
2515	007012	166704	171666		SUB	TEMP,R4	
2516	007016	020402			CMP	R4,R2	:CHECK R2, SOURCE2 LENGTH
2517	007020	001401			BEQ	7%	:BR IF OK
2518	007022	104010			ERROR	10	:*****TEST 22 - ERROR 10*****
2519							:SOURCE2 LENGTH ERROR
2520							:R4 CONTAINS THE EXPECTED LENGTH DESCRIPTOR
2521							:R2 CONTAINS THE BAD VALUE
2522	007024	000240		7%:	NOP		


```

2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534 007026 000004
2535 007030 004567 006156
2536 007034 000007
2537 007036 016434
2538 007040 000007
2539 007042 016434
2540 007044 000377
2541 007046 004567 006234
2542 007052 000204
2543 007054 004767 006066
2544 007060 000277
2545 007062 000244
2546
2547 007064 076044
2548
2549 007066 106767 171600
2550 007072 042767 177400 171572
2551 007100 023767 000674 171564
2552 007106 001401
2553 007110 104001
2554
2555
2556
2557 007112
2558 007112 023706 000676
2559 007116 001403
2560 007120 010637 000700
2561 007124 104002
2562
2563
2564
2565 007126
2566 007126 105700
2567 007130 001401
2568 007132 104003
2569
2570 007134 105702
2571 007136 001401
2572 007140 104004
2573
2574 007142 016700 171500
2575 007146 066700 171472
2576 007152 020001
2577 007154 001401
2578 007156 104005

*****
*TEST 23 TEST "CMPC" FOR S1L=S2L, SOURCE1=SOURCE2 STRINGS
*****
*PROPER TERMINATION OF THIS INSTRUCTION TEST IS THE
*INDICATION THAT SOURCE1 STRING EQUALS SOURCE2 STRING:
*RO & R2 EQUAL ZERO, R1 & R3 EQUAL THE ADDRESS+1 OF THE
*LSB OF THE SOURCE1 & SOURCE2 STRINGS RESPECTIVELY,
*AND ALL CONDITION CODES CLEAR, EXCEPT Z=1
*****
TST23: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      7          ;SOURCE1 LENGTH
      BUF1      ;SOURCE1 ADDRESS
      7          ;SOURCE2 LENGTH
      BUF1      ;SOURCE2 ADDRESS
      377       ;FILL CHARACTER
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD     204
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      SCC       ;SET ALL CONDITION CODES, EXCEPT
      CLZ       ;CLEAR "Z"
               ;EXECUTE "CMPC" INSTRUCTION
               ;CHECK RESULTS
               ;STORE THE PSW
      MFPS      CCODES      ;CLEAR ALL UNUSED BITS
      BIC      #177400,CCODES
      CMP      3#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
               *****TEST 23 - ERROR 1*****
               PSW ERROR
               EXPECTED PSW IS STORED AT "EXPPSW"
               ACTUAL PSW IS STORED AT "CCODES"
      64$:      CMP      3#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;BR IF OK
      MOV      SP,3#BADR6 ;STORE BAD SP VALUE
      ERROR    2          *****TEST 23 - ERROR 2*****
               STACK POINTER NOT RESTORED BY INSTRUCTION
               EXPECTED VALUE OF SP IS STORED AT "SAVR6"
               ERRONEOUS VALUE IS AT "BADR6"
      65$:      TSTB     R0          ;CHECK R0
      BEQ      1$          ;BR, IF ZERO
      ERROR    3          *****TEST 23 - ERROR 3*****
               R0 SHOULD BE ZERO
      1$:      TSTB     R2          ;CHECK R2
      BEQ      2$          ;BR, IF ZERO
      ERROR    4          *****TEST 23 - ERROR 4*****
               R2 SHOULD BE ZERO
      2$:      MOV      SRCAD,R0    ;CALCULATE ADDRESS+1 OF LSB
      ADD      SRCLN,R0
      CMP      R0,R1
      BEQ      3$
      ERROR    5          ;CHECK R1
               ;BR, IF EQUAL TO ADDRESS+1 OF LSB
               *****TEST 23 - ERROR 5*****

```

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 59
 DVKAIA.P11 22-DEC-76 10:47 T23 TEST "CMPC" FOR S1L=S2L, SOURCE1=SOURCE2 STRINGS

Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419
---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 60
 DVKATA.F11 22-DEC-76 10:47 T23 TEST "CMPC" FOR S1L=S2L, SOURCE1=SOURCE2 STRINGS

SEQ 0061

```

26000 007206 000004
26001
26002
26003
26004
26005
26006
26007 007210 004567 005776 JSR RS,PREP ;SET UP INSTRUCTION ARGUMENTS
26008 0 214 000004 4 ;SOURCE1 LENGTH
26009 0 216 017034 BUF2 ;SOURCE1 ADDRESS
26010 007220 000000 0 ;SOURCE2 LENGTH
26011 007222 177777 NXM ;SOURCE2 ADDRESS
26012 007224 000377 377 ;FILL CHARACTER
26013 007226 012700 017034 MOV #BUF2,RO ;GENERATE SOURCE1 STRING WITH
26014 007232 012710 177777 MOV #177777,(RO) ;ALL FILLS
26015 007236 012010 MOV (RO)+(RO)
26016 007240 004567 006042 JSR RS,XPSW ;STORE EXPECTED PSW VALUE
26017 007244 000204 .WORD 204
26018 007246 004767 005674 JSR PC,GENR ;SET UP GENERAL REGISTERS
26019 007252 000277 SCC ;SET ALL CONDITION CODES, EXCEPT
26020 007254 000244 CLZ ;CLEAR Z
26021
26022 007256 076044 CMPC ;EXECUTE "CMPC" INSTRUCTION
26023
26024 007260 106767 171406 MFPS CCODES ;STORE THE PSW
26025 007264 042767 177400 171400 BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
26026 007272 023767 000674 171372 CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
26027 007300 001401 64$ BR, IF EQUAL
26028 007302 104001 1 *****TEST 24 - ERROR 1*****
26029 ;PSW ERROR
26030 ;EXPECTED PSW IS STORED AT "EXPPSW"
26031 ;ACTUAL PSW IS STORED AT "CCODES"
26032 007304 64$: 023706 000676 CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
26033 007304 001403 65$ BR IF OK
26034 007310 010637 000700 MOV SP,#BADR6 ;STORE BAD SP VALUE
26035 007312 104002 2 *****TEST 24 - ERROR 2*****
26036 ;STACK POINTER NOT RESTORED BY INSTRUCTION
26037 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
26038 ;ERRONEOUS VALUE IS AT "BADR6"
26039 007320 65$: 005700 TST RO ;CHECK RO=0
26040 007320 001401 1$ BEQ 1$
26041 007322 104003 3 *****TEST 24 - ERROR 3*****
26042 ;SOURCE1 LENGTH ERROR
26043 ;RO SHOULD BE ZERO
26044 ;CALCULATE ADDRESS OF LSB+1
26045 007326 016700 171314 1$: MOV SRCAD,RO
26046 007332 066700 171306 ADD SRCLN,RO
26047 007336 020001 CMP RO,R1 ;CHECK R1, SOURCE1 ADDRESS
26048 007340 001401 2$ BEQ 2$ ;BR IF OK
26049 007342 104004 4 *****TEST 24 - ERROR 4*****
26050

```

ADDRESS	INSTR	OP1	OP2	OP3	OP4	COMMENT
007344	005702	2\$:	TST	R2		:SOURCE1 ADDRESS ERROR
007346	001401		BEQ	3\$		:R0 CONTAINS THE EXPECTED ADDRESS POINTER
007350	104005		ERROR	5		:R1 CONTAINS THE BAD VALUE
						:CHECK R2=0
						:BR IF OK
						:*****TEST 24 - ERROR 5*****
						:SOURCE2 LENGTH ERROR
						:R2 SHOULD BE ZERO
007352	016702	3\$:	MOV	DSTAD,R2		:CALCULATE ADDRESS OF LSB+1
007356	066702		ADD	DSTLN,R2		
007362	020203		CMP	R2,R3		:CHECK R3,SOURCE2 ADDRESS
007364	001401		BEQ	4\$		:BR IF OK
007366	104006		ERROR	6		:*****TEST 24 - ERROR 6*****
						:SOURCE2 ADDRESS ERROR
						:R2 CONTAINS THE EXPECTED ADDRESS POINTER
						:R3 CONTAINS THE BAD VALUE
007370		4\$:				
007370	026704		CMP	FILL,R4		:CHECK R4 UNCHANGED
007374	001401		BEQ	6\$		:BR IF OK
007376	104007		ERROR	7		:*****TEST 24 - ERROR 7*****
						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
007400	026705	66\$:	CMP	TABLE,R5		:CHECK R5 UNCHANGED
007404	001401		BEQ	67\$		:BR IF OK
007406	104010		ERROR	10		:*****TEST 24 - ERROR 10*****
						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
007410		67\$:				


```

2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690 007410 000004
2691 007412 004567 005574
2692 007416 000004
2693 007420 017034
2694 007422 000000
2695 007424 177777
2696 007426 000377
2697 007430 012700 017034
2698 007434 012720 177777
2699 007440 005020
2700 007442 004567 005640
2701 007446 000201
2702 007450 004767 005472
2703 007454 000277
2704 007456 000250
2705
2706 007460 076044
2707
2708 007462 106767 171204
2709 007466 042767 177400 171176
2710 007474 023767 000674 171170
2711 007502 001401
2712 007504 104001
2713
2714
2715
2716 007506
2717 007506 023706 000676
2718 007512 001403
2719 007514 010637 000700
2720 007520 104002
2721
2722
2723
2724 007522
2725 007522 022700 000002
2726 007526 001401
2727 007530 104003
2728
2729 007532 022701 017036
2730 007536 001401
2731 007540 104004
2732
2733 007542 005702
2734 007544 001401

*****
*TEST 25 TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO
*****
*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
*****
†ST25: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
                        4      ;SOURCE1 LENGTH
                        BUF2    ;SOURCE1 ADDRESS
                        0      ;SOURCE2 LENGTH
                        NXM     ;SOURCE2 ADDRESS
                        377     ;FILL CHARACTER
                        #BUF2,R0 ;GENERATE SOURCE1 STRING STARTING
                        #177777,(R0)+ ;WITH TWO FILL BYTES
                        (R0)+    ;THEN TWO "ZERO" BYTES
                        R5,XPSW  ;STORE EXPECTED PSW VALUE
                        201
                        PC,GENR ;SET UP GENERAL REGISTERS
                        SCC      ;SET ALL CONDITION CODES, EXCEPT
                        CLN      ;CLEAR N=1
                        ;EXECUTE "CMPC" INSTRUCTION

                        ;CHECK RESULTS
                        MFPS     ;STORE THE PSW
                        CCODES   ;CLEAR ALL UNUSED BITS
                        BIC      ;#177400,CCODES
                        CMP      ;@#EXPPSW,CCODES
                        BEQ      ;64$
                        ERROR    ;1
                        ;*****TEST 25 - 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 25 - ERROR 2*****
      ;STACK POINTER NOT RESTORED BY INSTRUCTION
      ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ;ERRONEOUS VALUE IS AT "BADR6"

65$:
      CMP      #2,R0
      BEQ      1$
      ERROR    3
      ;*****TEST 25 - ERROR 3*****
      ;SRC1 LENGTH ERROR
      ;CHECK R1=BUF2+2
      1$:
      CMP      #BUF2+2,R1
      BEQ      2$
      ERROR    4
      ;*****TEST 25 - ERROR 4*****
      ;SRC1 ADDRESS ERROR
      ;CHECK R2=0
      2$:
      TST      R2
      BEQ      3$

```

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 63

DVKAIA.P11 22-DEC-76 10:47

T25

TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO

SEQ 0064

2735	007546	104005		ERROR	5	*****TEST 25 - ERROR 5*****
2736						:SRC2 LENGTH ERROR
2737	007550	026703	171076	3\$:	CMP	DSTAD,R3
2738	007554	001401			BEQ	4\$
2739	007556	104006			ERROR	6
2740						*****TEST 25 - ERROR 6*****
2741	007560			4\$:		:SRC2 ADDRESS ERROR
2742	007560	026704	171070		CMP	FILL,R4
2743	007564	001401			BEQ	66\$
2744	007566	104007			ERROR	7
2745						*****TEST 25 - ERROR 7*****
2746	007570	026705	171062	66\$:	CMP	TABLE,R5
2747	007574	001401			BEQ	67\$
2748	007576	104010			ERROR	10
2749						*****TEST 25 - ERROR 10*****
2750	007600			67\$:		:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

M05

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 64
 DVKAIA.P11 22-DEC-76 10:47 T25 TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO

SEQ 0065

```

2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764 007600 000004
2765 007602 004567 005404
2766 007606 000007
2767 007610 017034
2768 007612 000003
2769 007614 016434
2770 007616 000377
2771 007620 004567 005442
2772 007624 000007
2773 007626 017034
2774 007630 105037 017036
2775 007634 004567 005446
2776 007640 000211
2777 007642 004767 005300
2778 007646 000251
2779 007650 000266
2780
2781 007652 076044
2782
2783 007654 106767 171012
2784 007660 042767 177400 171004
2785 007666 023767 000674 170776
2786 007674 001401
2787 007676 104001
2788
2789
2790
2791 007700
2792 007700 023706 000676
2793 007704 001403
2794 007706 010637 000700
2795 007712 104002
2796
2797
2798
2799 007714
2800 007714 026704 170734
2801 007720 001401
2802 007722 104003
2803
2804 007724 026705 170726
2805 007730 001401
2806 007732 104004

;*****
;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
;*****
†ST26: 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
      MFPS CCODES ;CHECK RESULT
      BIC #177400,CCODES ;STORE THE PSW
      CMP @#EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
      BEQ 64$ ;CHECK PSW AGAINST EXPECTED VALUE
      ERROR 1 ;BR, IF EQUAL
      ;*****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*****

```

Address	Hex	Dec	Label	Instruction	Comment
2807	007734				
2808	007734		67\$:		;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2809	007734		1\$:		;CHECK OTHER REGISTERS
2810	007734	005067	170744	CLR TEMP	
2811	007740	016704	170702	MOV SRCAD,R4	;POINT R4 TO SOURCE1 ADDRESS
2812	007744	016705	170702	MOV DSTAD,R5	;POINT R5 TO SOURCE2 ADDRESS
2813	007750	121415		2\$: CMPB (R4),(R5)	;COMPARE SOURCES
2814	007752	001004		BNE 3\$	;BR, IF NOT EQUAL
2815	007754	005267	170724	INC TEMP	
2816	007760	122425		CMPB (R4)+,(R5)+	;POINT STRINGS TO NEXT BYTES
2817	007762	000772		BR 2\$	
2818	007764			3\$:	;CHECK ADDRESS REGISTERS
2819	007764	020401		CMP R4,R1	;CHECK R1 (SOURCE1 STRING)
2820	007766	001401		BEQ 4\$	
2821	007770	104005		ERROR 5	;*****TEST 26 - ERROR 5*****
2822					;SRC1 ADDRESS ERROR
2823	007772	020503		4\$: CMP R5,R3	;CHECK R3 (SOURCE2 STRING)
2824	007774	001401		BEQ 5\$	;BR IF EQUAL
2825	007776	104006		ERROR 6	;*****TEST 26 - ERROR 6*****
2826					;SRC2 ADDRESS ERROR
2827	010000	016704	170640	5\$: MOV SRCLN,R4	;CALCULATE LENGTH OF SRC1 REMAINDER
2828	010004	166704	170674	SUB TEMP,R4	
2829	010010	020400		CMP R4,R0	;CHECK R0 (SRC1 LENGTH)
2830	010012	001401		BEQ 6\$	;BR, IF EQUAL
2831	010014	104007		ERROR 7	;*****TEST 26 - ERROR 7*****
2832					;SRC1 STRING LENGTH ERROR
2833	010016	016704	170626	6\$: MOV DSTLN,R4	;CALCULATE LENGTH OF SRC2 REMAINDER
2834	010022	166704	170656	SUB TEMP,R4	
2835	010026	020402		CMP R4,R2	;CHECK R2 (SRC2 LENGTH)
2836	010030	001401		BEQ 40\$	;BR, IF EQUAL
2837	010032	104010		ERROR 10	;*****TEST 26 - ERROR 10*****
2838					;SRC2 STRING LENGTH ERROR
2839	010034			40\$:	


```

2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850 010034 000004
2851 010036 004567 005150
2852 010042 000010
2853 010044 016434
2854 010046 000012
2855 010050 017034
2856 010052 000377
2857 010054 004767 005160
2858 010060 004567 005202
2859 010064 000010
2860 010066 017034
2861 010070 012721 177777
2862 010074 004567 005206
2863 010100 000204
2864 010102 004767 005040
2865 010106 000277
2866 010110 000244
2867
2868 010112 076044
2869
2870 010114 106767 170552
2871 010120 042767 177400 170544
2872 010126 023767 000674 170536
2873 010134 001401
2874 010136 104001
2875
2876
2877
2878 010140
2879 010140 023706 000676
2880 010144 001403
2881 010146 010637 000700
2882 010152 104002
2883
2884
2885
2886 010154
2887 010154 005700
2888 010156 001401
2889 010160 104003
2890
2891 010162 016700 170460
2892 010166 066700 170452
2893 010172 020001
2894 010174 001401
2895 010176 104004

;*****
;TEST 27 TEST "CMPC" WITH SOURCE1 LENGTH .GT. SOURCE2 LENGTH & SOURCE1=SOURCE2
;*****
;THIS TEST VERIFIES THAT "CMPC" INDICATES EQUAL STRINGS WHEN THE
;STRING LENGTHS ARE NOT EQUAL BUT THE EXCESS OF THE LONGER
;STRING CONTAINS ALL FILLS.
;*****
TST27: SCOPE
        JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
        JSR      10              ;SOURCE1 LENGTH
        JSR      BUF1            ;SOURCE1 ADDRESS
        JSR      12              ;SOURCE2 LENGTH
        JSR      BUF2            ;SOURCE2 ADDRESS
        JSR      377             ;FILL CHARACTER
        JSR      PC,CLDST         ;CLEAR SOURCE2 AREA
        JSR      R5,GENSRC        ;GENERATE SOURCE2 STRING
        .WORD    10              ;MAKE FIRST 10 BYTES IDENTICAL
        .WORD    BUF2            ;TO SOURCE1 STRING AND THE
        MOV      #177777,(R1)+    ;LAST TWO BYTES EQUAL TO FILL CHARACTER
        JSR      R5,XPSW          ;STORE EXPECTED PSW VALUE
        .WORD    204
        JSR      PC,GENR         ;SET UP GENERAL REGISTERS
        SCC      CLZ              ;SET ALL CONDITION CODES, EXCEPT
        CLZ                      ;CLEAR Z
        CMPC                     ;EXECUTE "CMPC" INSTRUCTION
        MFPS      CCODES          ;STORE THE PSW
        BIC      #177400,CCODES   ;CLEAR ALL UNUSED BITS
        CMP      @EXPPSW,CCODES   ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$              ;BR, IF EQUAL
        ERROR    1                ;*****TEST 27 - ERROR 1*****
        PSW ERROR
        EXPECTED PSW IS STORED AT "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 27 - ERROR 2*****
        STACK POINTER NOT RESTORED BY INSTRUCTION
        EXPECTED VALUE OF SP IS STORED AT "SAVR6"
        ERRONEOUS VALUE IS AT "BADR6"
        65$:
        TST      R0                ;CHECK R0=0
        BEQ      1$              ;*****TEST 27 - ERROR 3*****
        ERROR    3                ;SOURCE1 LENGTH ERROR
        MOV      SRCAD,R0          ;CALCULATE ADDRESS OF LSB+1
        ADD      SRCLN,R0
        CMP      R0,R1            ;CHECK R1, SOURCE1 ADDRESS
        BEQ      2$              ;*****TEST 27 - ERROR 4*****
        ERROR    4

```

2896						;SOURCE1 ADDRESS ERROR
2897	010200	005702		2\$:	TST R2	;CHECK R2=0
2898	010202	0C1401			BEG 3\$	
2899	010204	104005			ERROR 5	;*****TEST 27 - ERROR 5*****
2900						;SOURCE2 LENGTH ERROR
2901	010206	016702	170440	3\$:	MOV DSTAD,R2	;CALCULATE ADDRESS OF LSB+1
2902	010212	066702	170432		ADD DSTLN,R2	
2903	010216	020203			CMP R2,R3	;CHECK R3,SOURCE2 ADDRESS
2904	010220	001401			BEG 4\$	
2905	010222	104006			ERROR 6	;*****TEST 27 - ERROR 6*****
2906						;SOURCE2 ADDRESS ERROR
2907	010224			4\$:		
2908	010224	026704	170424		CMP FILL,R4	;CHECK R4 UNCHANGED
2909	010230	001401			BEG 66\$	;BR IF OK
2910	010232	104007			ERROR 7	;*****TEST 27 - ERROR 7*****
2911						;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
2912	010234	026705	170416	66\$:	CMP TABLE,R5	;CHECK R5 UNCHANGED
2913	010240	001401			BEG 67\$	;BR IF OK
2914	010242	104010			ERROR 10	;*****TEST 27 - ERROR 10*****
2915						;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
2916	010244			67\$:		

[illegible]

```

2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963 010346 000004
2964 010350 105777 170164
2965 010354 100537
2966 010356 026767 170166 170274
2967 010364 001007
2968 010366 032767 000001 170212
2969 010374 001403
2970 010376 005767 170172
2971 010402 001124
2972 010404
2973 010404 004567 004602
2974 010410 000400
2975 010412 016434
2976 010414 000400
2977 010416 016434
2978 010420 000377
2979 010422 012767 010510 170240
2980 010430 012777 015356 170226
2981 010436 005077 170224
2982 010442 004767 004664
2983 010446 013777 000554 170206
2984 010454 004567 004626
2985 010460 000004
2986 010462 106427 000000
2987 010466 052777 000100 170164
2988 010474 004767 004446
2989 010500 010637 000676
2990 010504 000277
2991 010506 000244
2992
2993 010510 076044
2994
2995
2996 010512 106767 170154
2997 010516 032777 000100 170134
2998 010524 001363
2999 010526 042767 177400 170136
3000 010534 023767 000674 170130
3001 010542 001401
3002 010544 104001
3003
3004
3005
3006 010546

```

 *TEST 31 TEST INTERRUPTABILITY OF "CMPC" INSTRUCTION

 *THIS TEST INTERRUPTS THE EXECUTION OF THE "CMPC"
 *INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
 *INTERRUPT, VERIFIES THE RESULT. THE PROPER RESULT
 *IS FOR SOURCE1=SOURCE2.

TST31: SCOPE
 TSTB @SWR ;TEST BIT 7 OF SWR
 BMI TST32 ;SKIP TO NEXT TEST IF SET
 CMP \$TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
 BNE T31CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
 BIT @BITC, \$ENV ;IF YES, CHECK IF ON APT
 BEQ T31CONT ;BR IF NOT UNDER APT, AND DO THIS TEST
 TST \$PASS ;CHECK IF ON FIRST PASS
 BNE TST32 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T31CONT:
 JSR R5, PREP ;SET UP INSTRUCTION ARGUMENTS
 400 ;SOURCE1 LENGTH
 BUF1 ;SOURCE1 ADDRESS
 400 ;SOURCE2 LENGTH
 BUF1 ;SOURCE2 ADDRESS
 377 ;FILL CHARACTER
 MOV #CMC, PC ;STORE PC OF TEST INSTRUCTION
 MOV #INTRA, @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 04
 MTPS #0 ;SET PSW TO ALLOW INTERRUPTS
 BIS #100, @TCSR ;ENABLE TTY INTERRUPTS
 JSR PC, GENR ;SET UP GENERAL REGISTERS
 MOV SP, @#SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
 SCC ;SET ALL CONDITION CODES, EXCEPT
 CLZ ; CLEAR Z=1

REPCMC: JSR PC, GENR
 MOV SP, @#SAVR6

CMC: CMPC ;EXECUTE "CMPC" INSTRUCTION

MFPS CCODES ;CHECK RESULTS
 BIT #100, @TCSR ;STORE THE PSW
 BNE REPCMC ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
 BIC #177400, CCODES ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
 CMP @#EXPPSW, CCODES ;CLEAR ALL UNUSED BITS
 BEQ 64\$;CHECK PSW AGAINST EXPECTED VALUE
 ERROR 1 ;BR, IF EQUAL
 *****TEST 31 - ERROR 1*****
 PSW ERROR
 EXPECTED PSW IS STORED AT "EXPPSW"
 ACTUAL PSW IS STORED AT "CCODES"

64\$:

3007	010546	023706	000676		CMP	3#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
3008	010552	001403			BEG	65\$	:BR IF OK
3009	010554	010637	000700		MOV	SP,3#BADR6	:STORE BAD SP VALUE
3010	010560	104002			ERROR	2	:*****TEST 31 - ERROR 2*****
3011							:STACK POINTER NOT RESTORED BY INSTRUCTION
3012							:EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3013							:ERRONEOUS VALUE IS AT "BADR6"
3014	010562			65\$:			
3015	010562	105700			TSTB	R0	:CHECK R0
3016	010564	001401			BEG	1\$	:BR, IF ZERO
3017	010566	104003			ERROR	3	:*****TEST 31 - ERROR 3*****
3018							:R0 SHOULD BE ZERO
3019	010570	105702		1\$:	TSTB	R2	:CHECK R2
3020	010572	001401			BEG	2\$	:BR, IF ZERO
3021	010574	104004			ERROR	4	:*****TEST 31 - ERROR 4*****
3022							:R2 SHOULD BE ZERO
3023	010576	016700	170044	2\$:	MOV	SRCAD,R0	:CALCULATE ADDRESS+1 OF LSB
3024	010602	066700	170036		ADD	SRCLN,R0	
3025	010606	020001			CMP	R0,R1	:CHECK R1
3026	010610	001401			BEG	3\$	:BR, IF EQUAL TO ADDRESS+1 OF LSB
3027	010612	104005			ERROR	5	:*****TEST 31 - ERROR 5*****
3028							:R1 SHOULD EQUAL (SRCAD)+(SRCLN)
3029	010614	020003		3\$:	CMP	R0,R3	:CHECK R3
3030	010616	001401			BEG	40\$	:BR, IF EQUAL TO ADDRESS+1 OF LSB
3031	010620	104006			ERROR	6	:*****TEST 31 - ERROR 6*****
3032							:R3 SHOULD EQUAL (DSTAD)+(DSTLN)
3033	010622			40\$:			
3034	010622	026704	170026		CMP	FILL,R4	:CHECK R4 UNCHANGED
3035	010626	001401			BEG	66\$	:BR IF OK
3036	010630	104007			ERROR	7	:*****TEST 31 - ERROR 7*****
3037							:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3038	010632	026705	170020	66\$:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
3039	010636	001401			BEG	67\$	:BR IF OK
3040	010640	104010			ERROR	10	:*****TEST 31 - ERROR 10*****
3041							:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3042	010642			67\$:			
3043	010642	106427	000200		MTPS	#200	:RESTORE PSW TO PRIORITY 7
3044	010646	016777	170014 170010		MOV	TPSW,3TVECT	:RESTORE TRAP CATCHER

```

3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057 010654 000004
3058 010656 004567 004330
3059 010662 000010
3060 010664 016434
3061 010666 177777
3062 010670 177777
3063 010672 000004
3064 010674 004567 004406
3065 010700 000200
3066 010702 004767 004240
3067 010706 000277
3068
3069 010710 076040
3070
3071 010712 106767 167754
3072 010716 042767 177400 167746
3073 010724 023767 000674 167740
3074 010732 001401
3075 010734 104001
3076
3077
3078
3079 010736
3080 010736 023706 000676
3081 010742 001403
3082 010744 010637 000700
3083 010750 104002
3084
3085
3086
3087 010752
3088 010752 026704 167676
3089 010756 001401
3090 010760 104003
3091
3092 010762 026705 167670
3093 010766 001401
3094 010770 104004
3095
3096 010772
3097 010772 016703 167650
3098 010776 016702 167642
3099 011002 121367 167646
3100 011006 001403

```

```

*****
*TEST 32      TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE
*****
*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
*ADDRESS OF THE FIRST OCCURRENCE OF THE "CHARACTER"
*IN THE SOURCE STRING, R0 EQUALS REMAINING LENGTH
*OF THE SOURCE STRING, AND ALL CONDITION CODES CLEAR.
*****
TST32: SCOPE
      JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
      10      ;SOURCE LENGTH
      BUF1    ;SOURCE ADDRESS
      NXM     ;STORE NON-ZERO VALUES TO TEST R2 & R3
      NXM     ;UNAFFECTED BY DIS STRING INSTRUCTIONS
      4       ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
      JSR     RS,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD   200
      JSR     PC,GENR      ;SET UP GENERAL REGISTERS
      SCC     ;SET ALL CONDITION CODES
      LOCC    ;EXECUTE "LOCC"
      MFPS    CCODES      ;CHECK RESULTS
      BIC     #177400,CCODES ;STORE THE PSW
      CMP     3#EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
      BEQ     64$          ;CHECK PSW AGAINST EXPECTED VALUE
      ERROR   1            ;BR, IF EQUAL
      *****TEST 32 - ERROR 1*****
      PSW ERROR
      EXPECTED PSW IS STORED AT "EXPPSW"
      ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP     3#SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ     65$          ;BR IF OK
      MOV     SP,3#BADR6   ;STORE BAD SP VALUE
      ERROR   2            ;*****TEST 32 - 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 32 - 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 32 - ERROR 4*****
      R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
      MOV     SRCAD,R3     ;LOCATE "CHARACTER"
      MOV     SRCLN,R2
      68$:
      CMPB    (R3),FILL
      BEQ     69$

```


H06

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 72
 DVKAIA.P11 22-DEC-76 10:47 T32

TEST "LOOC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE

SEG 0073

01101	011010	105723	TSTB	(R3)+	:POINT TO NEXT BYTE
01102	011012	005302	DEC	R2	:DECREMENT BYTE COUNT
01103	011014	001372	BNE	69\$	:BR, IF NOT FINISHED
01104	011016	020200	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
01105	011020	001401	BEG	70\$	
01106	011022	104005	ERROR	5	:*****TEST 32 - ERROR 5*****
01107					:LOCATE LENGTH ERROR
01108	011024	020301	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
01109	011026	001401	BEG	71\$	
01110	011030	104006	ERROR	6	:*****TEST 32 - ERROR 6*****
01111					:LOCATE ADDRESS ERROR
01112	011032				

```

*****
*TEST 33      TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE, BL .GT. 100000
*****
*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
*ADDRESS OF THE FIRST OCCURRENCE OF THE "CHARACTER"
*IN THE SOURCE STRING, R0 EQUALS REMAINING LENGTH
*OF THE SOURCE STRING, AND ALL CONDITION CODES CLEAR
*EXCEPT N=1.
*****

```

```

*****
†ST33:  SCOPE
        JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
                                ;SOURCE LENGTH
                                ;SOURCE ADDRESS
                                ;STORE NON-ZERO VALUES TO TEST R2 & R3
                                ; UNAFFECTED BY DIS STRING INSTRUCTIONS
                                ;SEARCH (MASK FOR SCANC/SPANC) CHARACTER
                                ;STORE EXPECTED PSW VALUE
        JSR      R5,XPSW
        .WORD    210
        JSR      PC,GENR      ;SET UP GENERAL REGISTERS
        SCC                      ;SET ALL CONDITION CODES
                                ;EXECUTE "LOCC"
        LOCC
                                ;CHECK RESULTS
                                ;STORE THE PSW
        MFPS      CCODES      ;CLEAR ALL UNUSED BITS
        BIC      #177400,CCODES
        CMP      2#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$          ;BR, IF EQUAL
        ERROR    1
                                ;*****TEST 33 - ERROR 1*****
                                ;PSW ERROR
                                ;EXPECTED PSW IS STORED AT "EXPPSW"
                                ;ACTUAL PSW IS STORED AT "CCODES"
        64$:
        CMP      2#SAVR6,SP    ;VERIFY STACK POINTER IS RESTORED
        BEQ      65$          ;BR IF OK
        MOV      SP,2#BADR6    ;STORE BAD SP VALUE
        ERROR    2
                                ;*****TEST 33 - 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 33 - ERROR 3*****
                                ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
                                ;CHECK R5 UNCHANGED
        66$:
        CMP      TABLE,R5    ;CHECK R5 UNCHANGED
        BEQ      67$          ;BR IF OK
        ERROR    4
                                ;*****TEST 33 - ERROR 4*****
                                ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
        67$:
        MOV      SRCAD,R3      ;LOCATE "CHARACTER"
        MOV      SRCLN,R2
        68$:
        CMPB     (R3),FILL
        BEQ      69$

```

```

011032 000004
011034 004567 004152
011040 100010
011042 016434
011044 177777
011046 177777
011050 000004
011052 004567 004230
011056 000210
011060 004767 004062
011064 000277
011066 076040
011070 106767 167576
011074 042767 177400 167570
011102 023767 000674 167562
011110 001401
011112 104001
011114
011114 023706 000676
011120 001403
011122 010637 000700
011126 104002
011130
011130 026704 167520
011134 001401
011136 104003
011140 026705 167512
011144 001401
011146 104004
011150
011150 016703 167472
011154 016702 167464
011160 121367 167470
011164 001403

```


J06

```

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 74
DVKAIA.F11 22-DEC-76 10:47 T33 TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE, SL .GT. 100000 SEQ 0075

3169 011166 105723 TSTB (R3)+ :POINT TO NEXT BYTE
3170 011170 005302 DEC R2 :DECREMENT BYTE COUNT
3171 011172 001372 BNE 69$ :BR. IF NOT FINISHED
3172 011174 020200 69$: CMP R2,R0 :CHECK R0=REMAINING SOURCE LENGTH
3173 011176 001401 BEQ 70$
3174 011200 104005 ERROR 5 :*****TEST 33 - ERROR 5*****
3175 :LOCATE LENGTH ERROR
3176 011202 020301 70$: CMP R3,R1 :CHECK R1=ADDRESS OF CHARACTER
3177 011204 001401 BEQ 71$
3178 011206 104005 ERROR 6 :*****TEST 33 - ERROR 6*****
3179 :LOCATE ADDRESS ERROR
3180 011210 71$:

```

K06

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 75
 DVKATA.P11 22-DEC-76 10:47 T33

TEST "LOCC" INSTRUCTION WITH SEARCH CHARACTER IN SOURCE, SL .GT. 100000

SEQ 0076

```

3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192 011210 000004
3193 011212 004567 003774
3194 011216 000377
3195 011220 016434
3196 011222 177777
3197 011224 177777
3198 011226 000000
3199 011230 004567 004052
3200 011234 000204
3201 011236 004767 003704
3202 011242 000244
3203 011244 000276
3204
3205 011246 076040
3206
3207 011250 106767 167416
3208 011254 042767 177400 167410
3209 011262 023767 000674 167402
3210 011270 001401
3211 011272 104001
3212
3213
3214
3215 011274
3216 011274 023706 000676
3217 011300 001403
3218 011302 010637 000700
3219 011306 104002
3220
3221
3222
3223 011310
3224 011310 026704 167340
3225 011314 001401
3226 011316 104003
3227
3228 011320 026705 167332
3229 011324 001401
3230 011326 104004
3231
3232 011330
3233 011330 016703 167312
3234 011334 016702 167304
3235 011340 121367 167310
3236 011344 001403

;*****
;TEST 34 TEST "LOCC" WITH SEARCH CHARACTER NOT IN SOURCE
;*****
;PROPER TERMINATION OF THIS TEST IS R0 EQUALS ZERO, R1
;EQUALS ADDRESS OF LSB OF SOURCE PLUS ONE, AND ALL
;CONDITION CODES CLEAR EXCEPT Z=1.
;*****
;*****
†ST34: SCOPE
      JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
      377 ;SOURCE LENGTH
      BUF1 ;SOURCE ADDRESS
      NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
      NXM ;UNAFFECTED BY DIS STRING INSTRUCTIONS
      0 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
      JSR R5,XPSW ;STORE EXPECTED PSW VALUE
      .WORD 204
      JSR PC,GENR ;SET UP GENERAL REGISTERS
      CLZ ;CLEAR CONDITION CODE Z
      +SEV!SEN!SEZ ;SET ALL OTHER CONDITION CODES
      LOCC ;EXECUTE "LOCC"

      MFPS CCODES ;CHECK RESULTS
      BIC #177400,CCODES ;STORE THE PSW
      CMP @#EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
      BEQ 64$ ;CHECK PSW AGAINST EXPECTED VALUE
      ERROR 1 ;BR, IF EQUAL
      ;*****TEST 34 - 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 34 - 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 34 - ERROR 3*****
      ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
      ;CHECK R5 UNCHANGED
      CMP TABLE,R5
      BEQ 67$ ;BR IF OK
      ERROR 4 ;*****TEST 34 - ERROR 4*****
      ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

67$:
      MOV SRCAD,R3 ;LOCATE "CHARACTER"
      MOV SRCIN,R2
      CMPB (R3),FILL
      BEQ 68$
      68$:

```


L06

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 76
 DVKAIA.P11 22-DEC-76 10:47 T34

TEST "LOCC" WITH SEARCH CHARACTER NOT IN SOURCE

SEQ 0077

3237	011346	105723	TSTB	(R3)+	:POINT TO NEXT BYTE
3238	011350	005302	DEC	R2	:DECREMENT BYTE COUNT
3239	011352	001372	BNE	68\$	:BR IF NOT FINISHED
3240	011354	020200	69\$: CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3241	011356	001401	BEG	70\$	
3242	011360	104005	ERROR	5	:*****TEST 34 - ERROR 5*****
3243					:LOCATE LENGTH ERROR
3244	011362	020301	70\$: CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3245	011364	001401	BEG	71\$	
3246	011366	104006	ERROR	6	:*****TEST 34 - ERROR 6*****
3247					:LOCATE ADDRESS ERROR
3248	011370		71\$:		

```

3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259 011370 000004
3260 011372 004567 003614
3261 011376 000000
3262 011400 177777
3263 011402 177777
3264 011404 177777
3265 011406 000001
3266 011410 004567 003672
3267 011414 000204
3268 011416 004767 003524
3269 011422 000244
3270 011424 000273
3271
3272 011426 076040
3273
3274 011430 106767 167236
3275 011434 042767 177400 167230
3276 011442 023767 000674 167222
3277 011450 001401
3278 011452 104001
3279
3280
3281
3282 011454
3283 011454 023706 000676
3284 011460 001403
3285 011462 010637 000700
3286 011466 104002
3287
3288
3289
3290 011470
3291 011470 105700
3292 011472 001401
3293 011474 104003
3294
3295 011476 026701 167144
3296 011502 001401
3297 011504 104004
3298
3299 011506

*****
*TEST 35 TEST "LOCC" WITH ZERO SOURCE LENGTH
*****
*PROPER TERMINATION OF TEST IS NO CHANGE IN CONTENTS OF
*RO&R1, AND ALL CONDITION CODES CLEAR EXCEPT Z=1.
*****
†ST35: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
      0        ;SOURCE LENGTH
      NXM      ;SOURCE ADDRESS
      NXM      ;STORE NON-ZERO VALUES TO TEST R2 & R3
      1        ;UNAFECTED BY DIS STRING INSTRUCTIONS
      JSR      R5,XPSW      ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
      .WORD    204          ;STORE EXPECTED PSW VALUE
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      CLZ      ;CLEAR CONDITION CODE Z
      +SEV!SEN!SEC          ;SET ALL OTHER CONDITION CODES
      LOCC      ;EXECUTE "LOCC"

      MFPS     CCODES      ;CHECK RESULTS
      BIC      #177400,CCODES ;STORE THE PSW
      CMP      @#EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
      BEQ      64$          ;CHECK PSW AGAINST EXPECTED VALUE
      ERROR    1            ;BR, IF EQUAL
      *****TEST 35 - 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 35 - ERROR 2*****
      STACK POINTER NOT RESTORED BY INSTRUCTION
      EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ERRONEOUS VALUE IS AT "BADR6"

      65$:
      TSTB     R0            ;CHECK R0
      BEQ      1$           ;R0 CHANGED
      ERROR    3            ;*****TEST 35 - ERROR 3*****
      ;R0 CHANGED
      1$:
      CMP      SRCAD,R1      ;CHECK R1
      BEQ      40$          ;*****TEST 35 - ERROR 4*****
      ERROR    4            ;R1 CHANGED

      40$:

```



```

3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311 011506 000004
3312 011510 105777 167024
3313 011514 100536
3314 011516 026767 167026 167134
3315 011524 001007
3316 011526 032767 000001 167052
3317 011534 001403
3318 011536 005767 167032
3319 011542 001123
3320 011544
3321 011544 004567 003442
3322 011550 000377
3323 011552 016434
3324 011554 177777
3325 011556 177777
3326 011560 000200
3327 011562 012767 011646 167100
3328 011570 012777 015356 167066
3329 011576 005077 167064
3330 011602 004767 003524
3331 011606 013777 000554 167046
3332 011614 004567 003466
3333 011620 000000
3334 011622 106427 000000
3335 011626 052777 000100 167024
3336 011634 004767 003306
3337 011640 010637 000676
3338 011644 000277
3339
3340 011646 076040
3341
3342 011650 106767 167016
3343 011654 032777 000100 166776
3344 011662 001364
3345 011664 042767 177400 167000
3346 011672 023767 000674 166772
3347 011700 001401
3348 011702 104001
3349
3350
3351
3352 011704
3353 011704 023706 000676
3354 011710 001403
3355 011712 010637 000700

*****
*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.
*****
TST36: SCOPE
TSTB @SWR ;TEST BIT 7 OF SWR
BMI TST37 ;SKIP TO NEXT TEST IF SET
CMP $TPS, TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T36CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT @BIT0, $ENV ;IF YES, CHECK IF ON APT
BEQ T36CONT ;BR IF NOT UNDER APT, AND DO THIS TEST
TST $PASS ;CHECK IF ON FIRST PASS
BNE TST37 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T36CONT: JSR R5, PREP ;SET UP INSTRUCTION ARGUMENTS
377 ;SOURCE LENGTH
EJF1 ;SOURCE ADDRESS
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
200 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
MOV @LOC, PC ;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 00
MTPS #0 ;SET PSW TO ALLOW INTERRUPTS
BIS #100, @TCSR ;ENABLE TTY INTERRUPTS
RELOC: JSR PC, GENR ;SET UP GENERAL REGISTERS
MOV SP, @SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
SCC ;SET ALL CONDITION CODES

LOC: LOCC ;EXECUTE "LOCC"

MFFS CCODES ;STORE THE PSW
BIT #100, @TCSR ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
BNE RELOC ;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 36 - 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

```

3356	011716	104002		ERROR	2		*****TEST 36 - ERROR 2*****
3357							STACK POINTER NOT RESTORED BY INSTRUCTION
3358							EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3359							ERRONEOUS VALUE IS AT "BADR6"
3360	011720			65\$:			
3361	011720	026704	166730		CMP	FILL,R4	;CHECK R4 UNCHANGED
3362	011724	001401			BEQ	66\$	;BR IF OK
3363	011726	104003			ERROR	3	*****TEST 36 - ERROR 3*****
3364							R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3365	011730	026705	166722	66\$:	CMP	TABLE,R5	;CHECK R5 UNCHANGED
3366	011734	001401			BEQ	67\$	;BR IF OK
3367	011736	104004			ERROR	4	*****TEST 36 - ERROR 4*****
3368							R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3369	011740			67\$:			
3370	011740	016703	166702		MOV	SRCAD,R3	;LOCATE "CHARACTER"
3371	011744	016702	166674		MOV	SRCLN,R2	
3372	011750	121367	166700	68\$:	CMPB	(R3),FILL	
3373	011754	001403			BEQ	69\$	
3374	011756	105723			TSTB	(R3)+	;POINT TO NEXT BYTE
3375	011760	005302			DEC	R2	;DECREMENT BYTE COUNT
3376	011762	001372			BNE	68\$	;BR, IF NOT FINISHED
3377	011764	020200		69\$:	CMP	R2,R0	;CHECK R0=REMAINING SOURCE LENGTH
3378	011766	001401			BEQ	70\$	
3379	011770	104005			ERROR	5	*****TEST 36 - ERROR 5*****
3380							LOCATE LENGTH ERROR
3381	011772	020301		70\$:	CMP	R3,R1	;CHECK P1=ADDRESS OF CHARACTER
3382	011774	001401			BEQ	71\$	
3383	011776	104006			ERROR	6	*****TEST 36 - ERROR 6*****
3384							LOCATE ADDRESS ERROR
3385	012000			71\$:			
3386							
3387	012000	106427	000200	40\$:	MTPS	#200	;RESTORE PSW TO PRIORITY 7
3388	012004	016777	166656 166652		MOV	TPSW,ATVECT	;RESTORE TRAP CATCHER


```

3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402 012012 000004
3403 012014 004567 003172
3404 012020 000007
3405 012022 017034
3406 012024 177777
3407 012026 177777
3408 012030 000001
3409 012032 004567 003250
3410 012036 000200
3411 012040 004767 003102
3412 012044 000277
3413
3414 012046 076041
3415
3416 012050 106767 166616
3417 012054 042767 177400 166610
3418 012062 023767 000674 166602
3419 012070 001401
3420 012072 104001
3421
3422
3423
3424 012074
3425 012074 023706 000676
3426 012100 001403
3427 012102 010637 000700
3428 012106 104002
3429
3430
3431
3432 012110
3433 012110 026704 166540
3434 012114 001401
3435 012116 104003
3436
3437 012120 026705 166532
3438 012124 001401
3439 012126 104004
3440
3441 012130
3442 012130 016703 166512
3443 012134 016702 166504
3444 012140 121367 166510

*****
*TEST 37 TEST "SKPC" INSTRUCTION WITH NON-SKIP CHARACTERS IN SOURCE
*****
*PROPER TERMINATION OF THIS TEST IS R1 EQUALS THE
*ADDRESS OF THE FIRST OCCURRENCE 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
      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      @#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
          *****TEST 37 - ERROR 1*****
      PSW ERROR
      EXPECTED PSW IS STORED AT "EXPPSW"
      ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      @#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;BR IF OK
      MOV      SP,@#BADR6      ;STORE BAD SP VALUE
      ERROR    2          *****TEST 37 - ERROR 2*****
          ;STACK POINTER NOT RESTORED BY INSTRUCTION
          ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
          ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      CMP      FILL,R4          ;CHECK R4 UNCHANGED
      BEQ      66$          ;BR IF OK
      ERROR    3          *****TEST 37 - 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 37 - ERROR 4*****
          ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
      MOV      SRCAD,R3          ;SKIP "CHARACTER"
      MOV      SRCLN,R2
      68$:
      CMPB     (R3),FILL

```

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 81
 DVKAIA.P11 22-DEC-76 10:47 T37

TEST "SKPC" INSTRUCTION WITH NON-SKIP CHARACTERS IN SOURCE

SEQ 0082

3445	012144	001003	BNE	69\$	
3446	012146	105723	TSTB	(R3)+	:POINT TO NEXT BYTE
3447	012150	005302	DEC	R2	:DECREMENT BYTE COUNT
3448	012152	001372	BNE	68\$	:BR, IF NOT FINISHED
3449	012154	020200	69\$: CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3450	012156	001401	BEQ	70\$	
3451	012160	104005	ERROR	5	:*****TEST 37 - ERROR 5*****
3452					:SKIP LENGTH ERROR
3453	012162	020301	70\$: CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3454	012164	001401	BEQ	71\$	
3455	012166	104006	ERROR	6	:*****TEST 37 - ERROR 6*****
3456					:SKIP ADDRESS ERROR
3457	012170		71\$:		


```

3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469 012170 000004
3470 012172 004567 003014
3471 012176 000010
3472 012200 017034
3473 012202 177777
3474 012204 177777
3475 012206 000377
3476 012210 016700 166432
3477 012214 016701 166424
3478 012220 116720 166430
3479 012224 005301
3480 012226 001374
3481 012230 004567 003052
3482 012234 000204
3483 012236 004767 002704
3484 012242 000244
3485 012244 000273
3486
3487 012246 076041
3488
3489 012250 106767 166416
3490 012254 042767 177400 166410
3491 012262 023767 000674 166402
3492 012270 001401
3493 012272 104001
3494
3495
3496
3497 012274
3498 012274 023706 000676
3499 012300 001403
3500 012302 010637 000700
3501 012306 104002
3502
3503
3504
3505 012310
3506 012310 026704 166340
3507 012314 001401
3508 012316 104003
3509
3510 012320 026705 166332
3511 012324 001401
3512 012326 104004
3513

```

```

*****
*TEST 40      TEST "SKPC" WITH ALL SKIP CHARACTERS IN THE SOURCE
*****
*PROPER TERMINATION OF THIS TEST IS RO EQUALS ZERO.
*R1 EQUALS THE ADDRESS+1 OF THE LSB IN THE SOURCE STRING,
*AND ALL CONDITION CODES ARE CLEAR EXCEPT Z=1.
*****
TST40:  SCOPE
        JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
        10          ;SOURCE LENGTH
        BUF2        ;SOURCE ADDRESS
        NXM          ;STORE NON-ZERO VALUES TO TEST R2 & R3
        NXM          ;UNAFFECTED BY DIS STRING INSTRUCTIONS
        377          ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
        MOV SRCAD,R0 ;GENERATE SOURCE STRING
        MOV SRCLN,R1 ;OF ALL SKIP CHARACTERS
        MOVB FILL,(R0)+
        DEC R1
        BNE 1$
        JSR R5,XPSW          ;STORE EXPECTED PSW VALUE
        .WORD 204
        JSR PC,GENR          ;SET UP GENERAL REGISTERS
        CLZ          ;CLEAR CONDITION CODE "Z"
        +SEV!SEN!SEC        ;SET ALL OTHER CONDITION CODES
        SKPC          ;EXECUTE "SKPC"
        MFPS CCODES          ;CHECK RESULTS
        BIC #177400,CCODES   ;STORE THE PSW
        CMP #EXPPSW,CCODES   ;CLEAR ALL UNUSED BITS
        BEQ 64$              ;CHECK PSW AGAINST EXPECTED VALUE
        ERROR 1              ;BR, IF EQUAL
        *****TEST 40 - 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 40 - 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 40 - 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 40 - ERROR 4*****
        R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 83
 DVKAIA.P11 22-DEC-76 10:47 T40

TEST "SKPC" WITH ALL SKIP CHARACTERS IN THE SOURCE

SEQ 0084

3514	012330		67\$:	MOV	SRCAD,R3	:SKIP "CHARACTER"
3515	012330	016703		MOV	SRCIN,R2	
3516	012334	016702		CMPB	(R3),FILL	
3517	012340	121367	68\$:	BNE	69\$	
3518	012344	001003		TSTB	(R3)+	:POINT TO NEXT BYTE
3519	012346	105723		DEC	R2	:DECREMENT BYTE COUNT
3520	012350	005302		BNE	68\$	:BR. IF NOT FINISHED
3521	012352	001372	69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3522	012354	020200		BEQ	70\$	
3523	012356	001401		ERROR	5	:*****TEST 40 - ERROR 5*****
3524	012360	104005				:SKIP LENGTH ERROR
3525			70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3526	012362	020301		BEQ	71\$	
3527	012364	001401		ERROR	6	:*****TEST 40 - ERROR 6*****
3528	012366	104006				:SKIP ADDRESS ERROR
3529			71\$:			
3530	012370					
3531						

3637	012724	001401		BEQ	64\$	:BR, IF EQUAL
3638	012726	104001		ERROR	1	:*****TEST 42 - ERROR 1*****
3639						:PSW ERROR
3640						:EXPECTED PSW IS STORED AT "EXPPSW"
3641						:ACTUAL PSW IS STORED AT "CCODES"
3642	012730			64\$:		
3643	012730	023706	000676	CMP	2#SAVR6, SP	:VERIFY STACK POINTER IS RESTORED
3644	012734	001403		BEQ	65\$	:BR IF OK
3645	012736	010637	000700	MOV	SP, 2#BADR6	:STORE BAD SP VALUE
3646	012742	104002		ERROR	2	:*****TEST 42 - ERROR 2*****
3647						:STACK POINTER NOT RESTORED BY INSTRUCTION
3648						:EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3649						:ERRONEOUS VALUE IS AT "BADR6"
3650	012744			65\$:		
3651	012744	026704	165704	CMP	FILL, R4	:CHECK R4 UNCHANGED
3652	012750	001401		BEQ	66\$	:BR IF OK
3653	012752	104003		ERROR	3	:*****TEST 42 - ERROR 3*****
3654						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3655	012754	026705	165676	66\$:	CMP	TABLE, R5
3656	012760	001401		BEQ	67\$	:CHECK R5 UNCHANGED
3657	012762	104004		ERROR	4	:BR IF OK
3658						:*****TEST 42 - ERROR 4*****
3659	012764			67\$:		:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3660	012764	016703	165656	MOV	SRCAD, R3	:SKIP "CHARACTER"
3661	012770	016702	165650	MOV	SRCIN, R2	
3662	012774	121367	165654	68\$:	CMPB	(R3), FILL
3663	013000	001003		BNE	69\$	
3664	013002	105723		TSTB	(R3)+	:POINT TO NEXT BYTE
3665	013004	005302		DEC	R2	:DECREMENT BYTE COUNT
3666	013006	001372		BNE	68\$	:BR, IF NOT FINISHED
3667	013010	020200		69\$:	CMP	R2, R0
3668	013012	001401		BEQ	70\$	:CHECK R0=REMAINING SOURCE LENGTH
3669	013014	104005		ERROR	5	:*****TEST 42 - ERROR 5*****
3670						:SKIP LENGTH ERROR
3671	013016	020301		70\$:	CMP	R3, R1
3672	013020	001401		BEQ	71\$	:CHECK R1=ADDRESS OF CHARACTER
3673	013022	104006		ERROR	6	:*****TEST 42 - ERROR 6*****
3674						:SKIP ADDRESS ERROR
3675	013024			71\$:		
3676						
3677	013024	106427	000200	40\$:	MTPS	#200
3678	013030	016777	165632 165626	MOV	TPSW, STVECT	:RESTORE PSW TO PRIORITY 7
						:RESTORE TRAP CATCHER

```

3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690 013036 000004
3691 013040 004567 002146
3692 013044 000003
3693 013046 016434
3694 013050 177777
3695 013052 177777
3696 013054 000002
3697 013056 012767 016434 165572
3698 013064 004567 002216
3699 013070 000200
3700 013072 004767 002050
3701 013076 016705 165554
3702 013102 000277
3703
3704 013104 076042
3705
3706 013106 106767 165560
3707 013112 042767 177400 165552
3708 013120 023767 000674 165544
3709 013126 001401
3710 013130 104001
3711
3712
3713
3714 013132
3715 013132 023706 000676
3716 013136 001403
3717 013140 010637 000700
3718 013144 104002
3719
3720
3721
3722 013146
3723 013146 026704 165502
3724 013152 001401
3725 013154 104003
3726
3727 013156 026705 165474
3728 013162 001401
3729 013164 104004
3730
3731 013166
3732 013166 016703 165454
3733 013172 016702 165446
3734 013176 111304

```

```

*****
*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
        SCANC    ;EXECUTE "SCANC"
        MFPS     CCODES          ;CHECK RESULTS
        BIC      #177400,CCODES  ;STORE THE PSW
        CMP      3#EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
        BEQ      64$             ;CHECK PSW AGAINST EXPECTED VALUE
        ERROR    1               ;BR, IF EQUAL
        *****TEST 43 - ERROR 1*****
        PSW ERROR
        EXPECTED PSW IS STORED AT "EXPPSW"
        ACTUAL PSW IS STORED AT "CCODES"
        64$:
        CMP      3#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
        BEQ      65$             ;BR IF OK
        MOV      SP,3#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"
        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
        MOVB     (R3),R4        ;USE SOURCE BYTE AS INDEX

```


K07

```

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 98
OVKAIA.P11 22-DEC-76 10:47 T43 TEST "SCANC" WITH A MEMBER CHARACTER IN SOURCE & SL .GT. 0 AT CHARACTER SEQ 0089

3735 013200 042704 177400 BIC #177400,R4 ;CLEAR ANY BYTE SIGN EXTENSION
3736 013204 136467 016434 165442 BITB BUF1(R4),FILL ;"AND" TABLE ENTRY WITH MASK
3737 013212 001003 BNE 69$
3738 013214 105723 TSTB (R3)+ ;POINT TO NEXT SOURCE BYTE
3739 013216 005302 DEC R2 ;DECREMENT BYTE COUNT
3740 013220 001366 BNE 68$ ;BR, IF NOT FINISHED
3741 013222 020200 59$: CMP R2,R0 ;CHECK R0=REMAINING SOURCE LENGTH
3742 013224 001401 BEQ 70$
3743 013226 104005 ERROR 5 ;*****TEST 43 - ERROR 5*****
3744 ;SCAN LENGTH ERROR
3745 013230 020301 70$: CMP R3,R1 ;CHECK R1=ADDRESS OF CHARACTER
3746 013232 001401 BEQ 71$
3747 013234 104006 ERROR 6 ;*****TEST 43 - ERROR 6*****
3748 ;SCAN ADDRESS ERROR
3749 013236 71$:
3750

```

```

3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761 013236 000004
3762 013240 004567 001746
3763 013244 000200
3764 013246 016434
3765 013250 177777
3766 013252 177777
3767 013254 000000
3768 013256 012767 016434 165372
3769 013264 004567 002016
3770 013270 000204
3771 013272 004767 001650
3772 013276 016705 165354
3773 013302 000277
3774 013304 000244
3775
3776 013306 076042
3777
3778
3779 013310 106767 165356
3780 013314 042767 177400 165350
3781 013322 023767 000674 165342
3782 013330 001401
3783 013332 104001
3784
3785
3786
3787 013334
3788 013334 023706 000676
3789 013340 001403
3790 013342 010637 000700
3791 013346 104002
3792
3793
3794
3795 013350
3796 013350 026704 165300
3797 013354 001401
3798 013356 104003
3799
3800 013360 026705 165272
3801 013364 001401
3802 013366 104004
3803
3804 013370
3805 013370 016703 165252
3806 013374 016702 165244

```

```

*****
*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.
*****
TST44:  SCOPE
        JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
                                ;SOURCE LENGTH
                                ;SOURCE ADDRESS
                                ;STORE NON-ZERO VALUES TO TEST R2 & R3
                                ;UNAFECTED BY DIS STRING INSTRUCTIONS
                                ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
                                ;STORE EXPECTED PSW VALUE
                                ;SET UP GENERAL REGISTERS
                                ;SET ALL CONDITION CODES, EXCEPT
                                ;CLEAR "Z" CONDITION CODE
                                ;EXECUTE "SCANC"
                                ;CHECK RESULTS
                                ;STORE THE PSW
                                ;CLEAR ALL UNUSED BITS
                                ;CHECK PSW AGAINST EXPECTED VALUE
                                ;BR, IF EQUAL
                                ;*****TEST 44 - ERROR 1*****
                                ;PSW ERROR
                                ;EXPECTED PSW IS STORED AT "EXPPSW"
                                ;ACTUAL PSW IS STORED AT "CCODES"
                                ;VERIFY STACK POINTER IS RESTORED
                                ;BR IF OK
                                ;STORE BAD SP VALUE
                                ;*****TEST 44 - ERROR 2*****
                                ;STACK POINTER NOT RESTORED BY INSTRUCTION
                                ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                                ;ERRONEOUS VALUE IS AT "BADR6"
                                ;CHECK R4 UNCHANGED
                                ;BR IF OK
                                ;*****TEST 44 - ERROR 3*****
                                ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
                                ;CHECK R5 UNCHANGED
                                ;BR IF OK
                                ;*****TEST 44 - ERROR 4*****
                                ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
                                ;POINT R3 TO SOURCE STRING
                                ;STORE SOURCE LENGTH IN R2

```

```

        MOV      R5,PREP
        BUF1
        NXM
        NXM
        0
        MOV      #BUF1, TABLE
        JSR      R5, XPSW
        .WORD    204
        JSR      PC, GENR
        MOV      TABLE, R5
        SCC
        CLZ
        SCANC
        MFPS
        BIC      #177400, CCODES
        CMP      3#EXPPSW, CCODES
        BEQ      64$
        ERROR    1
        64$:
        CMP      3#SAVR6, SP
        BEQ      65$
        MOV      SP, 3#BADR6
        ERROR    2
        65$:
        CMP      FILL, R4
        BEQ      66$
        ERROR    3
        66$:
        CMP      TABLE, R5
        BEQ      67$
        ERROR    4
        67$:
        MOV      SRCAD, R3
        MOV      SRCLN, R2

```


M07

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 90
 DVKAIA.P11 22-DEC-76 10:47 T44

TEST "SCANC" FAILS, ALL NON-MEMBER CHARACTERS IN STRING

SEQ 0091

3807	013400	111304		68\$:	MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
3808	013402	042704	177400		BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
3809	013406	136467	016434 165240		BITB	BUF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
3810	013414	001003			BNE	69\$	
3811	013416	105723			TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
3812	013420	005302			DEC	R2	:DECREMENT BYTE COUNT
3813	013422	001366			BNE	68\$	:BR, IF NOT FINISHED
3814	013424	020200		69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3815	013426	001401			BEQ	70\$	
3816	013430	104005			ERROR	5	:*****TEST 44 - ERROR 5*****
3817							:SCAN LENGTH ERROR
3818	013432	020301		70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3819	013434	001401			BEQ	71\$	
3820	013436	104006			ERROR	6	:*****TEST 44 - ERROR 6*****
3821							:SCAN ADDRESS ERROR
3822	013440			71\$:			

N07

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 91
 DVKAIA.P11 22-DEC-76 10:47 T44

TEST "SCANC" FAILS, ALL NON-MEMBER CHARACTERS IN STRING

SEQ 0092

```

3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835 013440 000004
3836 013442 105777 165072
3837 013446 100550
3838 013450 026767 165074 165202
3839 013456 001007
3840 013460 032767 000001 165120
3841 013466 001403
3842 013470 005767 165100
3843 013474 001135
3844 013476
3845 013476 004567 001510
3846 013502 000400
3847 013504 016434
3848 013506 177777
3849 013510 177777
3850 013512 000000
3851 013514 012767 016434 165134
3852 013522 012767 013614 165140
3853 013530 012777 015356 165126
3854 013536 005077 165124
3855 013542 004767 001564
3856 013546 013777 000554 165106
3857 013554 004567 001526
3858 013560 000004
3859 013562 106427 000000
3860 013566 052777 000100 165064
3861 013574 004767 001346
3862 013600 010637 000676
3863 013604 016705 165046
3864 013610 000244
3865 013612 000267
3866
3867 013614 076042
3868
3869
3870 013616 106767 165050
3871 013622 032777 000100 165030
3872 013630 001361
3873 013632 042767 177400 165032
3874 013640 023767 000674 165024
3875 013646 001401
3876 013650 104001
3877
3878

;*****
;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.
;*****
;*****
TST45: SCOPE
      TSTB @SWR ;TEST BIT 7 OF SWR
      BMI TST46 ;SKIP TO NEXT TEST IF SET
      CMP $TPS,TCSR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
      BNE T45CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
      BIT @BIT0,$ENV ;IF YES, CHECK IF ON APT
      BEQ T45CONT ;BR IF NOT UNDER APT, AND DO THIS TEST
      TST $PASS ;CHECK IF ON FIRST PASS
      BNE TST46 ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T45CONT: JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
          ;SOURCE LENGTH
          ;SOURCE ADDRESS
          ;STORE NON-ZERO VALUES TO TEST R2 & R3
          ;UNAFFECTED BY DIS STRING INSTRUCTIONS
          ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
          MOV @BUF1,TABLE
          MOV @SCN,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 @R,$NULL,@TBUF ;SEND CARRIAGE RETURN
          JSR R5,XPSW ;STORE EXPECTED PSW VALUE
          .WORD 04
          MTPS #0 ;SET PSW TO ALLOW INTERRUPTS
          BIS #100,@TCR ;ENABLE TTY INTERRUPTS
          JSR PC,GENR ;SET UP GENERAL REGISTERS
          MOV SP,@$SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
          MOV TABLE,R5
          CLZ ;CLEAR "Z" CONDITION CODE
          +SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES

SCN: SCANC ;EXECUTE "SCANC"

          ;CHECK RESULTS
          ;STORE THE PSW
          ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
          ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
          BNE REPCN
          BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
          CMP @EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
          BEQ 64$ ;BR, IF EQUAL
          ERROR 1 ;*****TEST 45 - ERROR 1*****
          ;PSW ERROR
          ;EXPECTED PSW IS STORED AT "EXPPSW"

```


3879								;ACTUAL PSW IS STORED AT "CCODES"
3880	013652			64\$:				
3881	013652	023706	000676		CMP	Q#SAVR6, SP		;VERIFY STACK POINTER IS RESTORED
3882	013656	001403			BEQ	65\$		;BR IF OK
3883	013660	010637	000700		MOV	SP, Q#BADR6		;STORE BAD SP VALUE
3884	013664	104002			ERROR	2		;*****TEST 45 - ERROR 2*****
3885								;STACK POINTER NOT RESTORED BY INSTRUCTION
3886								;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
3887								;ERRONEOUS VALUE IS AT "BADR6"
3888	013666			65\$:				
3889	013666	026704	164762		CMP	FILL, R4		;CHECK R4 UNCHANGED
3890	013672	001401			BEQ	66\$		;BR IF OK
3891	013674	104003			ERROR	3		;*****TEST 45 - ERROR 3*****
3892								;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
3893	013676	026705	164754	66\$:	CMP	TABLE, R5		;CHECK R5 UNCHANGED
3894	013702	001401			BEQ	67\$		;BR IF OK
3895	013704	104004			ERROR	4		;*****TEST 45 - ERROR 4*****
3896								;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
3897	013706			67\$:				
3898	013706	016703	164734		MOV	SRCAD, R3		;POINT R3 TO SOURCE STRING
3899	013712	016702	164726		MOV	SRCLN, R2		;STORE SOURCE LENGTH IN R2
3900	013716	111304		68\$:	MOVB	(R3), R4		;USE SOURCE BYTE AS INDEX
3901	013720	042704	177400		BIC	#177400, R4		;CLEAR ANY BYTE SIGN EXTENSION
3902	013724	136467	016434 164722		BITB	BUF1(R4), FILL		; "AND" TABLE ENTRY WITH MASK
3903	013732	001003			BNE	69\$		
3904	013734	105723			TSTB	(R3)+		;POINT TO NEXT SOURCE BYTE
3905	013736	005302			DEC	R2		;DECREMENT BYTE COUNT
3906	013740	001366			BNE	68\$		;BR, IF NOT FINISHED
3907	013742	020200		69\$:	CMP	R2, R0		;CHECK R0=REMAINING SOURCE LENGTH
3908	013744	001401			BEQ	70\$		
3909	013746	104005			ERROR	5		;*****TEST 45 - ERROR 5*****
3910								;SCAN LENGTH ERROR
3911	013750	020301		70\$:	CMP	R3, R1		;CHECK R1=ADDRESS OF CHARACTER
3912	013752	001401			BEQ	71\$		
3913	013754	104006			ERROR	6		;*****TEST 45 - ERROR 6*****
3914								;SCAN ADDRESS ERROR
3915	013756			71\$:				
3916								
3917	013756	106427	000200	40\$:	MTPS	#200		;RESTORE PSW TO PRIORITY 7
3918	013762	016777	164700 164674		MOV	TPSW, TVECT		;RESTORE TRAP CATCHER

```

3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931 013770 000004
3932 013772 004567 001214
3933 013776 000020
3934 014000 016434
3935 014002 177777
3936 014004 177777
3937 014006 000007
3938 014010 012767 016434 164640
3939 014016 004567 001264
3940 014022 000200
3941 014024 004767 001116
3942 014030 016705 164622
3943 014034 000244
3944 014036 000267
3945
3946 014040 076043
3947
3948 014042 106767 164624
3949 014046 042767 177400 164616
3950 014054 023767 000674 164610
3951 014062 001401
3952 014064 104001
3953
3954
3955
3956 014066
3957 014066 023706 000676
3958 014072 001403
3959 014074 010637 000700
3960 014100 104002
3961
3962
3963
3964 014102
3965 014102 026704 164546
3966 014106 001401
3967 014110 104003
3968
3969 014112 026705 164540
3970 014116 001401
3971 014120 104004
3972
3973 014122
3974 014122 016703 164520

*****
*TEST 46 TEST "SPANC" WITH NON-MEMBER CHARACTER IN SOURCE, SL .GT. 0 AT CHARACTER
*****
*THIS TEST VERIFIES THAT "SPANC" INDICATES A NON-MEMBER
*FOUND.
*THE RESULT IS R1 EQUALS ADDRESS OF NON-MEMBER CHARACTER, R0
*EQUALS THE REMAINING SOURCE LENGTH, AND ALL
*CONDITION CODES CLEAR.
*****
TST46: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
20 ;SOURCE LENGTH
BUF1 ;SOURCE ADDRESS
NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
NXM ; UNAFFECTED BY DIS STRING INSTRUCTIONS
7 ;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
CLZ ;CLEAR "Z" CONDITION CODE
+SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES
;EXECUTE "SPANC"
SPANC ;CHECK RESULTS
MFPS CCODES ;STORE THE PSW
BIC #177400, CCODES ;CLEAR ALL UNUSED BITS
CMP @#EXPPSW, CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64$ ;BR, IF EQUAL
ERROR 1 ;*****TEST 46 - ERROR 1*****
;PSW ERROR
;EXPECTED PSW IS STORED AT "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 46 - 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 46 - 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 46 - ERROR 4*****
;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$: MOV SRCAD, R3 ;POINT R3 TO SOURCE STRING

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 94
 DVKAIA.P11 22-DEC-76 10:47 T46 TEST "SPANC" WITH NON-MEMBER CHARACTER IN SOURCE, SL .GT. 0 AT CHARACTER SEQ 0095

3975	014126	016702	164512		MOV	SACLN,R2	:STORE SOURCE LENGTH IN R2
3976	014132	111304		68\$:	MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
3977	014134	042704	177400		BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
3978	014140	136467	016434 164506		BITB	BUF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
3979	014146	001403			BEQ	69\$	
3980	014150	105723			TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
3981	014152	005302			DEC	R2	:DECREMENT BYTE COUNT
3982	014154	001366			BNE	68\$	:BR, IF NOT FINISHED
3983	014156	020200		69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
3984	014160	001401			BEQ	70\$	
3985	014162	104005			ERROR	5	:*****TEST 46 - ERROR 5*****
3986							:SPAN LENGTH ERROR
3987	014164	020301		70\$:	CMF	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
3988	014166	001401			BEQ	71\$	
3989	014170	104006			ERROR	6	:*****TEST 46 - ERROR 6*****
3990							:SPAN ADDRESS ERROR
3991	014172			71\$:			

3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047

014172 000004
014174 004567 001012
014200 000100
014202 016434
014204 177777
014206 177777
014210 000377
014212 012767 016434 164436
014220 004567 001062
014224 000204
014226 004767 000714
014232 016705 164420
014236 000244
014240 000267
014242 076043
014244 106767 164422
014250 042767 177400 164414
014256 023767 000674 164406
014264 001401
014266 104001
014270
014270 023706 000676
014274 001403
014276 010637 000700
014302 104002
014304
014304 026704 164344
014310 001401
014312 104003
014314 026705 164336
014320 001401
014322 104004
014324
014324 016703 164316
014330 016702 164310

```

*****
*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
*****
*ST47: SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
                          ;SOURCE LENGTH
                          ;SOURCE ADDRESS
                          ;STORE NON-ZERO VALUES TO TEST R2 & R3
                          ;UNAFFECTED BY DIS STRING INSTRUCTIONS
                          ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER
      MOV      #BUF1,TABLE
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD    204
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      MOV      TABLE,R5
      CLZ
      +SEV!SEZ!SEC *      ;CLEAR "Z" CONDITION CODE
                          ;SET ALL OTHER CONDITION CODES
                          ;EXECUTE "SPANC"
      SPANC
                          ;CHECK RESULTS
                          ;STORE THE PSW
      MFPS     CCODES      ;CLEAR ALL UNUSED BITS
      BIC      #177400,CCODES
      CMP      3#EXPPSW,CCODES
      BEQ      64$
      ERROR    1
                          ;*****TEST 47 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "EXPPSW"
                          ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      3#SAVR6,SP   ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$
      MOV      SP,3#BADR6
      ERROR    2
                          ;*****TEST 47 - 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
                          ;*****TEST 47 - ERROR 3*****
                          ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
                          ;CHECK R5 UNCHANGED
      66$:
      CMP      TABLE,R5
      BEQ      67$
      ERROR    4
                          ;*****TEST 47 - 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

```


MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 96
 DVKAIA.P11 22-DEC-76 10:47 T47

TEST "SPANC" FAILS ONLY MEMBER CHARACTERS IN SOURCE

SEQ 0097

4048	014334	111304		69\$:	MOVB	(R3),R4	:USE SOURCE BYTE AS INDEX
4049	014336	042704	177400		BIC	#177400,R4	:CLEAR ANY BYTE SIGN EXTENSION
4050	014342	136467	016434 164304		BITB	BUF1(R4),FILL	: "AND" TABLE ENTRY WITH MASK
4051	014350	001403			BEQ	69\$	
4052	014352	105723			TSTB	(R3)+	:POINT TO NEXT SOURCE BYTE
4053	014354	005302			DEC	R2	:DECREMENT BYTE COUNT
4054	014356	001366			BNE	68\$	:BR, IF NOT FINISHED
4055	014360	020200		69\$:	CMP	R2,R0	:CHECK R0=REMAINING SOURCE LENGTH
4056	014362	001401			BEQ	70\$	
4057	014364	104005			ERROR	5	:*****TEST 47 - ERROR 5*****
4058							:SPAN LENGTH ERROR
4059	014366	020301		70\$:	CMP	R3,R1	:CHECK R1=ADDRESS OF CHARACTER
4060	014370	001401			BEQ	71\$	
4061	014372	104006			ERROR	6	:*****TEST 47 - ERROR 6*****
4062							:SPAN ADDRESS ERROR
4063	014374			71\$:			

4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119

014374 000004
014376 105777 164136
014402 100547
014404 026767 164140 164246
014412 001007
014414 032767 000001 164164
014422 001403
014424 005767 164144
014430 001134
014432
014432 004567 000554
014436 000400
014440 016434
014442 177777
014444 177777
014446 000177
014450 012767 016434 164200
014456 012767 014546 164204
014464 012777 015356 164172
014472 005077 164170
014476 004767 000630
014502 013777 000554 164152
014510 004567 000572
014514 000000
014516 106427 000000
014522 052777 000100 164130
014530 004767 000412
014534 010637 000676
014540 016705 164112
014544 000277
014546 076043
014550 106767 164116
014554 032777 000100 164076
014562 001362
014564 042767 177400 164100
014572 023767 000674 164072
014600 001401
014602 104001

```

*****
*TEST 50      TEST INTERRUPTABILITY OF "SPANC"
*****
*THIS TEST INTERRUPTS THE "SPANC" INSTRUCTION, RESUMES
*EXECUTION AFTER THE INTERRUPT, AND CHECKS RESULTS.
*RESULTS ARE R1 EQUALS ADDRESS OF FIRST BYTE WHICH
*WHEN ANDED WITH MASK=0, R0 EQUALS REMAINING SOURCE
*LENGTH, AND ALL CONDITION CODES CLEAR
*****
TST50:  SCOPE
        TSTB    $SWR      ;TEST BIT 7 OF SWR
        BMI     $EOP      ;SKIP TO NEXT TEST IF SET
        CMP     $TPS,TCR  ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
        BNE     SPCCONT   ;BR. IF NOT & PERFORM INTERRUPTABILITY TEST
        BIT     $BIT0,$ENV ;IF YES, CHECK IF ON APT
        BEQ     SPCCONT   ;BR IF NOT UNDER APT, AND DO THIS TEST
        TST     $PASS     ;CHECK IF ON FIRST PASS
        BNE     $EOP      ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

SPCCONT: JSR     R5,PREP   ;SET UP INSTRUCTION ARGUMENTS
        400      ;SOURCE LENGTH
        BUF1     ;SOURCE ADDRESS
        NXM      ;DESTINATION LENGTH
        NXM      ;DESTINATION ADDRESS
        177      ;FILL CHARACTER

        MOV     $BUF1,TABLE ;STORE PC OF TEST INSTRUCTION
        MOV     $SPN,PCI    ;POINT TTY VECTOR TO INTERRUPT ROUTINE
        MOV     $INTR,$TVECT ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
        CLR     $TPSW
        JSR     PC,TDONE
        MOV     $,$NULL,$TBUF ;SEND CARRIAGE RETURN
        JSR     R5,$XPSW     ;STORE EXPECTED PSW VALUE
        00
        .WORD   00
        MTPS    $0          ;SET PSW TO ALLOW INTERRUPTS
        BIS     $100,$TCR   ;ENABLE TTY INTERRUPTS
        JSR     PC,GENR     ;SET UP GENERAL REGISTERS
        MOV     SP,$SAVR6   ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
        MOV     TABLE,R5
        SCC
        ;SET ALL CONDITION CODES

SPN:     SPANC              ;EXECUTE "SPANC"

        MFPS    CCODES     ;CHECK RESULTS
        BIT     $100,$TCR  ;STORE THE PSW
        BNE     REPSN      ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
        BIC     $177400,CCODES ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
        CMP     $,$EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
        BEQ     $4$        ;CHECK PSW AGAINST EXPECTED VALUE
        ERROR   1          ;BR, IF EQUAL
        ;*****TEST 50 - ERROR *****
        ;PSW ERROR
        ;EXPECTED PSW IS STORED AT "EXPPSW"

```


PC	Instruction	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418
----	-------------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

```

4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215

.SBTTL  END OF PASS ROUTINE

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

$EOP:
SCOPE
CLR $STNM          ;; ZERO THE TEST NUMBER
INC $PASS          ;; INCREMENT THE PASS NUMBER
BIC #100000,$PASS  ;; DON'T ALLOW A NEG. NUMBER
DEC (PC)+          ;; LOOP?
$EOPCT: .WORD 1
BGT $DOAGN         ;; YES
MOV (PC)+,2(PC)+   ;; RESTORE COUNTER
$ENDCT: .WORD 1
$EOPCT
TYPE. MSG          ;; TYPE "END PASS"
$GET42: MOV #12,RO  ;; GET MONITOR ADDRESS
BEG $DOAGN         ;; BRANCH IF NO MONITOR
CLR -(SP)          ;; INSURE THE "T" BIT IS CLEAR
MOV #SCLR.T, -(SP) ;; SETUP FOR AN RTI OR RTT
BR $RTN           ;; GO DO AN RTI OR RTT TO LOAD THE PSW
                ;; WITH A CLEARED "T" BIT

SCLR.T:
MOV #42,RO        ;; INSURE RO CONTAINS THE MONITORS
BEG $DOAGN        ;; RETURN ADDRESS
RESET            ;; CLEAR THE WORLD
$ENDAD: JSR PC,(RO) ;; GO TO MONITOR
NOP              ;; SAVE ROOM
NOP              ;; FOR
NOP              ;; ACT11
$DOAGN: TRAP
BIC #20,(SP)      ;; PUSH OLD PSW AND PC ON STACK
BIT #BIT12,$SWR   ;; CLEAR THE "T" BIT
BNE 1$           ;; RUN WITH TRACE TRAP?
COM $TBIT        ;; BR IF NO
BMI 1$           ;; IS IT TIME FOR TRACE TRAP
BIS #20,(SP)     ;; BR IF NO
MOV #SLOOP, -(SP) ;; SET TRACE TRAP
$RTN: RTI        ;; JUMP TO START OF TEST
                ;; RETURN--THIS IS CHANGED TO
                ;; AN "RTT" IF "RTT" IS A LEGAL
                ;; INSTRUCTION

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

```


J08

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 100
OVKAIA.F11 22-DEC-76 10:47 END OF PASS ROUTINE

SEG 0101

4216 015104 000
4217 015106

.EVEN

ADDRESS	PC	PC+2	PC+4	PC+6	PC+8	PC+10	PC+12	PC+14	PC+16	PC+18	PC+20	PC+22	PC+24	PC+26	PC+28	PC+30	PC+32	PC+34	PC+36	PC+38	PC+40	PC+42	PC+44	PC+46	PC+48	PC+50	PC+52	PC+54	PC+56	PC+58	PC+60	PC+62	PC+64	PC+66	PC+68	PC+70	PC+72	PC+74	PC+76	PC+78	PC+80	PC+82	PC+84	PC+86	PC+88	PC+90	PC+92	PC+94	PC+96	PC+98	PC+100	PC+102	PC+104	PC+106	PC+108	PC+110	PC+112	PC+114	PC+116	PC+118	PC+120	PC+122	PC+124	PC+126	PC+128	PC+130	PC+132	PC+134	PC+136	PC+138	PC+140	PC+142	PC+144	PC+146	PC+148	PC+150	PC+152	PC+154	PC+156	PC+158	PC+160	PC+162	PC+164	PC+166	PC+168	PC+170	PC+172	PC+174	PC+176	PC+178	PC+180	PC+182	PC+184	PC+186	PC+188	PC+190	PC+192	PC+194	PC+196	PC+198	PC+200	PC+202	PC+204	PC+206	PC+208	PC+210	PC+212	PC+214	PC+216	PC+218	PC+220	PC+222	PC+224	PC+226	PC+228	PC+230	PC+232	PC+234	PC+236	PC+238	PC+240	PC+242	PC+244	PC+246	PC+248	PC+250	PC+252	PC+254	PC+256	PC+258	PC+260	PC+262	PC+264	PC+266	PC+268	PC+270	PC+272	PC+274	PC+276	PC+278	PC+280	PC+282	PC+284	PC+286	PC+288	PC+290	PC+292	PC+294	PC+296	PC+298	PC+300	PC+302	PC+304	PC+306	PC+308	PC+310	PC+312	PC+314	PC+316	PC+318	PC+320	PC+322	PC+324	PC+326	PC+328	PC+330	PC+332	PC+334	PC+336	PC+338	PC+340	PC+342	PC+344	PC+346	PC+348	PC+350	PC+352	PC+354	PC+356	PC+358	PC+360	PC+362	PC+364	PC+366	PC+368	PC+370	PC+372	PC+374	PC+376	PC+378	PC+380	PC+382	PC+384	PC+386	PC+388	PC+390	PC+392	PC+394	PC+396	PC+398	PC+400	PC+402	PC+404	PC+406	PC+408	PC+410	PC+412	PC+414	PC+416	PC+418	PC+420	PC+422	PC+424	PC+426	PC+428	PC+430	PC+432	PC+434	PC+436	PC+438	PC+440	PC+442	PC+444	PC+446	PC+448	PC+450	PC+452	PC+454	PC+456	PC+458	PC+460	PC+462	PC+464	PC+466	PC+468	PC+470	PC+472	PC+474	PC+476	PC+478	PC+480	PC+482	PC+484	PC+486	PC+488	PC+490	PC+492	PC+494	PC+496	PC+498	PC+500	PC+502	PC+504	PC+506	PC+508	PC+510	PC+512	PC+514	PC+516	PC+518	PC+520	PC+522	PC+524	PC+526	PC+528	PC+530	PC+532	PC+534	PC+536	PC+538	PC+540	PC+542	PC+544	PC+546	PC+548	PC+550	PC+552	PC+554	PC+556	PC+558	PC+560	PC+562	PC+564	PC+566	PC+568	PC+570	PC+572	PC+574	PC+576	PC+578	PC+580	PC+582	PC+584	PC+586	PC+588	PC+590	PC+592	PC+594	PC+596	PC+598	PC+600	PC+602	PC+604	PC+606	PC+608	PC+610	PC+612	PC+614	PC+616	PC+618	PC+620	PC+622	PC+624	PC+626	PC+628	PC+630	PC+632	PC+634	PC+636	PC+638	PC+640	PC+642	PC+644	PC+646	PC+648	PC+650	PC+652	PC+654	PC+656	PC+658	PC+660	PC+662	PC+664	PC+666	PC+668	PC+670	PC+672	PC+674	PC+676	PC+678	PC+680	PC+682	PC+684	PC+686	PC+688	PC+690	PC+692	PC+694	PC+696	PC+698	PC+700	PC+702	PC+704	PC+706	PC+708	PC+710	PC+712	PC+714	PC+716	PC+718	PC+720	PC+722	PC+724	PC+726	PC+728	PC+730	PC+732	PC+734	PC+736	PC+738	PC+740	PC+742	PC+744	PC+746	PC+748	PC+750
---------	----	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------


```

4267
4268
4269
4270 015240 016700 163406      CLDST:  MOV    DSTAD,RO      ;POINT RO TO DESTINATION AREA
4271 015244 005300              DEC    RO              ;CLEAR LOWER BOUNDARY BYTE OF DEST
4272 015246 105020              CLRB   (RO)+
4273 015250 016701 163374      MOV    DSTLN,R1      ;STORE DEST. BYTE COUNT IN R1
4274 015254 105020      CONCL:  CLRB   (RO)+      ;CLEAR DESTINATION
4275 015256 005301              DEC    R1              ;DECREMENT DEST. BYTE COUNT
4276 015260 001375              BNE    CONCL        ;BR. IF NOT FINISHED
4277 015262 105020              CLRB   (RO)+      ;CLEAR UPPER BYTE BOUNDARY OF DEST
4278 015264 000207              RTS     PC              ;RETURN
4279
4280
4281 015266 012700 016434      GENSRC: MOV    #BUF1,RO      ;POINT RO TO THE PATTERN STORED
4282 015272 012502              MOV    (R5)+,R2      ;STORE SOURCE BYTE COUNT IN R2
4283 015274 012501              MOV    (R5)+,R1      ;POINT R1 TO THE SOURCE ADDRESS
4284 015276 112021      CON:  MOVB   (RO)+,(R1)+      ;TRANSFER PATTERN TO SOURCE
4285 015300 005302              DEC    R2              ;DECREMENT BYTE COUNT
4286 015302 001375              BNE    GENCON        ;BR. IF NOT FINISHED
4287 015304 000205              RTS     R5              ;RETURN
4288
4289
4290
4291 015306 012537 000674      ;SUBROUTINE TO RECORD EXPECTED PSW
4292 015312 106700      XPSW:  MOV    (R5)+,3#EXPPSW  ;STORE EXPECTED PSW VALUE
4293 015314 032700              MFPS   RO
4294 015320 001403              BIT    #TBIT,RO
4295 015322 052737 000020 000674  BEQ    1$
4296 015330 000205              BIS    #TBIT,3#EXPPSW ;OTHERWISE SET T-BIT IN EXPECTED PSW VALUE
4297 1$:  RTS     R5
4298

```

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 103
 DVK1A.P11 22-DEC-76 10:47 END OF PASS ROUTINE

SEQ 0104

```

4299
4300
4301      ;SUBROUTINE TO TEST FOR TRANSMIT DONE FLAG
4302 015332 005077 163346      TDONE: CLR      @TEMP      ;CLEAR A TIMER
4303 015336 105777 163316      1$:  TSTB      @TCSR      ;IS SLU READY?
4304 015342 100404              BMI      RETN      ;BR IF READY
4305 015344 005277 163334      INC      @TEMP      ;OTHERWISE INCREMENT TIMER
4306 015350 001372              BNE      1$        ;BR IF NOT TIMED OUT
4307 015352 104300              ERROR    300        ;*****ERROR 300*****
4308                                ;NEVER GOT TRANSMIT DONE FLAG
4309 015354 000207      RETN:    RTS      PC        ;RETURN
4310
4311      ;SUBROUTINE TO HANDLE TTY INTERRUPTS IN INSTRUCTION
4312      ;INTERRUPTABILITY TESTS
4313
4314 015356      INTR:
4315 015356 021667 163306      CMP      (SP),PCI      ;WAS PC AT INSTRUCTION UNDER TEST?
4316 015362 001003              BNE      SEND        ;BR, IF NO
4317 015364 032704 177400      CKR4:  BIT      #177400,R4 ;IF YES, CHECK UPPER BYTE OF R4
4318 015370 001004              BNE      CLRINT      ;IF ZERO, INSTRUCTION WAS NOT INTERRUPTED-TRY AGAIN
4319 015372 013777 000554 163262 SEND:  MOV      @#NULL,@TBUF ;SEND ANOTHER CHARACTER
4320 015400 000002              RTI
4321 015402 042777 000100 163250 CLRINT: BIC      #100,@TCSR ;IF NON-ZERO, CLEAR INTERRUPT ENABLE
4322 015410 000002              RTI                ;CONTINUE INSTRUCTION

```


4323
 4324
 4325
 4326
 4327
 4328
 4329
 4330
 4331
 4332
 4333
 4334
 4335
 4336
 4337
 4338
 4339
 4340
 4341
 4342
 4343
 4344
 4345
 4346
 4347
 4348
 4349
 4350
 4351
 4352
 4353
 4354
 4355
 4356
 4357
 4358
 4359
 4360
 4361
 4362
 4363
 4364
 4365
 4366
 4367
 4368
 4369
 4370
 4371
 4372
 4373
 4374

.SBTTL SCOPE HANDLER ROUTINE

```

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

$SCOPE:
1$: BIT      #BIT14,$SWR      ;;LOOP ON PRESENT TEST?
   BNE      $OVER           ;;YES IF SW14=1
*****START OF CODE FOR THE XOR TESTER*****
$XTSTR: BR      6$          ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                               THIS INSTRUCTION TO A "NOP" (NOP=240)
                               SAVE THE CONTENTS OF THE ERROR VECTOR
                               SET FOR TIMEOUT
                               TIME OUT ON XOR?
                               RESTORE THE ERROR VECTOR
                               GO TO THE NEXT TEST
                               CLEAR THE STACK AFTER A TIME OUT
                               RESTORE THE ERROR VECTOR
                               LOOP ON THE PRESENT TEST
6$: *****END OF CODE FOR THE XOR TESTER*****
   BIT      #BIT08,$SWR      ;;LOOP ON SPEC. TEST?
   BEQ      2$              ;;BR IF NO
   MOV      $SWR,-(SP)       ;;SET DESIRED TEST NUM. FROM SWR
   BIC      #$$WRMK,(SP)     ;;STRIP AWAY UNDESIRED BITS
   CMPB     (SP)+,$TSTNM     ;;ON THE RIGHT TEST?
   BEQ      $OVER           ;;BR IF YES
   TSTB     $ERFLG          ;;HAS AN ERROR OCCURRED?
   BEQ      $SVLAD          ;;BR IF NO
   BIT      #BIT09,$SWR     ;;LOOP ON ERROR?
   BEQ      4$              ;;BR IF NO
   MOV      $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
   BR      $OVER
4$: CLRB     $ERFLG          ;;ZERO THE ERROR FLAG
   $SVLAD: INCB     $TSTNM   ;;COUNT TEST NUMBERS
   MOVB     $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
   MOV      (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
   MOV      (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
   CLR      $ESCAPE        ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
   MOVB     #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
   MOV      $TSTNM,$DISPLAY ;;DISPLAY TEST NUMBER
   MOV      $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
   RTI                    ;;FIXES PS

```

```

4375
4376
4377
4378
4379
4380 015610 012737 015772 000024 $PWRDN: MOV $SILLUP, @#PWRVEC ;; SET FOR FAST UP
4381 015616 012737 000340 000026 MOV #340, @#PWRVEC+2 ;; PRI0:7
4382 015624 010046 MOV R0, -(SP) ;; PUSH R0 ON STACK
4383 015626 010146 MOV R1, -(SP) ;; PUSH R1 ON STACK
4384 015630 010246 MOV R2, -(SP) ;; PUSH R2 ON STACK
4385 015632 010346 MOV R3, -(SP) ;; PUSH R3 ON STACK
4386 015634 010446 MOV R4, -(SP) ;; PUSH R4 ON STACK
4387 015636 010546 MOV R5, -(SP) ;; PUSH R5 ON STACK
4388 015640 017746 162674 MOV @SWR, -(SP) ;; PUSH @SWR ON STACK
4389 015644 010667 000126 MOV SP, $SAVR6 ;; SAVE SP
4390 015650 012737 015662 000024 MOV $PWRUP, @#PWRVEC ;; SET UP VECTOR
4391 015656 000000 HALT
4392 015660 000776 BR -2 ;; HANG UP
4393
4394
4395
4396 015662 012737 015772 000024 $PWRUP: MOV $SILLUP, @#PWRVEC ;; SET FOR FAST DOWN
4397 015670 016706 000102 MOV $SAVR6, SP ;; GET SP
4398 015674 005067 000076 CLR $SAVR6 ;; WAIT LOOP FOR THE TTY
4399 015700 005267 000072 1$: INC $SAVR6 ;; WAIT FOR THE INC
4400 015704 001375 BNE 1$ ;; OF WORD
4401 015706 005067 162570 CLR $STNM
4402 015712 012677 162622 MOV (SP)+, @SWR ;; POP STACK INTO @SWR
4403 015716 012605 MOV (SP)+, R5 ;; POP STACK INTO R5
4404 015720 012604 MOV (SP)+, R4 ;; POP STACK INTO R4
4405 015722 012603 MOV (SP)+, R3 ;; POP STACK INTO R3
4406 015724 012602 MOV (SP)+, R2 ;; POP STACK INTO R2
4407 015726 012601 MOV (SP)+, R1 ;; POP STACK INTO R1
4408 015730 012600 MOV (SP)+, R0 ;; POP STACK INTO R0
4409 015732 012737 015610 000024 MOV $PWRDN, @#PWRVEC ;; SET UP THE POWER DOWN VECTOR
4410 015740 012737 000340 000026 MOV #340, @#PWRVEC+2 ;; PRI0:7
4411 015746 104401 TYPE $PWRMG: .WORD $POWER ;; REPORT THE POWER FAILURE
4412 015750 016000 MOV (PC)+, (SP) ;; POWER FAIL MESSAGE POINTER
4413 015752 012716 $PWRAD: .WORD $LOOP ;; RESTART AT $LOOP
4414 015754 015056 BIC #20, 2(SP) ;; RESTART ADDRESS
4415 015756 042766 000020 000002 CLR $TBIT ;; CLEAR "T" BIT
4416 015764 005067 177072 RTI ;; CLEAR THE "T" BIT FLAG
4417 015770 000002 $SILLUP: HALT
4418 015772 000000 BR -2 ;; THE POWER UP SEQUENCE WAS STARTED
4419 015774 000776 $SAVR6: 0 ;; BEFORE THE POWER DOWN WAS COMPLETE
4420 015776 000000 $POWER: .ASCIZ <15><12>"POWER" ;; PUT THE SP HERE
4421 016000 005015 047520 042527
4422 016006 000122
4423
4424

```



```

4479
4480      .SBTTL  TYPE ROUTINE
4481
4482      ;*****
4483      ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4484      ;*
4485      ;*CALL:
4486      ;*1) USING A TRAP INSTRUCTION
4487      ;*      TYPE      ,MESADR      ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4488      ;*OR
4489      ;*      TYPE
4490      ;*      MESADR
4491      ;*
4492      ;*****
4493
4494
4495 016204 010046      $TYPE:  MOV      RD, -(SP)      ;SAVE RD
4496 016206 017600 000002      MOV      @2(SP), RD      ;GET ADDRESS OF ASCIZ STRING
4497 016212 132767 000040 162367      BITB      #APCSUP, $ENVM      ;IS CONSOLE SUPPRESSED?
4498 016220 001003      BNE      60$      ;YES, SKIP TYPE OUT
4499 016222 112046      2$:      MOVB      (RD)+, -(SP)      ;PUSH CHARACTER TO BE TYPED ONTO STACK
4500 016224 001005      BNE      4$      ;BR IF IT ISN'T THE TERMINATOR
4501 016226 005726      TST      (SP)+      ;IF TERMINATOR POP IT OFF THE STACK
4502 016230 012600      60$:      MOV      (SP)+, RD      ;RESTORE RD
4503 016232 062716 000002      3$:      ADD      #2, (SP)      ;ADJUST RETURN PC
4504 016236 000002      RTI      ;RETURN
4505 016240 105777 162304      4$:      TSTB      @$TPS      ;WAIT UNTIL PRINTER IS READY
4506 016244 100375      BPL      4$
4507 016246 112677 162300      MOVB      (SP)+, @$TPB      ;LOAD CHAR TO BE TYPED INTO DATA REG.
4508 016252 000763      BR      2$

```


.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 108
 DVKAIA.P11 22-DEC-76 10:47 TYPE ROUTINE

SEQ 0109

```

4509
4510      .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
4511
4512      ;*****
4513      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
4514      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
4515      ;*CALL:
4516      ;*      MOV      NUM, -(SP)      ;NUMBER TO BE TYPED
4517      ;*      JSR      PC, TYPOCT      ;CALL FOR TYPEOUT
4518
4519      ;*****
4520
4521      016254  112767  000005  000070  TYPOCT:  MOVB      #5, $OCNT      ;SET ITERATION COUNT
4522      016262  010446                      MOV      R4, -(SP)      ;SAVE R4
4523      016264  010546                      MOV      R5, -(SP)      ;SAVE R5
4524      016266  016605  000006                      MOV      6(SP), R5      ;PICKUP THE INPUT NUMBER
4525      016272  005004                      CLR      R4      ;CLEAR THE OUTPUT WORD
4526      016274  006105                      1$:  ROL      R5      ;ROTATE MSB INTO "C"
4527      016276  000404                      BR       3$      ;GO DO MSB
4528      016300  006105                      2$:  ROL      R5      ;FORM THIS DIGIT
4529      016302  006105                      ROL      R5
4530      016304  006105                      ROL      R5
4531      016306  010504                      MOV      R5, R4
4532      016310  006104                      3$:  ROL      R4      ;GET LSB OF THIS DIGIT
4533      016312  042704  177770                      BIC      #177770, R4      ;GET RID OF JUNK
4534      016316  052704  000060                      BIS      #0, R4      ;MAKE THIS DIGIT ASCII
4535      016322  110467  000022                      MOVB      R4, 8$      ;SAVE FOR TYPING
4536      016326  104401  016350                      TYPE      8$      ;GO TYPE THIS DIGIT
4537      016332  105367  000014                      7$:  DECB      $OCNT      ;COUNT BY 1
4538      016336  002360                      BGE      2$      ;BR IF MORE TO DO
4539      016340  012605                      6$:  MOV      (SP)+, R5      ;RESTORE R5
4540      016342  012604                      MOV      (SP)+, R4      ;RESTORE R4
4541      016344  012616                      MOV      (SP)+, (SP)      ;SET THE STACK FOR RETURNING
4542      016346  000207                      RTS      PC      ;RETURN
4543      016350  000                      8$:  .BYTE      0      ;STORAGE FOR ASCII DIGIT
4544      016351  000                      .BYTE      0      ;TERMINATOR FOR TYPE ROUTINE
4545      016352  000000                      $OCNT:  .WORD      0      ;OCTAL DIGIT COUNTER
4546

```

```

4547
4548
4549
4550
4551
4552
4553
4554
4555
4556 016354 010046
4557 016356 016600 000002
4558 016362 005740
4559 016364 111000
4560 016366 006300
4561 016370 016000 016410
4562 016374 000200
4563
4564
4565
4566
4567 016376 011646
4568 016400 016666 000004 000002
4569 016406 000002
4570
4571
4572
4573
4574
4575
4576
4577
4578 016410 016376
4579 016412 016204
4580
4581
4582
4583 016414 005015 042115 030455
4584 016422 026461 053104 040513
4585 016430 026511 000101
4586
4587
4588 016434 000200
4589 017034 000200
4590 000001

.SBTTL TRAP DECODER

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

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

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

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

.SBTTL TRAP TABLE

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

; ROUTINE
;-----
$TRPAD: .WORD $TRAP2
        $TYPE ;;CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE

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

BUF1: .BLKW 200
BUF2: .BLKW 200
.END

```


MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 113
 DVKRIA.P11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0113

CALF = 000200	368#													
CCISP = 177570	374#	572												
CCISPLD = 000542	572#	714*	4372*	4444*										
CCISPLM = 000174	488#	714												
CCAC = 000652	652#	1335	2340	2487	2660	2737	2812	2901	4253	4264*	4270			
DSTLN = 000650	651#	824	983	1017	1123	1234	1308	1338	1451	1523	1551	1712	1787	
	1815	1817	1913	1998	2026	2028	2126	2255	2336	2514	2661	2833	2902	
	4252	4263*	4273											
DSWR = 177570	373#	571												
EMTVEC = 000030	462#	686*	687*											
ENDMSG = 015070	4181	4214#												
ENDT1 = 001550	1551	1565#												
ENDT10 = 003714	1721	1725#												
ENDT12 = 004374	1827	1831#												
ENDT13 = 004642	1935	1939#												
ENDT14 = 005106	2038	2042#												
ENDT15 = 005352	2148	2152#												
ENDT16 = 005626	2277	2281#												
ENDT17 = 006206	927	931#												
ENDT2 = 001764	1027	1031#												
ENDT3 = 002210	1135	1139#												
ENDT4 = 002454	1246	1250#												
ENDT5 = 002730	1460	1464#												
ENDT7 = 003460	455#	4344	4345*	4347*	4350*									
EMRVEC = 000004	661#	773	871	964	1063	1175	1283	1391	1498	1651	1762	1864	1973	
EXPPSW = 000674	2077	2205	2313	2387	2461	2551	2626	2710	2785	2872	3000	3073	3141	
	3209	3276	3346	3418	3491	3557	3636	3708	3791	3874	3950	4022	4115	
	4291*	4295*												
FILL = 000654	653#	807	825	904	979	1096	1126	1208	1237	1298	1424	1453	1513	
	1685	1714	1777	1897	1916	1988	2110	2129	2239	2258	2345	2419	2476	
	2585	2669	2742	2800	2908	3034	3088	3099	3156	3167	3224	3235	3361	
	3372	3433	3444	3478	3506	3517	3612	3651	3662	3723	3736	3796	3809	
	3889	3902	3965	3978	4037	4050	4130	4143	4254	4265*				
GENCON = 015276	4284#	4286												
GENR = 015146	765	864	957	1055	1167	1276	1383	1491	1586	1641	1755	1856	1966	
	2069	2194	2305	2380	2454	2543	2618	2702	2777	2864	2936	2998	3066	
	3134	3201	3268	3336	3411	3483	3549	3625	3700	3771	3861	3941	4013	
	4103	4250#												
GENSRC = 015266	1162	1271	1378	1496	1951	2064	2446	2771	2859	4281#				
GNS = ***** U	487	4579												
HT = 000011	365#													
ILLTRP = 015126	496	4235#												
INTR = 015356	1633	2186	2990	3328	3617	3853	4095	4314#						
ICOTRP = 015136	4242#													
ICOTVEC = 000020	460#	684*	685*											
LF = 000012	366#													
LCC = 011646	3327	3340#												
LOCC = 076040	479#	3069	3137	3205	3272	3340								
MC = 004160	1632	1645#												
MOVOC = 076030	479#	769	867	960	1059	1171	1279	1387	1494	1645				
MOVRC = 076031	479#	1758	1860	1969	2073	2199								
MRC = 005774	2185	2199#												
N = 000007	475#	748#	775	776#	783	784#	790	791#	796	797#	800	801#	804	
	805#	909	810#	813	814#	820	821#	824	827	828#	835	836#	843#	
	873	874#	881	882#	888	889#	893	894#	897	898#	901	902#	906	

907#	910	911#	917	918#	923	924#	928	929#	936#	966	967#	974
975#	981	982#	985	986#	992	993#	997	998#	1001	1002#	1005	1006#
1012	1013#	1017	1020	1021#	1028	1029#	1035#	1065	1066#	1073	1074#	1080
1081#	1085	1086#	1089	1090#	1093	1094#	1098	1099#	1102	1103#	1109	1110#
1114	1117	1118#	1125	1128	1129#	1136	1137#	1141#	1177	1178#	1185	1186#
1192	1193#	1197	1198#	1201	1202#	1205	1206#	1210	1211#	1214	1215#	1220
1221#	1225	1228	1229#	1236	1239	1240#	1247	1248#	1253#	1285	1286#	1293
1294#	1300	1301#	1304	1305#	1311	1312#	1316	1317#	1320	1321#	1324	1325#
1332	1333#	1341	1342#	1351	1352#	1358#	1393	1394#	1401	1402#	1408	1409#
1413	1414#	1417	1418#	1421	1422#	1426	1427#	1430	1431#	1437	1438#	1442
1445	1446#	1455	1456#	1461	1462#	1468#	1500	1501#	1508	1509#	1515	1516#
1519	1520#	1526	1527#	1531	1532#	1535	1536#	1539	1540#	1546	1547#	1551
1554	1555#	1562	1563#	1569#	1591	1592#	1604#	1653	1654#	1662	1663#	1669
1670#	1674	1675#	1676	1679#	1682	1683#	1687	1688#	1691	1692#	1698	1699#
1703	1706	1707#	1716	1717#	1722	1723#	1733#	1764	1765#	1772	1773#	1779
1780#	1783	1784#	1790	1791#	1795	1796#	1799	1800#	1803	1804#	1810	1811#
1817	1820	1821#	1828	1829#	1835#	1866	1867#	1874	1875#	1881	1882#	1886
1887#	1890	1891#	1894	1895#	1899	1900#	1903	1904#	1909	1910#	1915	1919
1919#	1925	1928	1929#	1936	1937#	1942#	1975	1976#	1983	1984#	1990	1991#
1994	1995#	2001	2002#	2006	2007#	2010	2011#	2014	2015#	2021	2022#	2028
2031	2032#	2039	2040#	2046#	2079	2080#	2087	2088#	2094	2095#	2099	2100#
2103	2104#	2107	2108#	2112	2113#	2116	2117#	2122	2123#	2128	2131	2132#
2138	2141	2142#	2149	2150#	2157	2207	2208#	2216	2217#	2223	2224#	2229
2229#	2232	2233#	2236	2237#	2241	2242#	2245	2246#	2251	2252#	2257	2260
2261#	2267	2270	2271#	2278	2279#	2287	2315	2316#	2323	2324#	2330	2331#
2334	2335#	2338	2339#	2342	2343#	2347	2348#	2351	2352#	2357	2389	2390#
2397	2398#	2404	2405#	2408	2409#	2412	2413#	2416	2417#	2421	2422#	2425
2426#	2430#	2463	2464#	2471	2472#	2478	2479#	2482	2483#	2496	2497#	2502
2503#	2510	2511#	2518	2519#	2524#	2553	2554#	2561	2562#	2568	2569#	2572
2573#	2578	2579#	2582	2583#	2587	2588#	2591	2592#	2597#	2628	2629#	2636
2637#	2643	2644#	2650	2651#	2656	2657#	2664	2665#	2671	2672#	2675	2676#
2681#	2712	2713#	2720	2721#	2727	2728#	2731	2732#	2735	2736#	2739	2740#
2744	2745#	2748	2749#	2754#	2787	2788#	2795	2796#	2802	2803#	2806	2807#
2821	2822#	2825	2826#	2831	2832#	2837	2838#	2842	2874	2875#	2882	2883#
2889	2890#	2895	2896#	2899	2900#	2905	2906#	2910	2911#	2914	2915#	2919
2941	2942#	2954#	3002	3003#	3010	3011#	3017	3018#	3021	3022#	3027	3028#
3031	3032#	3036	3037#	3040	3041#	3048#	3075	3076#	3083	3084#	3090	3091#
3094	3095#	3106	3107#	3110	3111#	3115#	3143	3144#	3151	3152#	3159	3159#
3162	3163#	3174	3175#	3178	3179#	3184#	3211	3212#	3219	3220#	3226	3227#
3230	3231#	3242	3243#	3246	3247#	3252#	3278	3279#	3286	3287#	3293	3294#
3297	3298#	3303#	3348	3349#	3356	3357#	3363	3364#	3367	3368#	3379	3380#
3383	3384#	3392#	3420	3421#	3428	3429#	3435	3436#	3439	3440#	3451	3452#
3455	3456#	3461#	3493	3494#	3501	3502#	3508	3509#	3512	3513#	3524	3525#
3528	3529#	3533#	3559	3560#	3567	3568#	3574	3575#	3578	3579#	3584#	3638
3639#	3646	3647#	3653	3654#	3657	3658#	3669	3670#	3673	3674#	3682#	3710
3711#	3718	3719#	3725	3726#	3729	3730#	3743	3744#	3747	3748#	3753#	3783
3784#	3791	3792#	3798	3799#	3802	3803#	3816	3817#	3820	3821#	3826#	3876
3877#	3884	3885#	3891	3892#	3895	3896#	3909	3910#	3913	3914#	3921#	3952
3953#	3960	3961#	3967	3968#	3971	3972#	3985	3986#	3991	3990#	3995#	4024
4025#	4032	4033#	4039	4040#	4043	4044#	4057	4059#	4061	4062#	4067#	4117
4118#	4125	4126#	4132	4133#	4136	4137#	4150	4151#	4154	4155#		
723	4583#											
476#	758	854	2299	2301	2371	2611	2695	3061	3062	3129	3130	3196
3197	3262	3263	3264	3324	3325	3406	3407	3473	3474	3543	3544	3545
3606	3607	3694	3695	3765	3766	3848	3849	3935	3936	4007	4008	4090
4091												

NAME = 016414
NXM = 177777

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 115
DYKRAIA.P11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- USER SYMBOLS

OLDPC	=	000702	664#	4221*	4228*	4235*	4242*										
ONEBYT		000714	670#														
ONES		000712	669#	851													
PATGEN		001300	722	735#													
PCI		000670	659#	1632*	2185*	2979*	3327*	3616*	3852*	4094*	4315						
PIRQ =		177772	372#														
PIRQE=		000240	466#														
PREP		015212	756	852	948	1046	1155	1265	1371	1479	1577	1625	1746	1847	1954		
			2057	2178	2297	2369	2441	2535	2607	2691	2765	2851	2927	2973	3059		
			3126	3193	3260	3321	3403	3470	3541	3603	3691	3762	3845	3932	4004		
			4097	4261#													
			389#														
PRO	=	000000	390#														
PR1	=	000040	391#														
PR2	=	000100	392#														
PR3	=	000140	393#														
PR4	=	000200	394#														
PR5	=	000240	395#														
PR6	=	000300	396#														
PR7	=	000340	370#	370													
PS	=	177776	461#	690*	691*	4380*	4391*	4390*	4396*	4409*	4410*						
PSW	=	177776	2988#	2998													
PWRVEC=		000024	3336#	3344													
REPCM		010474	1641#	1649													
REPLOC		011634	2194#	2203													
REPMC		004150	3861#	3872													
REPMRC		005760	3625#	3634													
REPSCN		013574	4103#	4113													
REPSKP		012660	1586#	1598													
REPSPN		014530	2936#	2948													
REP11		003756	456#	700	701*	708*											
REP30		010306	4304	4309#													
RESVEC=		000010	662#	780	878	971	1070	1182	1290	1398	1505	1659	1769	1871	1980		
RETN		015354	2084	2195*	2213	2320	2394	2468	2558	2633	2717	2792	2879	2989*	3007		
SAVR6		000676	3080	3148	3216	3283	3337*	3353	3425	3498	3564	3626*	3643	3715	3789		
			3862*	3881	3957	4029	4104*	4122	4256*	4257*							
			479#	3704	3776	3867											
SCANC =		076042	3852	3867#													
SCN		013614	4316														

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 118

DVKAIA.P11 22-DEC-76 10:47

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0118

T4E13	002432		1126#	1133															
T42CON	012544		3597	3599	3602#														
T45CON	013476		3839	3841	3844#														
T5E12	002662		1226#	1233															
T5E13	002706		1237#	1244															
T7E12	003414		1443#	1450															
XPSW	015306		763	862	955	1053	1165	1274	1381	1489	1637	1753	1854	1964	2067				
			2190	2303	2378	2452	2541	2616	2700	2775	2862	2984	3064	3132	3199				
			3266	3332	3409	3481	3547	3621	3698	3769	3857	3939	4011	4099	4291#				
\$APTHD	000400		529	535#															
\$AUTOB	000534		568#																
\$BASE	000642		630#	725															
\$BDADR	000522		563#																
\$BDDAT	000526		565#																
\$CKSWR=	***** U		4582																
\$CLR.T	015000		4185	4188#															
\$CMTAG	000500		551#	678															
\$CM3 =	000000		581#																
\$CPUOP	000614		604#																
\$CRLF	000563		583#	4451	4455														
\$DEVCT	000576		595#																
\$DOAGN	015020		4177	4183	4190	4196#													
\$ENDAD	015010		516	721	4192#	4474													
\$ENDCT	014752		692	4179#															
\$ENULL	015064		4212#																
\$ENV	000606		600#	1620	2173	2968	3316	3598	3840	4082	4457								
\$ENVM	000607		601#	717	4497														
\$EOP	014722		4079	4085	4170#														
\$EOPCT	014744		692*	4176#	4180														
\$ERFLG	000503		554#	4330	4359	4365*	4375	4442*											
\$ERMAX	000515		560#	694*	4371*	4375													
\$ERROR	016010		686	4441#															
\$ERRPC	000516		561#	4446*	4447*	4448	4452												
\$ERRTB	000644		647#																
\$ERTTL	000512		558#	4445*															
\$ESCAP	000560		581#	693*	4370*	4470	4472												
\$ETABL	000606		599#																

[illegible]

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 120

DVKAIA.P11 22-DEC-76 10:47

CROSS REFERENCE TABLE -- USER SYMBOLS

```
BTKB      000546
BTKS      000544
BTN       = 000051
```

574#												
573#												
474#	748	756#	775	783	790	796	800	804	809	813	920	924
827	833	835	842	851#	873	881	888	893	897	901	906	910
917	923	925	928	936	948#	966	974	981	985	992	997	1001
1005	1012	1017	1020	1026	1028	1035	1046#	1065	1073	1080	1085	1089
1093	1098	1102	1109	1114	1117	1125	1128	1134	1136	1141	1155#	1177
1185	1192	1197	1201	1205	1210	1214	1220	1225	1228	1236	1239	1245
1247	1253	1265#	1285	1293	1300	1304	1311	1316	1320	1324	1332	1341
1351	1358	1371#	1393	1401	1408	1413	1417	1421	1426	1430	1437	1442
1445	1455	1459	1461	1468	1479#	1500	1508	1515	1519	1526	1531	1535
1539	1546	1551	1554	1560	1562	1569	1577#	1591	1604	1616#	1653	1662
1669	1674	1678	1682	1687	1691	1698	1703	1706	1716	1720	1722	1733
1745#	1764	1772	1779	1783	1790	1795	1799	1803	1810	1817	1820	1826
1828	1835	1847#	1866	1874	1881	1886	1890	1894	1899	1903	1909	1915
1918	1925	1928	1934	1936	1942	1954#	1975	1983	1990	1994	2001	2006
2010	2014	2021	2028	2031	2037	2039	2046	2057#	2079	2087	2094	2099
2103	2107	2112	2116	2122	2128	2131	2138	2141	2147	2149	2157	2169#
2207	2216	2223	2228	2232	2236	2241	2245	2251	2257	2260	2270	2276
2278	2287	2297#	2315	2323	2330	2334	2338	2342	2347	2351	2357	2368#
2389	2397	2404	2408	2412	2416	2421	2425	2430	2441#	2463	2471	2478
2482	2496	2502	2510	2518	2524	2535#	2553	2561	2568	2572	2578	2582
2587	2591	2597	2606#	2628	2636	2643	2650	2656	2664	2671	2675	2681
2691#	2712	2720	2727	2731	2735	2739	2744	2748	2754	2765#	2787	2795
2802	2806	2821	2825	2831	2837	2842	2851#	2874	2882	2889	2895	2899
2905	2910	2914	2919	2927#	2941	2954	2964#	3002	3010	3017	3021	3027
3031	3036	3040	3048	3058#	3075	3083	3090	3094	3106	3110	3115	3126#
3143	3151	3158	3162	3174	3178	3184	3193#	3211	3219	3226	3230	3242
3246	3252	3260#	3278	3286	3293	3297	3303	3312#	3348	3356	3363	3367
3379	3383	3392	3403#	3420	3428	3435	3439	3451	3455	3461	3470#	3493
3501	3508	3512	3524	3528	3533	3541#	3559	3567	3574	3578	3584	3594#
3638	3646	3653	3657	3669	3673	3682	3691#	3710	3718	3725	3729	3743
3747	3753	3762#	3783	3791	3798	3802	3816	3820	3826	3836#	3876	3884
3891	3895	3909	3913	3921	3932#	3952	3960	3967	3971	3985	3989	3995
4004#	4024	4032	4039	4043	4057	4061	4067	4078#	4117	412		

```

$TPB 000552
$TPFLG 000557
$TPS 000550
$TRAP 016354
$TRAP2 016376
$TRP = 000002
$TRPAD 016410
$TSTM 000404
$TSTM 000502
$STYPBN= ***** U
$STYDPS= ***** U
$STYPE 016204
$STYPOC= ***** U
$SUNIT 000600
$SUNIT 000410
$SUSWR 000612
$VECT1 000636
$VECT2 000640
$XTSTR 015422
$$GET4= 000001

```


.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 121

DVKAIA.P11 22-DEC-76 10:47

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0121

S40CAT= ***** U
= 017434

4339													
479#	483#	487#	491#	501#	506#	509#	514	515#	517#	519#	525	526#	
528#	530#	550#	595	681	710	711	826	1019	1116	1127	1227	1238	
1444	1553	1705	1819	1917	1927	2030	2130	2140	2259	2269	4211	4213#	
4217#	4375	4392	4419	4588#	4589#								
525#	530												

.SX = 000400

CATCH	330#	4221	4228	4235	4242										
COMMEN	467#														
CPREP	333#	2297	2369	2441	2535	2607	2691	2765	2851	2973					
EHLT	338#	775	783	790	796	800	804	809	813	820	827	835	873	881	888
	893	897	901	906	910	917	923	928	966	974	981	985	992	997	1001
	1005	1012	1020	1028	1065	1073	1080	1085	1089	1093	1098	1102	1109	1117	1128
	1136	1177	1185	1192	1197	1201	1205	1210	1214	1220	1228	1239	1247	1285	1293
	1300	1304	1311	1316	1320	1324	1332	1341	1351	1393	1401	1408	1413	1417	1421
	1426	1430	1437	1445	1455	1461	1500	1508	1515	1519	1526	1531	1535	1539	1546
	1554	1562	1591	1653	1662	1669	1674	1678	1682	1687	1691	1698	1706	1716	1722
	1764	1772	1779	1783	1790	1795	1799	1803	1810	1820	1828	1866	1874	1881	1886
	1890	1894	1899	1903	1909	1918	1928	1936	1975	1983	1990	1994	2001	2006	2010
	2014	2021	2031	2039	2079	2087	2094	2099	2103	2107	2112	2116	2122	2131	2141
	2149	2207	2216	2223	2228	2232	2236	2241	2245	2251	2260	2270	2278	2315	2323
	2330	2334	2338	2342	2347	2351	2389	2397	2404	2408	2412	2416	2421	2425	2463
	2471	2478	2482	2496	2502	2510	2518	2553	2561	2568	2572	2578	2582	2587	2591
	2628	2636	2643	2650	2656	2664	2671	2675	2712	2720	2727	2731	2735	2739	2744
	2748	2787	2795	2802	2806	2821	2825	2831	2837	2874	2882	2889	2895	2899	2905
	2910	2914	2941	3002	3010	3017	3021	3027	3031	3036	3040	3075	3083	3090	3094
	3106	3110	3143	3151	3158	3162	3174	3178	3211	3219	3226	3230	3242	3246	3278
	3286	3293	3297	3348	3356	3363	3367	3379	3383	3420	3428	3435	3439	3451	3455
	3493	3501	3508	3512	3524	3528	3559	3567	3574	3578	3638	3646	3653	3657	3669
	3673	3710	3718	3725	3729	3743	3747	3783	3791	3798	3802	3816	3820	3876	3884
	3891	3895	3909	3913	3952	3960	3967	3971	3985	3989	4024	4032	4039	4043	4057
	4061	4117	4125	4132	4136	4150	4154								
ENDCOM	467#														
ENDPAS	336#	4181													
ERR	339#	775	783	790	796	800	804	809	813	820	827	835	873	881	888
	893	897	901	906	910	917	923	928	966	974	981	985	992	997	1001
	1005	1012	1020	1028	1065	1073	1080	1085	1089	1093	1098	1102	1109	1117	1128
	1136	1177	1185	1192	1197	1201	1205	1210	1214	1220	1228	1239	1247	1285	1293
	1300	1304	1311	1316	1320	1324	1332	1341	1351	1393	1401	1408	1413	1417	1421
	1426	1430	1437	1445	1455	1461	1500	1508	1515	1519	1526	1531	1535	1539	1546
	1554	1562	1591	1653	1662	1669	1674	1678	1682	1687	1691	1698	1706	1716	1722
	1764	1772	1779	1783	1790	1795	1799	1803	1810	1820	1828	1866	1874	1881	1886
	1890	1894	1899	1903	1909	1918	1928	1936	1975	1983	1990	1994	2001	2006	2010
	2014	2021	2031	2039	2079	2087	2094	2099	2103	2107	2112	2116	2122	2131	2141
	2149	2207	2216	2223	2228	2232	2236	2241	2245	2251	2260	2270	2278	2315	2323
	2330	2334	2338	2342	2347	2351	2389	2397	2404	2408	2412	2416	2421	2425	2463
	2471	2478	2482	2496	2502	2510	2518	2553	2561	2568	2572	2578	2582	2587	2591
	2628	2636	2643	2650	2656	2664	2671	2675	2712	2720	2727	2731	2735	2739	2744
	2748	2787	2795	2802	2806	2821	2825	2831	2837	2874	2882	2889	2895	2899	2905
	2910	2914	2941	3002	3010	3017	3021	3027	3031	3036	3040	3075	3083	3090	3094
	3106	3110	3143	3151	3158	3162	3174	3178	3211	3219	3226	3230	3242	3246	3278
	3286	3293	3297	3348	3356	3363	3367	3379	3383	3420	3428	3435	3439	3451	3455
	3493	3501	3508	3512	3524	3528	3559	3567	3574	3578	3638	3646	3653	3657	3669
	3673	3710	3718	3725	3729	3743	3747	3783	3791	3798	3802	3816	3820	3876	3884
	3891	3895	3909	3913	3952	3960	3967	3971	3985	3989	4024	4032	4039	4043	4057
	4061	4117	4125	4132	4136	4150	4154								
ERROR	361#	775	783	790	796	800	804	809	813	820	827	835	873	881	888
	893	897	901	906	910	917	923	928	966	974	981	985	992	997	1001
	1005	1012	1020	1028	1065	1073	1080	1085	1089	1093	1098	1102	1109	1117	1128
	1136	1177	1185	1192	1197	1201	1205	1210	1214	1220	1228	1239	1247	1285	1293
	1300	1304	1311	1316	1320	1324	1332	1341	1351	1393	1401	1408	1413	1417	1421
	1426	1430	1437	1445	1455	1461	1500	1508	1515	1519	1526	1531	1535	1539	1546
	1554	1562	1591	1653	1662	1669	1674	1678	1682	1687	1691	1698	1706	1716	1722

MSG7	1358#	1360													
MSG7A	1466#	1470													
MSG7B	1569#	1571													
MSG8	1603#	1606													
MSG9	1732#	1735													
MULT	467#														
NEWST	467#	748	842	936	1035	1141	1253	1358	1468	1569	1604	1733	1835	1942	2046
	2157	2287	2357	2430	2524	2597	2681	2754	2842	2919	2954	3048	3115	3184	3252
	3303	3392	3461	3533	3594	3682	3753	3826	3921	3995	4067				
POP	323#	467#	4402	4403											
PREP	332#	756	852	948	1046	1155	1265	1371	1479	1577	1625	1746	1847	1954	2057
	2178	2927	4086												
PUSH	323#	467#	4392	4388											
REPORT	467#														
RESETN	328#	748	842	936	1035	1141	1253	1358	1468	1569	1604	1733	1835	1942	2046
	2157	2287	2357	2430	2524	2597	2681	2754	2842	2919	2954	3048	3115	3184	3252
	3303	3392	3461	3533	3594	3682	3753	3826	3921	3995	4067				
R23	350#														
R45	342#	807	904	979	1096	1208	1298	1424	1513	1685	1777	1897	1988	2110	2239
	2344	2418	2476	2584	2668	2741	2800	2907	3033	3088	3156	3224	3361	3433	3506
	3651	3723	3796	3889	3965	4037	4130								
SCOPE	362#	755	850	947	1045	1154	1264	1370	1478	1576	1615	1744	1846	1953	2056
	2168	2296	2367	2440	2534	2605	2690	2764	2850	2926	2963	3057	3125	3192	3259
	3311	3402	3469	3540	3593	3690	3761	3835	3931	4003	4077	4171			
SETPRI	467#														
SETTRA	4571#														
SETUP	323#	467#													
SKIP	467#														
SKPINT	337#	1616	2169	2964	3312	3594	3836								
SLASH	467#														
SPACE	467#														
STARS	323#	467#	512	522	524	531	546	585	588	748	753	754	842	844	848
	849	936	938	945	946	1035	1037	1043	1044	1141	1143	1152	1153	1253	1255
	1262	1263	1358	1360	1368	1369	1468	1470	1476	1477	1569	1571	1574	1575	1604
	1606	1613	1614	1733	1735	1742	1743	1835	1838	1844	1845	1942	1944	1951	1952
	2046	2048	2054	2055	2157	2159	2166	2167	2287	2289	2294	2295	2357	2359	2365
	2366	2430	2432	2438	2439	2524	2526	2532	2533	2597	2599	2603	2604	2681	2683
	2688	2689	2754	2756	2762	2763	2842	2844	2848	2849	2919	2921	2924	2925	2954
	2956	2961	2962	3048	3050	3055	3056	3115	3117	3123	3124	3184	3186	3190	3191
	3252	3254	3257	3258	3303	3305	3309	3310	3392	3394	3400	3401	3461	3463	3467
	3468	3533	3535	3538	3539	3584	3586	3591	3592	3682	3684	3688	3689	3753	3755
	3759	3760	3826	3828	3833	3834	3921	3923	3929	3930	3995	3997	4001	4002	4067
	4069	4075	4076	4164	4327	4378	4394	4439	4493	4519	4550				
STPREP	334#	3058	3126	3193	3260	3321	3403	3470	3541	3603	3691	3762	3845	3932	4004
SWRSU	467#														
TGRZ	340#	794	891	995	1083	1195	1314	1411	1573	1672	1793	1884	2004	2097	2226
TRMTRP	4571#														
TRUNC	345#	1522	1786	1997											
TSTCC	344#	771	869	962	1061	1173	1281	1389	1496	1647	1760	1862	1971	2075	2201
	2311	2385	2459	2549	2624	2708	2783	2870	2996	3071	3139	3207	3274	3342	3416
	3489	3555	3632	3706	3779	3870	3948	4020	4111						
TSTSP	335#	780	878	971	1070	1182	1290	1398	1505	1659	1769	1871	1980	2094	2213
	2320	2394	2468	2558	2633	2717	2792	2879	3007	3080	3149	3216	3293	3353	3425
	3498	3564	3643	3715	3788	3881	3957	4029	4122						
TTRAP	355#	1577	2927												
TTY	349#	1632	2185	2979	3327	3616	3852	4094							

.MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 126
 DVKAIA.P11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- MACRO NAMES

SEG 0125

TYPBIN	467#														
TYPDEC	467#														
TYPNAM	467#														
TYPNUM	467#														
TYPDCS	467#														
TYPDCY	467#														
TYPDXT	467#														
UBYTE	347#	833	925	1026	1134	1245	1459	1560	1720	1826	1934	2037	2147	2276	
VDEST	336#	824	1017	1114	1125	1225	1236	1442	1551	1703	1817	1915	1925	2028	2129
	2138	2257	2267												
SSCMRE	544#														
SSCMTM	544#														
SSESCA	467#														
SSNEWT	467#	748	842	936	1035	1141	1253	1359	1469	1569	1604	1733	1835	1942	2046
	2157	2287	2357	2430	2524	2597	2681	2754	2842	2919	2954	3049	3115	3194	3252
	3303	3392	3461	3533	3584	3682	3753	3826	3921	3995	4067				
SSSET	4571#														
SSSKIP	467#														
SSQUAT	323#	357													
SSSETUP	323#	500													
SSACT1	323#	510													
SSAPTB	586#														
SSAPTH	323#	520													
SSCATC	323#	481													
SSCMTA	323#	544													
SSFCP	323#	4162													
SSPOWE	323#	4376													
SSSCOP	323#	4325													
SSTRAP	323#	4548													

. ABS. 017434 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DVKAIA,DVKAIA/SOL/CRF=DVKAIA.P11
 RUN-TIME: 62 52 5 SECONDS
 RUN-TIME RATIO: 745/121=6.1
 CORE USED: 21K (41 PAGES)