

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

RECEIVED 22-DEC-76 10:47

MAC222427(1006) 22-DEC-76 10:49 PAGE 0010VKAIASEG

00010000

773224
SEC 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, MOV, 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

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

2.0 OPERATING INSTRUCTIONS

2.1 LOADING PROCEDURES

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED TAPES

2.2 STARTING PROCEDURE

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

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

2.3 OPERATING PROCEDURES

2.3.1 OPERATIONAL SWITCH REGISTER

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

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

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

2.3.2 RUNNING UNDER APT

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

USING THE CONSOLE INTERFACE AS THE INTERRUPTING DEVICE,
 THE INTERRUPTABILITY TESTS WILL BE RUN ON ONLY THE FIRST PASS TO
 AVOID INTERFERENCE WITH THE APT INTERFACE. IF INTERRUPTABILITY
 TESTS ARE DESIRED ON ALL PASSES, ANOTHER SLL MUST BE SUPPLIED
 AND ITS RECEIVER STATUS REGISTER ADDRESS & ITS INTERRUPT VECTOR MUST
 BE PLACED IN THE APT E-TABLE AT LOCATIONS "\$BASE" & "\$VEC1" 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: RBLF 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*
 \$VEC1: *RECEIVER INTERRUPT VECTOR*

2.4 EXECUTION TIMES

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

3.0 ERROR REPORTING

 IF A ROUTINE FAILS AND THE INHIBIT ERROR 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 JDT.

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

WHERE "XXXXXX" IS AN OCTAL VALUE

4.0 SUBROUTINE ABSTRACTS

4.1 TRAPCATCHER

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

4.2 SCOPE

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

4.3 ERROR

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

4.4 \$POWER

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

4.5 NPREP

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

4.6 GENR

THIS ROUTINE IS USED TO TRANSFER INSTRUCTION TEST ARGUMENTS TO THE GENERAL REGISTERS AND TO COPY THE STACK POINTER BEFORE THE TEST INSTRUCTION EXECUTION.

4.7 XPSW

THIS ROUTINE IS USED TO STORED THE EXPECTED PSW OF THE INSTRUCTION TEST AND TO SET THE T-BIT IN THE EXPECTED PSW ON PASSES USING THE TRACE TRAP.

4.9 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 "SPASS" IS USED TO KEEP THE PASS COUNT. IT IS CLEARED BY STARTING AT 200.

5.3 TEST NUMBER

A 16 BIT LOCATION "STSTNM" IS USED TO KEEP TRACK OF THE SUBTEST NUMBER. THIS NUMBER IS ALSO PLACED IN THE APT E-TABLE AT "STESTN" 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

MACY11 27(1006) 22-DEC-76 10:49 PAGE 9
C.KAH.F11 22-DEC-76 10:47

SEG 00:0

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

MACY11 27(1006) 22-DEC-76 10:49 PAGE 9
C.KAH.F11 22-DEC-76 10:47

.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

```

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

MACY11 27(1006) 22-DEC-76 10:49 PAGE 12
BASIC DEFINITIONS

SEQ 0013

000200	APTSIZE=	200
000001	APTENV=	001
000100	APTSPool=	100
000040	APTCSUP=	040
171400	\$SWR=	171400
000300	\$SWRMK=	300
000020	TBIT=	20
000001	\$TN=	1
000001	N=	1
177777	NXM=	177777
177560	ABASE=	177560
000060	AVECT1=	60
000000	. =C	

NO1

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

SEQ 0014

```

480
481      .SETTL TRAP CATCHER
482
483      000000
484      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2.HALT"
485      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
486      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
487      ;*=174
488      000174 000000  DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
489      000176 000000  SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
490
491      000000 000000      ;*=0
492      000000 015106      TZERO      ;SET LOCATIONS 0,4,6 TO ERROR REPORT
493      000002 000340      340
494      000004 015116      TIMTRP
495      000006 000340      340
496      000010 015126      ILLTRP
497      000012 000340      340
498
499
500
501      000100 000100      ;*=100
502      000100 000102      .WORD 102      ;HANDLE EVENT LINE INTERRUPTS
503      000102 000002      .WORD 2
504
505
506      000200 000200      ;*=200
507      000203 000167 000512  JMP START      ;STARTING ADDRESS OF PROGRAM
  
```

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

SEQ 0015

```

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

```

C02

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

SEQ 0016

S43

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 16
 DVKATA.P11 22-DEC-76 10:47 COMMON TAGS

SEQ 0017

```

544 .SBTTL COMMON TAGS
545
546 ::*****
547 :*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
548 :*USED IN THE PROGRAM.
549
550 000500
551 000500 000500 $CMTAG: . =500 ::START OF COMMON TAGS
552 000500 000000 $TSTNM: .WORD 0 ::CONTAINS THE TEST NUMBER
553 000502 000 $ERFLG: .BYTE 0 ::CONTAINS ERROR FLAG
554 000503 000 $ICNT: .WORD 0 ::CONTAINS SUBTEST ITERATION COUNT
555 000504 000000 $LPADR: .WORD 0 ::CONTAINS SCOPE LOOP ADDRESS
556 000506 000000 $LPERR: .WORD 0 ::CONTAINS SCOPE RETURN FOR ERRORS
557 000510 000000 $ERTTL: .WORD 0 ::CONTAINS TOTAL ERRORS DETECTED
558 000512 000000 $ITEMB: .BYTE 0 ::CONTAINS ITEM CONTROL BYTE
559 000514 000 $ERMAX: .BYTE 1 ::CONTAINS MAX. ERRORS PER TEST
560 000515 001 $ERRPC: .WORD 0 ::CONTAINS PC OF LAST ERROR INSTRUCTION
561 000516 000000 $GDADR: .WORD 0 ::CONTAINS ADDRESS OF 'GOOD' DATA
562 000520 000000 $BDADR: .WORD 0 ::CONTAINS ADDRESS OF 'BAD' DATA
563 000522 000000 $GDDAT: .WORD 0 ::CONTAINS 'GOOD' DATA
564 000524 000000 $BDDAT: .WORD 0 ::CONTAINS 'BAD' DATA
565 000526 000000 ::RESERVED--NOT TO BE USED
566 000530 000000
567 000532 000000
568 000534 000 $AUTOB: .BYTE 0 ::AUTOMATIC MODE INDICATOR
569 000535 000 $INTAG: .BYTE 0 ::INTERRUPT MODE INDICATOR
570 000536 000000
571 000540 177570 $SWR: .WORD DSWR ::ADDRESS OF SWITCH REGISTER
572 000542 177570 $DISPLAY: .WORD DDISP ::ADDRESS OF DISPLAY REGISTER
573 000544 177560 $TKS: 177560 ::TTY KBD STATUS
574 000546 177562 $TKB: 177562 ::TTY KBD BUFFER
575 000550 177564 $TPS: 177564 ::TTY PRINTER STATUS REG. ADDRESS
576 000552 177566 $TPB: 177566 ::TTY PRINTER BUFFER REG. ADDRESS
577 000554 000 $NULL: .BYTE 0 ::CONTAINS NULL CHARACTER FOR FILLS
578 000555 002 $FILLS: .BYTE 2 ::CONTAINS # OF FILLER CHARACTERS REQUIRED
579 000556 012 $FILLC: .BYTE 12 ::INSERT FILL CHARS. AFTER A "LINE FEED"
580 000557 000 $TPFLG: .BYTE 0 ::"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
581 000560 000000 $ESCAPE: 0 ::ESCAPE ON ERROR ADDRESS
582 000562 077 $QUES: .ASCII /?/ ::QUESTION MARK
583 000563 015 $CRLF: .ASCII <15> ::CARRIAGE RETURN
584 000564 000012 $LF: .ASCII <12> ::LINE FEED
585 ::*****
586 .SBTTL APT MAILBOX-ETABLE
587
588 ::*****
589 .EVEN
590 $MAIL: ::APT MAILBOX
591 000566 000000 $MSGTY: .WORD AMSGTY ::MESSAGE TYPE CODE
592 000570 000000 $FATAL: .WORD AFATAL ::FATAL ERROR NUMBER
593 000572 000000 $TESTN: .WORD ATESTN ::TEST NUMBER
594 000574 000000 $PASS: .WORD APASS ::PASS COUNT
595 000576 000000 $DEVCT: .WORD ADEVCT ::DEVICE COUNT
596 000600 000000 $UNIT: .WORD AUNIT ::I/O UNIT NUMBER
597 000602 000000 $MSGAD: .WORD AMSGAD ::MESSAGE ADDRESS
598 000604 000000 $MSGLG: .WORD AMSLG ::MESSAGE LENGTH
599 $ETABLE: ::APT ENVIRONMENT TABLE

```

E02

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 17
 CVKATA.P11 22-DEC-76 10:49 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	000000	\$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		\$E*END:			
632			.MEXIT			

MAIN. MACY11 27(1005) 22-DEC-76 10:49 PAGE 19
 CVKRIA.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 INFORMATIC 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	:SOURCE LENGTH
SRCAD: .WORD	:SOURCE ADDRESS
DSTLN: .WORD	:DESTINATION LENGTH
DSTAD: .WORD	:DESTINATION ADDRESS
FILL: .WORD	:FILL CHARACTER
TABLE: .WORD 177777	:TRANSLATION TABLE ADDRESS
TCSR: 177564	:TCSR ADDRESS OF SLU USED FOR INTERRUPTS
TBUF: 177566	:TBUF ADDRESS
TVECT: 64	:TRANSMIT INTERRUPT VECTOR
TPSW: 66	: AND PSW LOCATION
PCI: 0	:ADDRESS OF TEST INSTRUCTION TO INTERRUPT
CCODES: 0	:CONDITION CODES AFTER TEST INSTRUCTION EXECUTION
EXPPSW: 0	:EXPECTED CONDITION CODES
SAVR6: 0	:STACK POINTER VALUE BEFORE TEST INSTRUCTION EXECUTION
BADR6: 0	:BAD STACK POINTER VALUE
OLDPC: 0	:PC WHERE UNEXPECTED TRAP OR INTERRUPT OCCURRED
TEMP: 0	
TEMP1: 0	
TEMP2: 0	
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
 SVKATA. =11 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,\$IOTVEC		:IOT VECTO FOR SCOPE ROUTINE		
685	000760	012737	000340	000022		MOV	\$340,\$IOTVEC+2		:LEVEL 7		
686	000766	012737	016010	000030		MOV	\$ERROR,\$EMTVEC		:EMT VECTOR FOR ERROR ROUTINE		
687	000774	012737	000340	000032		MOV	\$340,\$EMTVEC+2		:LEVEL 7		
688	001002	012737	016354	000034		MOV	\$TRAP,\$TRAPVEC		:TRAP VECTOR FOR TRAP CALLS		
689	001010	012737	000340	000036		MOV	\$340,\$TRAPVEC+2		:LEVEL 7		
690	001016	012737	015610	000024		MOV	\$PWDON,\$PWRVEC		:POWER FAILURE VECTOR		
691	001024	012737	000340	000026		MOV	\$340,\$PWRVEC+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,\$SERMAX		: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,\$TBITVEC		:SET "T" BIT VECTOR TO SRTN		
698	001060	012737	000340	000016		MOV	\$340,\$TBITVEC+2		:LEVEL 7		
699	001066	012767	000002	013760		MOV	\$RTI,\$SRTN		:SET SRTN TO A RTI		
700	001074	013767	000010	177602		MOV	\$RESVEC,TEMP		:SAVE ILLEGAL INSTRUCTION TRAP VECTOR		
701	001102	012737	001130	000010		MOV	\$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,\$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				BEG	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,\$42		:UNDER ACT11 AUTO-ACCEPT		
722	001226	001424				BEG	PATGEN		:BR IF YES, & SKIP PROGRAM ID TYPEOUT		
723	001230	104401	016414			TYPE,	NAME				
724									:SET UP ADDRESSES OF SLU TO USE FOR INTERRUPTABILITY TEST		
725	001234	013700	000642			MOV	\$BASE,\$R0		:GET ADDRESS OF THE SLU		
726	001240	062700	000004			ADD	\$4,\$R0		:ADJUST TO TCSR ADDRESS		
727	001244	010037	000660			MOV	\$PC,\$TCSR		:STORE TCSR ADDRESS		

H02

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

SEG 0021

728	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#SVECT1,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
 DVKATA.P11 22-DEC-76 10:49 INITIALIZE THE COMMON TAGS

SEC 0022

```

746
747
748
749
750
751
752
753
754
755 001334 000004
756 001336 004567 01365C
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
*SUCCESSFUL 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
      S ;DESTINATION LENGTH
      BUF2+1 ;DESTINATION ADDRESS
      377 ;FILL CHARACTER
      JSR PC,CLODST ;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
      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 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
      *****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
      *****TEST 1 - ERROR 3*****
      R0 SHOULD BE ZERO
      1$:
      TST R1 ;CHECK STATE OF OTHER GENERAL REGISTERS
      BEQ 66$ ;TEST R1
      EPROR 4 ;BR, IF ZERO
      *****TEST 1 - ERROR 4*****
      R1 SHOULD BE ZERO
      66$:
      TST R2 ;TEST R2
      BEQ 67$ ;BR IF ZERO
      EPROR 5 ;*****TEST 1 - ERROR 5*****
      R2 SHOULD BE ZERO

```

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

TEST "MOVC" INSTRUCTION WITH ZERO SOURCE LENGTH

SEG 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	177150		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:				

```

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  RS, PREP          ;SET UP INSTRUCTION ARGUMENTS
      1
      NXM                    ;SOURCE LENGTH
      0                      ;SOURCE ADDRESS
      BUF2+1                 ;DESTINATION LENGTH
      377                   ;DESTINATION ADDRESS
      377                   ;FILL CHARACTER
      377                   ;CLEAR DESTINATION
      377                   ;POINT RD TO LOWER BYTE BOUNDARY
      377                   ;CLEAR BOUNDARY & DEST. BYTES
      377                   ;CLEAR UPPER BYTE BOUNDARY
      377                   ;STORE EXPECTED PSW VALUE
      377                   ;SET UP GENERAL REGISTERS
      377                   ;SET ALL CONDITION CODES
      377                   ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      377                   ;CHECK RESULTS
      377                   ;STORE THE PSW
      377                   ;CLEAR ALL UNUSED BITS
      377                   ;CHECK PSW AGAINST EXPECTED VALUE
      377                   ;BR, IF EQUAL
      377                   *****TEST 2 - ERROR 1*****
      377                   ;PSW ERROR
      377                   ;EXPECTED PSW IS STORED AT "EXPPSW"
      377                   ;ACTJAL PSW IS STORED AT "CCODES"
      377                   ;VERIFY STACK POINTER IS RESTORED
      377                   ;BR IF OK
      377                   ;STORE BAD SP VALUE
      377                   *****TEST 2 - ERROR 2*****
      377                   ;STACK POINTER NOT RESTORED BY INSTRUCTION
      377                   ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      377                   ;ERRONEOUS VALUE IS AT "BADR6"
      377                   ;CHECK RD EQUAL TO SOURCE LENGTH
      377                   ;BR, IF EQUAL
      377                   *****TEST 2 - ERROR 3*****
      377                   ;RD SHOULD EQUAL SOURCE LENGTH
      377                   ;CHECK OTHER GENERAL REGISTERS
      377                   ;TEST R1
      377                   ;BR, IF ZERO
      377                   *****TEST 2 - ERROR 4*****
      377                   ;R1 SHOULD BE ZERO

```

SEQ 0025

Address	Hex	Assembly	Comment
895	001702	005702	
896	001704	001401	
897	001706	104005	
898			
899	001710	005703	
900	001712	001401	
901	001714	104006	
902			
903	001716		
904	001716	026704	176732
905	001722	001401	
906	001724	104007	
907			
908	001726	026705	176724
909	001732	001401	
910	001734	104010	
911			
912	001736		
913			
914	001736	012700	017034
915	001742	105720	
916	001744	001401	
917	001746	104011	
918			
919			
920	001750		
921	001750	105720	
922	001752	001401	
923	001754	104012	
924			
925	001756		
926	001756	105720	
927	001760	001401	
928	001762	104013	
929			
930			
931	001764		
932			

M02

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 25
 DVKATA.P11 22-DEC-76 10:47 T2 TEST "MOVC" INSTRUCTION WITH ZERO DESTINATION LENGTH

SEQ 0026

```

933
934
935
936
937
938
939
940
941
942
943
944
945
946
947 001764 000004
948 001766 004567 013220
949 001772 000020
950 001774 016434
951 001776 000011
952 002000 017035
953 002002 000377
954 002004 004767 013230
955 002010 004567 013272
956 002014 000200
957 002016 004767 013124
958 002022 000277
959
960 002024 076030
961
962 002026 106767 176640
963 002032 042767 177400 176632
964 002040 023767 000674 176624
965 002046 001401
966 002050 104001
967
968
969
970 002052
971 002052 023706 000676
972 002056 001403
973 002060 010637 000700
974 002064 104002
975
976
977
978 002066
979 002066 026704 176562
980 002072 001401
981 002074 104003
982
983 002076 026705 176554
984 002102 001401
985 002104 104004
986
987 002106
988 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,CLOST     ;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
      ;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"
84$:  CMP      @#SAVR6,SP    ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$         ;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

```

N02

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 26
 CVKRA.F11 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	RO,R5	
991	002120	001401		BEQ	2\$	:BR IF OK
992	002122	104005		ERROR	5	:*****TEST 3 - ERROR 5*****
993						:RO NOT EQUAL TO (SRCLN-DSTLN)
994	002124		2\$:			:CHECK OTHER GENERAL REGISTERS
995	002124	005701		TST	R1	:TEST R1
996	002126	001401		BEQ	68\$	:BR, IF ZERO
997	002130	104005		ERROR	6	:*****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	104005		ERROR	7	:*****TEST 3 - ERROR 7*****
1002						:R2 SHOULD BE ZERO
1003	002140	005703	69\$:	TST	R3	:TEST R3
1004	002142	001401		BEQ	70\$	:BR, IF ZERO
1005	002144	104005		ERROR	10	:*****TEST 3 - ERROR 10*****
1006						:R3 SHOULD BE ZERO
1007	002146		70\$:			:VERIFY DESTINATION AREA
1008						:POINT RO TO DESTINATION LOWER BYTE BOUNDARY
1009	002146	012700	017034	MOV	#BUF2,RO	:TEST CONTENTS OF BOUNDARY
1010	002152	005720		TSTB	(RO)+	:BR, IF STILL ZERO
1011	002154	001401		BEQ	71\$	:*****TEST 3 - ERROR 11*****
1012	002156	104011		ERROR	11	:LOWER BOUNDARY OF DESTINATION CHANGED
1013						: SHOULD STILL EQUAL ZERO
1014						
1015	002160		71\$:			:POINT R2 TO SOURCE STRING
1016	002160	016702	176462	MOV	SRCAD,R2	:STORE TRANSFER BYTE COUNT IN R1
1017	002164	016701	176460	MOV	DSTLN,R1	:CHECK CHARACTERS IN DESTINATION
1018	002170	122022		CMPB	(RO)+,(R2)+	:BR IF OK
1019	002172	001401		BEQ	+4	:*****TEST 3 - ERROR 12*****
1020	002174	104012		ERROR	12	:COMPARE ERROR IN DESTINATION
1021						:RO CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1022						
1023						
1024	002176	005301		DEC	R1	:DECREMENT BYTE COUNT
1025	002200	001373		BNE	T3E12	:BR, IF NOT FINISHED CHECKING
1026	002202	105720		TSTB	(RO)+	: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
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.
*RO-->R3 EQUAL TO ZERO, AND CONDITION CODES N,C=1
*AND Z,V=0
*****
*****
TST4:  SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
                        374  ;SOURCE LENGTH
                        BUF1  ;SOURCE ADDRESS
                        376  ;DESTINATION LENGTH
                        BUF2+1 ;DESTINATION ADDRESS
                        377  ;FILL CHARACTER
      JSR      PC,CLDST     ;CLEAR DESTINATION AREA & SET DEST. ADDRESS
      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD    211
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      +CLN!CLC              ;CLEAR CONDITION CODES N&C
      +SEV!SEZ              ;SET CONDITION CODES V&Z
                        ;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 4 - ERROR 1*****
                        ;PSW ERROR
                        ;EXPECTED PSW IS STORED AT "EXPPSW"
                        ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      @SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;BR IF OK
      MOV      SP,@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
      1$:
      TST      R1            ;CHECK OTHER GENERAL REGISTERS
      BEQ      66$          ;TEST R1
      ERROR    4            ;BR, IF ZERO
                        ;*****TEST 4 - ERROR 4*****
                        ;R1 SHOULD BE ZERO
      66$:
      TST      R2            ;TEST 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		BFQ	68\$		;BR, IF ZERO
1093	002342	104006		ERROR	6		*****TEST 4 - ERROR 6*****
1094							;R3 SHOULD BE ZERO
1095	002344			68\$: CMP	FILL,R4		;CHECK R4 UNCHANGED
1096	002344	026704	176304	BEQ	69\$		;BR IF OK
1097	002350	001401		ERROR	7		*****TEST 4 - ERROR 7*****
1098	002352	104007					;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1099							;CHECK R5 UNCHANGED
1100	002354	026705	176276	69\$: CMP	TABLE,R5		;BR IF OK
1101	002360	001401		BEQ	70\$		*****TEST 4 - ERROR 10*****
1102	002362	104010		ERROR	10		;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1103							
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	SRCAD,R1		;STORE TRANSFER BYTE COUNT IN R1
1115	002406	122022		74E12: 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	74E12		;BR, IF NOT FINISHED CHECKING
1123	002420	016705	176224	MOV	DSTAD,R5		;CALCULATE THE NUMBER OF "FILL"
1124	002424	166705	176214	SUB	SRCAD,R5		;CHARACTERS THAT SHOULD APPEAR IN DEST.
1125	002430	010501		MOV	R5,R1		;STORE TRANSFER BYTE COUNT IN R1
1126	002432	122067	176216	74E13: 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	74E13		;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							
1140	002454			ENDT4:			

```

1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154 002454 000004
1155 002456 004567 012533
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
*****
*****
TSTS:  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

```

E03

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

TEST "MOVC" INSTRUCTION WITH SRCAD=DSTAD, SL .LT. 3L

SEQ 003:

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	56\$:	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	59\$:	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	SRCIN,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	DSTIN,R2		:CALCULATE FILL LENGTH
1235	002700	166702	175740	SUB	SRCIN,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	ENDTS		: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			ENDTS:			

F03

MAIN MACY11 27:1006 22-DEC-76 10:49 PAGE 31
 C.K.A.H.A. P.11 22-DEC-76 10:49

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

SEQ 0022

```

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
*****
TEST6:  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
        ;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 5 - 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*BADP6      ;STORE BAD SP VALUE
        ERROR    2
        *****TEST 6 - ERROR 2*****
        STACK POINTER NOT RESTORED BY INSTRUCTION
        EXPECTED VALUE OF SP IS STORED AT "SAVR6"
        ERRONEOUS VALUE IS AT "BADP6"
        65$:
        CMP      FILL,R4         ;CHECK R4 UNCHANGED
        BEQ      66$             ;BR IF OK
        ERROR    3
        *****TEST 6 - ERROR 3*****
        R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
        CHECK R5 UNCHANGED
        BEQ      67$             ;BR IF OK
        ERROR    4
        *****TEST 6 - ERROR 4*****
        R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
        67$:

```

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 32
 SYKATA.P11 22-DEC-76 10:47 T6

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

SEC 0033

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	69\$:	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

H03

MAIN MACY11 27-DEC-76 22-DEC-76 10:49 PAGE 33
 SYKATA.F11 22-DEC-76 10:49

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

SEC 3034

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 011572
 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 001402
 003314 010627 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
      1          ;DESTINATION LENGTH
      BUF2+1      ;DESTINATION ADDRESS
      377        ;FILL CHARACTER
      JSR      PC,CLOST     ;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$          ;BR, IF EQUAL
      ERROR    1            ;*****TEST 7 - ERROR 1*****
                              ;PSW ERROR
                              ;EXPECTED PSW IS STORED AT "EXPPSW"
                              ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      2*SAVR6,SP    ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;BR IF OK
      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$:
      TST      R0            ;CHECK R0 FOR ZERO
      BEQ      1$          ;BR, IF ZERO
      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
 SUPP.A.P11 22-DEC-76 10:47 TEST "MOV" WITH DSTAD .GT. SRCAD, SRCLN .LT. DSTLN

SEG 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                                     ;
1424 003352 026704 175276 CMP      FILL,R4    ;CHECK R4 UNCHANGED
1425 003356 001401      BEQ      69$      ;BR IF OK
1426 003360 104007      ERROR      7      ;*****TEST 7 - ERROR 7*****
1427                                     ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
1428 003362 026705 175270 69$: CMP      TABLE,R5 ;CHECK R5 UNCHANGED
1429 003366 001401      BEQ      70$      ;BR IF OK
1430 003370 104010      ERROR      10     ;*****TEST 7 - ERROR 10*****
1431                                     ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
1432 003372                                     ;
1433                                     ;VERIFY DESTINATION
1434 003372 012700 017034 MOV      #BUF2,R0 ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1435 003376 105720      TSTB     (R0)+    ;TEST CONTENTS OF BOUNDARY
1436 003400 001401      BEQ      71$      ;BR, IF STILL ZERO
1437 003402 104011      ERROR      11     ;*****TEST 7 - ERROR 11*****
1438                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
1439                                     ; SHOULD STILL EQUAL ZERO
1440 003404                                     ;
1441 003404 012702 016434 71$: MOV      #BUF1,R2 ;POINT R2 TO ORIGINAL SOURCE BYTES
1442 003410 016701 175230 2$: MOV      SRCLN,R1 ;STORE TRANSFER BYTE COUNT IN R1
1443 003414 122022      *7E12: CMPB     (R0)+,(R2)+ ;CHECK CHARACTERS IN DESTINATION
1444 003416 001401      BEQ      12$      ;BR IF OK
1445 003420 104012      ERROR      12     ;*****TEST 7 - ERROR 12*****
1446                                     ;COMPARE ERROR IN DESTINATION
1447                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
1448 003422 005301      DEC      R1      ;DECREMENT BYTE COUNT
1449 003424 001373      BNE      17E12     ;BR, IF NOT FINISHED CHECKING
1450 003426 016705 175216 MOV      DSTLN,R5 ;CHECK FILL
1451 003432 166705 175206 SUB      SRCLN,R5
1452 003436 122067 175212 3$: CMPB     (R0)+,FILL
1453 003442 001401      BEQ      4$      ;
1454 003444 104013      ERROR      13     ;*****TEST 7 - ERROR 13*****
1455                                     ;"FILL" ERROR IN DESTINATION
1456 003446 105305      4$: DECB     R5      ;
1457 003450 001372      BNE      3$      ;
1458 003452 105720      TSTB     (R0)+    ;TEST CONTENTS OF DEST. UPPER BOUNDARY
1459 003454 001401      BEQ      END7     ;BR, IF STILL ZERO
1460 003456 104014      ERROR      14     ;*****TEST 7 - ERROR 14*****
1461                                     ;UPPER BOUNDARY OF DEST. CHANGED
1462                                     ; SHOULD STILL EQUAL ZERO
1463 003460      END7:

```

PC	Instruction	Source	Destination	Comment
003460	000004			TEST 10 TEST "MOV" WITH DSTAD .LT. SRCAD & SRCLEN .GT. DSTLEN. TRUNCATION.
003462	004567	011524		THIS TEST CHECKS FOR PROPER TRANSFER OF BYTES WHEN THE
003466	000020			SOURCE IS HIGHER IN MEMORY THAN THE DESTINATION AND THE TRANSFER
003470	017054			IS TRUNCATED (SRCLEN > DSTLEN).
003472	000007			THE RESULT IS A TRUNCATED SOURCE IN THE DESTINATION. R0-->R3
003474	017035			EQUAL ZERO, AND ALL CONDITION CODES CLEAR.
003476	000377			*****
003500	004767	011534		*****
003504	004567	011556		ST10: SCOPE
003510	000020			JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
003512	017054			20 ;SOURCE LENGTH
003514	004567	011566		BUF2+20 ;SOURCE ADDRESS
003520	000200			7 ;DESTINATION LENGTH
003522	004767	011420		BUF2+1 ;DESTINATION ADDRESS
003526	000277			377 ;FILL CHARACTER
003530	076030			PC,CLDST ;CLEAR DESTINATION AREA
003532	106767	175134		JSR R5,GENSRC ;GENERATE SOURCE STRING
003536	042767	177400	175126	.WORD 20
003544	023767	000674	175120	.WORD BUF2+20
003552	001401			JSR R5,XPSW ;STORE EXPECTED PSW VALUE
003554	104001			.WORD 200
				JSR PC,GENR ;SET UP GENERAL REGISTERS
				SCC ;SET ALL CONDITION CODES
				MOV ;EXECUTE "MOVE CHARACTER" INSTRUCTION
003532	106767	175134		MFPS CCODES ;STORE THE PSW
003536	042767	177400	175126	BIC #177400,CCODES ;CLEAR ALL UNUSED BITS
003544	023767	000674	175120	CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
003552	001401			BEG 64\$;BR, IF EQUAL
003554	104001			ERROR 1 ;*****TEST 10 - ERROR 1*****
				PSW ERROR
				EXPECTED PSW IS STORED AT "EXPPSW"
				ACTUAL PSW IS STORED AT "CCODES"
003556	023706	000676		64\$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
003562	001403			BEG 65\$;BR IF OK
003564	010637	000700		MOV SP,#BADSP6 ;STORE BAD SP VALUE
003570	104002			ERROR 2 ;*****TEST 10 - ERROR 2*****
				STACK POINTER NOT RESTORED BY INSTRUCTION
				EXPECTED VALUE OF SP IS STORED AT "SAVR6"
				ERRONEOUS VALUE IS AT "BADSP6"
003572	026704	175056		65\$: CMP FILL,R4 ;CHECK R4 UNCHANGED
003576	001401			BEG 66\$;BR IF OK
003600	104003			ERROR 3 ;*****TEST 10 - ERROR 3*****
				R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
003602	026705	175050		66\$: CMP TABLE,R5 ;CHECK R5 UNCHANGED
003606	001401			BEG 67\$;BR IF OK
003610	104004			ERROR 4 ;*****TEST 10 - ERROR 4*****
				R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

K03

 MAIN MAC11 27(1006) 22-DEC-76 10:49 PAGE 36
 CARRIA.P11 22-DEC-76 10:47 T10

TEST "MOV" 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
1529 003630          69$:      TST      R1      :TEST R1
1529 003630 005701          BEQ      69$      :BR, IF ZERO
1530 003632 001401          ERROR    6      :*****TEST 10 - ERROR 6*****
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$:          :VERIFY DESTINATION AREA
1542          :POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
1543 003652 012700 017034      MOV      #BUF2,R0
1544 003656 105720          TSTB     (R0)+
1545 003660 001401          BEQ      72$
1546 003662 104011          ERROR    11     :*****TEST 10 - ERROR 11*****
1547          :LOWER BOUNDARY OF DESTINATION CHANGED
1548          :  SHOULD STILL EQUAL ZERO
1549 003664          72$:          :STORE TRANSFER BYTE COUNT IN R1
1550 003664 016702 174756      MOV      SRCAD,R2
1551 003670 016701 174754      MOV      DSTLN,R1
1552 003674 122022          TICE12: CMPB     (R0)+,(R2)+
1553 003676 001401          BEQ      +4
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
1559 003704 001373          BNE      TICE12
1560 003706 105720          TSTB     (R0)+
1561 003710 001401          BEQ      END-10
1562 003712 104013          ERROR    13     :DECREMENT BYTE COUNT
1563          :BR, IF NOT FINISHED CHECKING
1564          :TEST CONTENTS OF DEST. UPPER BOUNDARY
1565          :BR, IF STILL ZERO
1566          :*****TEST 10 - ERROR 13*****
1567          :UPPER BOUNDARY OF DEST. CHANGED
1568          :  SHOULD STILL EQUAL ZERO
1569 003714          ENDT10:

```

```

1567
1568
1569
1570
1571
1572
1573
1574
1575
1576 003714 000004
1577 003716 004567 01127C
1578 003722 016434
1579 003724 000010
1580 003726 017034
1581 003730 000010
1582 003732 000377
1583 003734 012737 076032 003762
1584 003742 013767 003510 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, RD         ;STORE BAD OPCODE THAT DID NOT TRAP
        ERROR    1               ;*****TEST 11 - ERROR 1*****
        ;BAD MOVE OPCODE DID NOT TRAP
        ;RD 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 004052 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 17446E
1651 004204 023767 000674 17446C
1652 004212 001401
1653 004214 104001
1654
1655
1656

```

```

*****
*TEST 12      TEST INTERRUPTABILITY OF "MOVE" INSTRUCTION
*****
*THIS TEST INTERRUPTS THE EXECUTION OF THE "MOVE"
*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
*****
T12:  SCOPE
      TSTB      @SWR      ;TEST BIT 7 OF SWR
      BMI      TST13     ;SKIP TO NEXT TEST IF SET
      CMP      @TPS,TCR  ;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
                        374 ;SOURCE LENGTH
                        BUF1 ;SOURCE ADDRESS
                        376 ;DESTINATION LENGTH
                        BUF2+1 ;DESTINATION ADDRESS
                        377 ;FILL CHARACTER
      JSR      PC,CLDST  ;CLEAR DESTINATION
      MOV      @MC,PCI ;STORE PC OF TEST INSTRUCTION
      MOV      @INTR,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
      CLR      @TPSW     ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
      JSR      PC,TDONE
      MOV      @R,$NULL,@TBUF ;SEND CARRIAGE RETURN
      JSR      R5,XPSW   ;STORE EXPECTED PSW VALUE
                        11
      .WORD
      MTPS     @0
                        ;SET PSW TO ALLOW INTERRUPTS
      BIS      @100,@TCR ;ENABLE TTY INTERRUPTS
      JSR      PC,GENR   ;SET UP GENERAL REGISTERS
      +CLN!CLC
      +SEV!SEZ

REPMC:
      JSR      PC,GENR

MC:    MOV      CCODES ;EXECUTE "MOVE CHARACTER" INSTRUCTION
      MFPS     CCODES ;STORE THE PSW
      BIT      @100,@TCR ;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
 CVKATA.P11 22-DEC-76 10:47 T12

TEST INTERRUPTABILITY OF "MOVC" INSTRUCTION

SEQ 2041

1713	004346	166701	174272		SUB	SRCLN,R1	:CHARACTERS THAT SHOULD APPEAR IN DEST.
1714	004352	122067	174276	7\$:	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, @TVECT	: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 "MOVRC" 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"
MOVRC
          ;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 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
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
 CVKATA.P11 22-DEC-76 10:47 T13

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

SEQ 0042

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			68\$: TST	R1	:CHECK OTHER GENERAL REGISTERS
1793	004546	005701		BEQ	69\$	:TEST R1
1794	004550	001401		ERROR	6	:BR, IF ZERO
1795	004552	104006				:*****TEST 13 - ERROR 6*****
1796						:R1 SHOULD BE ZERO
1797	004554	005702		69\$: 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		70\$: 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			71\$:		
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			72\$:		
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 CONTAIN 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

*TEST 14 TEST "MOVRC" INSTRUCTION WITH SRCAD(LSB) .LT. DSTAD(LSB), DL .GT. SL

*PROPER TERMINATION FOR THIS INSTRUCTION TEST IS A
*TRANSFER OF ALL BYTES FROM SOURCE TO DESTINATION
*AND "FILL" CHARACTERS IN THE MSB OF THE DESTINATION.
*(R0-->R3 EQUAL ZERO, AND CONDITION CODES N,C=1 AND
*Z,V=0

TST14: SCOPE
JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
11 ;SOURCE LENGTH
BUF1 ;SOURCE ADDRESS
20 ;DESTINATION LENGTH
BUF2+1 ;DESTINATION ADDRESS
377 ;FILL CHARACTER
JSR PC,CLOST ;CLEAR DESTINATION
JSR 5,XPSW ;STORE EXPECTED PSW VALUE
.WORD 211
JSR PC,GENR ;SET UP GENERAL REGISTERS
+CLN:CLC ;CLEAR CONDITION CODES N&C
+SEV:SEZ ;SET CONDITION CODES V&Z
;EXECUTE "MOVE REVERSE CHARACTER"

MOVRC
;CHECK RESULTS
MFPS CCODES ;STORE THE PSW
BIC #17400,CCODES ;CLEAR ALL UNUSED BITS
CMP #EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
BEQ 64\$;BR, IF EQUAL
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\$;BR IF OK
MOV SP,#BADP6 ;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 "BADP6"

65\$:
TST R0 ;CHECK R0=ZERO
BEQ 1\$;BR, IF ZERO
ERROR 3
*****TEST 14 - ERROR 3*****
R0 SHOULD BE ZERO
CHECK OTHER GENERAL REGISTERS
1\$:
TST R1 ;TEST R1
BEQ 66\$;BR, IF ZERO
ERROR 4
*****TEST 14 - ERROR 4*****
R1 SHOULD BE ZERO

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

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

SEG 0045

1888	004762	005702		66\$:	TST R2		:TEST R2
1889	004764	001401			BEG 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			BEG 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			BEG 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			BEG 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			BEG 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			BEG +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			BEG +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			BEG 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:			

[illegible]

```

:*****
: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
                                ;SOURCE LENGTH
                                ;SOURCE ADDRESS
                                ;DESTINATION LENGTH
                                ;DESTINATION ADDRESS
                                ;FILL CHARACTER
        JSR      PC,CLDST     ;CLEAR DESTINATION AREA
        JSR      R5,GENSRC    ;GENERATE A SOURCE STRING
                                ;STORE EXPECTED PSW VALUE
        .WORD    20
        .WORD    BUF2+20
        JSR      R5,XPSW
        .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
        MCV      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$:
        CL      FILL,R4       ;CHECK R4 UNCHANGED
        BEQ      66$          ;BR IF OK
        ERROR    3
                                ;*****TEST 15 - 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 15 - ERROR 4*****
                                ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

```

```

1996 005240      67$: MOV      SRCLN,R5      ;CHECK R0=SOURCE LENGTH MINUS
1997 005242      166705 173400      SLB      DSTLN,R5      ;  DESTINATION LENGTH
1998 005244      166705 173400      CMP      R0,R5
1999 005250      02C005      BEQ      68$
2000 005252      001401      BEQ      68$
2001 005254      104005      ERROR      5      ;*****TEST 17 - ERROR 5*****
2002                                     ;R0 NOT EQUAL TO (SRCLN-DSTLN)
2003 005256      68$: TST      R1      ;CHECK OTHER GENERAL REGISTERS
2004 005256      005701      BEQ      69$      ;TEST R1
2005 005260      001401      BEQ      69$      ;BR, IF ZERO
2006 005262      104006      ERROR      6      ;*****TEST 15 - ERROR 6*****
2007                                     ;R1 SHOULD BE ZERO
2008 005264      005702      69$: TST      R2      ;TEST R2
2009 005266      001401      BEQ      70$      ;BR IF ZERO
2010 005270      104007      ERROR      7      ;*****TEST 15 - ERROR 7*****
2011                                     ;R2 SHOULD BE ZERO
2012 005272      005703      70$: TST      R3      ;TEST R3
2013 005274      001401      BEQ      71$      ;BR, IF ZERO
2014 005276      104010      ERROR      10     ;*****TEST 15 - ERROR 10*****
2015                                     ;R3 SHOULD BE ZERO
2016 005300      71$:
2017                                     ;VERIFY DESTINATION AREA
2018 005300      012700 017034      MOV      #BUF2,R0      ;POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
2019 005304      105720      TSTB      (R0)+      ;TEST CONTENTS OF BOUNDARY
2020 005306      001401      BEQ      72$      ;BR, IF STILL ZERO
2021 005310      104011      ERROR      11     ;*****TEST 15 - ERROR 11*****
2022                                     ;LOWER BOUNDARY OF DESTINATION CHANGED
2023                                     ; SHOULD STILL EQUAL ZERO
2024 005312      72$:
2025 005312      016702 173326      MOV      SRCLN,R2      ;CALCULATE ADDRESS OF MSB OF
2026 005316      166702 173326      SUB      DSTLN,R2      ;  THE SOURCE TO BE MOVED
2027 005322      066702 173320      ADD      SRCAD,R2
2028 005326      016701 173316      MOV      DSTLN,R1
2029 005332      122022      T15E12: CMPB      (R0)+,(R2)+      ;STORE TRANSFER BYTE COUNT IN R1
2030 005334      001401      BEQ      +4      ;CHECK CHARACTERS IN DESTINATION
2031 005336      104012      ERROR      12     ;BR IF OK
2032                                     ;*****TEST 15 - ERROR 12*****
2033                                     ;COMPARE ERROR IN DESTINATION
2034                                     ;R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
2035 005340      005301      DEC      R1      ;DECREMENT BYTE COUNT
2036 005342      001373      SNE      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:

```

22-DEC-76 10:49 PAGE 47
 22-DEC-76 10:47 T15

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

SEG 0048

```

*****
*TEST 16      TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB) & SRCLEN .GT. DESTLEN, 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.
*****
TS 16:  SCOPE
        JSR      R5,PREP          ;SET UP INSTRUCTION ARGUMENTS
        10
        BUF2+20          ;SOURCE LENGTH
        15              ;SOURCE ADDRESS
        BUF2+1          ;DESTINATION LENGTH
        377             ;DESTINATION ADDRESS
        JSR      PC,CLODST        ;FILL CHARACTER
        JSR      R5,GENSRC        ;CLEAR DESTINATION AREA
        .WORD    10              ;GENERATE SOURCE STRING
        .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      2*EXPPSW,CCODES  ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$              ;BR, IF EQUAL
        ERROR    1
        *****TEST 16 - ERROR 1*****
        PSW ERROR
        EXPECTED PSW IS STORED AT "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*BADSP6      ;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 "BADSP6"

        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*****

```

MACY 11 27 1006 22-DEC-76 10:49 PAGE 48
 00000000 22-DEC-76 10:47 16 TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB) & SRCLN .GT. DESTLN. FILL. SEG 0049

005502	005702		668:	TST	R2	:R1 SHOULD BE ZERO
005504	001401			BEG	678	:TEST R2
005506	104005			ERROR	5	:BR IF ZERO
						:*****TEST 16 - ERROR 5*****
005510	005703		678:	TST	R3	:R2 SHOULD BE ZERO
005512	001401			BEG	683	:TEST R3
005514	104006			ERROR	6	:BR, IF ZERO
						:*****TEST 16 - ERROR 6*****
005516			688:			:R3 SHOULD BE ZERO
005516	026704	173132		CMP	FILL,R4	:CHECK R4 UNCHANGED
005522	001401			BEG	695	:BR IF OK
005524	104007			ERROR	7	:*****TEST 16 - ERROR 7*****
						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
005526	026705	173124	698:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
005532	001401			BEG	703	:BR IF OK
005534	104010			ERROR	10	:*****TEST 16 - ERROR 10*****
						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
005536			708:			
005536	012730	017034		MOV	#BLF2,R0	:POINT R0 TO DESTINATION LOWER BYTE BOUNDARY
005542	105720			TSTB	(R0)+	:TEST CONTENTS OF BOUNDARY
005544	001401			BEG	713	:BR, IF STILL ZERO
005546	104011			ERROR	11	:*****TEST 16 - ERROR 11*****
						:LOWER BOUNDARY OF DESTINATION CHANGED
						: SHOULD STILL EQUAL ZERO
005550			718:			
005550	016702	173074		MOV	DSTLN,R2	:CALCULATE "FILL" COUNT
005554	166702	173064		SUB	SRCLN,R2	
005560	010201			MOV	R2,R1	:STORE TRANSFER BYTE COUNT IN R1
005562	122067	173066	716E12:	CMPB	(R0)+,FILL	:CHECK CHARACTERS IN DESTINATION
005566	001401			BEG	+4	:BR IF OK
005570	104012			ERROR	12	:*****TEST 16 - ERROR 12*****
						:COMPARE ERROR IN DESTINATION
						:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
005572	005301			DEC	R1	:DECREMENT BYTE COUNT
005574	001372			BNE	716E12	:BR, IF NOT FINISHED CHECKING
005576	016702	173044		MOV	SRCAD,R2	
005602	016701	173036		MOV	SRCLN,R1	:STORE TRANSFER BYTE COUNT IN R1
005606	122022		716E13:	CMPB	(R0)+,(R2)+	:CHECK CHARACTERS IN DESTINATION
005610	001401			BEG	+4	:BR IF OK
005612	104013			ERROR	13	:*****TEST 16 - ERROR 13*****
						:COMPARE ERROR IN DESTINATION
						:R0 CONTAINS THE PC+1 OF THE BAD DESTINATION CHARACTER
005614	005301			DEC	R1	:DECREMENT BYTE COUNT
005616	001373			BNE	716E13	:BR, IF NOT FINISHED CHECKING
005620	105720			TSTB	(R0)+	:TEST CONTENTS OF DEST. UPPER BOUNDARY
005622	001401			BEG	END16	:BR, IF STILL ZERO
005624	104014			ERROR	14	:*****TEST 16 - ERROR 14*****
						:UPPER BOUNDARY OF DEST. CHANGED
						: SHOULD STILL EQUAL ZERO
005626			END16:			

K04

 MACY11 2710061 22-DEC-76 10:49 PAGE 49
 22-DEC-76 10:47 716

TEST "MOVRC" WITH DSTAD(LSB) .LT. SRCAD(LSB) & SRCLN .GT. DSTLN. (FILL)

SEG 0050

 155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210

005626	000004		
005630	105777	172704	
005634	100571		
005636	026767	172706	173014
005644	001007		
005646	032767	000001	172732
005654	001403		
005656	005767	172712	
005662	001156		
005664			
005664	004567	007322	
005670	000374		
005672	016434		
005674	000376		
005676	017035		
005700	000377		
005702	004767	007332	
005706	012767	005774	172754
005714	012777	015356	172742
005722	005077	172740	
005726	004767	007403	
005732	013777	000554	172722
005740	004567	007342	
005744	000011		
005746	106427	000000	
005752	052777	000100	172700
005760	004767	007162	
005764	010637	000676	
005770	000251		
005772	000266		
005774	076031		
005776	106767	172670	
006002	032777	000100	172650
006010	001363		
006012	042767	177400	172652
006020	023767	000674	172644
006026	001401		
006030	104001		

```

*****
*TEST 17      TEST INTERRUPTABILITY OF "MOVRC" INSTRUCTION
*****
*THIS TEST INTERRUPTS THE EXECUTION OF THE "MOVRC"
*INSTRUCTION, RESUMES THE INSTRUCTION AFTER THE
*INTERRUPT, AND VERIFIES THE RESULTS.  THE PROPER
*RESULT IS ALL BYTES MOVED TO THE DESTINATION,
*RD-->R3 EQUAL ZERO, AND CONDITION CODES N,C=1
*AND Z,V=0.
*****
TST17:  SCOPE
        TSTB    @SWR      ;TEST BIT 7 OF SWR
        BMI     TST20     ;SKIP TO NEXT TEST IF SET
        CMP     $TPS,TCSP ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
        BNE     T17CONT   ;BR. IF NOT & PERFORM INTERRUPTABILITY TEST
        BIT     @BIT0,$ENV ;IF YES, CHECK IF ON APT
        BEQ     T17CONT   ;BR IF NOT UNDER APT, AND DO THIS TEST
        TST     $PASS     ;CHECK IF ON FIRST PASS
        BNE     TST20     ;IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T17CONT: JSR      R5,PREP   ;SET UP INSTRUCTION ARGUMENTS
        374      ;SOURCE LENGTH
        BUF1     ;SOURCE ADDRESS
        376     ;DESTINATION LENGTH
        BUF2+1   ;DESTINATION ADDRESS
        377     ;FILL CHARACTER
        JSR      PC,CLDST  ;CLEAR DESTINATION
        MOV      @MRC,PCI  ;STORE PC OF TEST INSTRUCTION
        MOV      @INT@,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
        CLR      @TPSW     ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
        JSR      PC,TDONE
        MOV      @@NULL,@TBUF ;SEND CARRIAGE RETURN
        JSR      R5,XPSW   ;STORE EXPECTED PSW VALUE
        .WORD    11
        MTPS     @0       ;SET PSW TO ALLOW INTERRUPTS
        BIS      @100,@TCSP ;ENABLE TTY INTERRUPTS
        JSR      PC,GENR   ;SET UP GENERAL REGISTERS
        MOV      SP,@$SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
        +CLN!CLC          ;CLEAR CONDITION CODES N&C
        +SEV!SEZ          ;CLEAR CONDITION CODES V&Z

MRC:    MOVRC           ;EXECUTE "MOVE REVERSE CHARACTER"

MFRS    CCODES         ;STORE THE PSW
        BIT      @100,@TCSP ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
        BNE     REPMRC   ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
        BIC      @177400,CCODES ;CLEAR ALL UNUSED BITS
        CMP      @EXPSPW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
        BEQ      64$     ;BR, IF EQUAL
        ERROR    1      ;*****TEST 17 - ERROR 1*****
        PSW ERROR
        EXPECTED PSW IS STORED AT "EXPSPW"

```


M04

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 51
 CVKAIH.F11 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			BEG	.+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			BEG	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,STVEC	: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 0C1401
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$:
        CMP      CSTLN,R2
        BEQ      3$
        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

SEG 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 5*****
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 0066C4 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 BJF2 ;SOURCE2 ADDRESS
2374 006416 000377 377 ;FILL CHARACTER
2375 006420 012700 017034 MOV #BUF2,R0 ;GENERATE SOURCE2 STRING STARTING
2376 006424 01272C 177777 MOV #177777,(R0)+ ;WITH TWO FILL BYTES
2377 006430 005C20 CLR (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 BEQ 64$ ;BR, IF EQUAL
2389 006472 104001 ERROR 1 ;*****TEST 21 - ERROR 1*****
2390
2391 ;PSW ERROR
2392 ;EXPECTED PSW IS STORED AT "EXPPSW"
2393 ;ACTUAL PSW IS STORED AT "CCODES"
2394 006474 023706 000676 64$: CMP #SAVR6,SP ;VERIFY STACK POINTER IS RESTORED
2395 006500 001403 BEQ 65$ ;BR IF OK
2396 006502 010637 000700 MOV SP,#BADR6 ;STORE BAD SP VALUE
2397 006506 104002 ERROR 2 ;*****TEST 21 - ERROR 2*****
2398 ;STACK POINTER NOT RESTORED BY INSTRUCTION
2399 ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
2400 ;ERRONEOUS VALUE IS AT "BADR6"
2401 006510 005700 55$: TST R0 ;CHECK R0=0
2402 006512 001401 BEQ 1$ ;
2403 006514 104003 ERROR 3 ;*****TEST 21 - ERROR 3*****
2404 ;SRC1 LENGTH ERROR
2405 ;CHECK R1 UNCHANGED
2406 006516 026701 172124 1$: CMP SRCAD,R1
2407 006522 001401 BEQ 2$
2408 006524 104004 ERROR 4 ;*****TEST 21 - ERROR 4*****
2409 ;SRC1 ADDRESS ERROR

```

D05

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

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

SEG 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	021703	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 000703
006702 104002

006704
006704 026704 171744
006710 001401
006712 104003

006714 026704 171736
006720 001401
006722 104004

```

*****
*TEST 22      TEST "CMPC" WITH SIL .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$          ;R3, 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
      MCV      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 27-10061 22-DEC-76 10:49 PAGE 57
 SYNTAX.P11 22-DEC-76 10:47 122

TEST "CMPC" WITH ALL .LT. 52L AND SRC1.GT.SRC2. NON-COMPAE IN SOURCE

SEG 0032

2484	006724			675:			
2485	006724	005067	171754	15:	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		25:	CMPB	(R4),(R5)	:COMPARE SOURCES
2489	006742	001004			BNE	35	: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	25	
2493	006754			35:			:CHECK ADDRESS & LENGTH DESCRIPTORS
2494	006754	020401			CMF	R4,R1	:CHECK R1, SOURCE1 ADDRESS
2495	006756	001401			BEG	45	: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		45:	CMP	R5,R3	:CHECK R3, SOURCE2 ADDRESS
2501	006764	001401			BEG	55	:BR IF OK
2502	006766	104006			ERROR	5	:*****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	55:	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			BEG	65	: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	65:	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			BEG	75	: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		75:	NCF		

000000
 000001
 000002
 000003
 000004
 000005
 000006
 000007
 000008
 000009
 000010
 000011
 000012
 000013
 000014
 000015
 000016
 000017
 000018
 000019
 000020
 000021
 000022
 000023
 000024
 000025
 000026
 000027
 000028
 000029
 000030
 000031
 000032
 000033
 000034
 000035
 000036
 000037
 000038
 000039
 000040
 000041
 000042
 000043
 000044
 000045
 000046
 000047
 000048
 000049
 000050
 000051
 000052
 000053
 000054
 000055
 000056
 000057
 000058
 000059
 000060
 000061
 000062
 000063
 000064
 000065
 000066
 000067
 000068
 000069
 000070
 000071
 000072
 000073
 000074
 000075
 000076
 000077
 000078

 007026 000004
 007030 004567 006156
 007034 000007
 007036 016434
 007040 000007
 007042 016434
 007044 000377
 007046 004567 006234
 007052 000204
 007054 004767 006066
 007060 000277
 007062 000244
 007064 076044
 007066 106767 171600
 007072 042767 177400 171572
 007100 023767 000674 171564
 007106 001401
 007110 104001
 007112
 007112 023706 000676
 007116 001403
 007120 010637 000700
 007124 104002
 007126
 007126 105700
 007130 001401
 007132 104003
 007134 105702
 007136 001401
 007140 104004
 007142 016700 171500
 007146 066700 171472
 007152 020001
 007154 001401
 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
*****
+5*23:  SCOPE
        JSR      RS,PREP          ;SET JP INSTRUCTION ARGUMENTS
        7
        BUI1          ;SOURCE1 LENGTH
        7
        BUI1          ;SOURCE1 ADDRESS
        377          ;SOURCE2 LENGTH
        377          ;SOURCE2 ADDRESS
        204          ;FILL CHARACTER
        JSR      RS,XPSW          ;STORE EXPECTED PSW VALUE
        .WORD
        JSR      PC,GENR          ;SET UP GENERAL REGISTERS
        SCC          ;SET ALL CONDITION CODES, EXCEPT
        CLZ          ;CLEAR "Z"
        CMPC          ;EXECUTE "CMPC" INSTRUCTION
        MFPS          ;CHECK RESULTS
        BIC          ;STORE THE PSW
        CMP          ;CLEAR ALL UNUSED BITS
        BEQ          ;CHECK PSW AGAINST EXPECTED VALUE
        ERROR        ;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$:   TSTB          R0
        BEQ          1$              ;CHECK R0
        ERROR        3              ;BR, IF ZERO
        *****TEST 23 - ERROR 3*****
        R0 SHOULD BE ZERO
1$:    TSTB          R2
        BEQ          2$              ;CHECK R2
        ERROR        4              ;BR, IF ZERO
        *****TEST 23 - ERROR 4*****
        R2 SHOULD BE ZERO
2$:    MOV          SRCAD,R0
        ADD          SRCEN,R0
        CMP          R0,R1
        BEQ          3$              ;CHECK R1
        ERROR        5              ;BR, IF EQUAL TO ADDRESS+1 OF LSB
        *****TEST 23 - ERROR 5*****

```

.MAIN. MACY11 2710051 22-DEC-76 10:49 PAGE 59
 CYNRA.P11 22-DEC-76 10:47 T23 TEST "CMPC" FOR S1L=S2L, SOURCE1=SOURCE2 STRINGS

007160	020003		38:	CMP	R0,R3	:R1 SHOULD EQUAL (SRCAD)+1(SRC_L).
007162	001401			BEG	4C8	:CHECK R3
007164	104006			ERROR	6	:BR, IF EQUAL TO ADDRESS+1 OF _S9
						:*****TEST 23 - ERROR 5*****
						:R3 SHOULD EQUAL (DSTAD)+1(DST_L).
007166			4C8:			
007168	026704	171452		CMP	FILL,R4	:CHECK R4 UNCHANGED
007170	001401			BEG	668	:BR IF OK
007172	104007			ERROR	7	:*****TEST 23 - ERROR 7*****
						:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
007176	026705	171454	668:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
007202	001401			BEG	578	:BR IF OK
007204	104010			ERROR	10	:*****TEST 23 - ERROR 10*****
						:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
007208			578:			

22-DEC-76 10:49 PAGE 60
 22-DEC-76 10:49 TEST "CMPC" FOR SIL=SEL, SOURCE1=SOURCE2 STRINGS

SEG 0061

```

*****
*TEST 24      TEST "CMPC" WITH SOURCE2=0 & SOURCE1 CONTAINS ALL FILLS
*****
*THIS TEST VERIFIES THAT "CMPC" COMPARES SOURCE1 WITH THE FILL
*CHARACTER WHEN SOURCE2 LENGTH IS ZERO.
*RESULT IS EQUAL STINGS WHEN SOURCE1 IS ALL FILLS.
*****
+ST24:  SCOPE

007206 000004
007210 004567 005776      JSR      R5,PREP      ;SET UP INSTRUCTION ARGUMENTS
007214 000004              4          ;SOURCE1 LENGTH
007216 017034              BUF2      ;SOURCE1 ADDRESS
007220 000000              C          ;SOURCE2 LENGTH
007222 177777              NXM       ;SOURCE2 ADDRESS
007224 000377              377       ;FILL CHARACTER
007226 012700 017034      MOV      #BUF2,R0      ;GENERATE SOURCE1 STRING WITH
007232 012710 177777      MOV      #177777,R0    ;ALL FILLS
007236 012010              MOV      (R0)+,(R0)
007240 004567 006042      JSR      R5,XPSW      ;STORE EXPECTED PSW VALUE
007244 000204              .WORD      204
007246 004767 005674      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
007252 000277              SCX          ;SET ALL CONDITION CODES, EXCEPT
007254 000244              CLZ          ;CLEAR Z

007256 076044              CMPC         ;EXECUTE "CMPC" INSTRUCTION

007260 106767 171406      MFPS      000DES      ;STORE THE PSW
007264 042767 177400 171400  BIC      #177400,000DES ;CLEAR ALL UNUSED BITS
007272 023767 000674 171372  CMP      #EXPPSW,000DES ;CHECK PSW AGAINST EXPECTED VALUE
007300 001401              BEQ      64$      ;BR, IF EQUAL
007302 104001              ERROR      1      ;*****TEST 24 - ERROR 1*****
                                           ;PSW ERROR
                                           ;EXPECTED PSW IS STORED AT "EXPPSW"
                                           ;ACTUAL PSW IS STORED AT "000DES"

007304
007304 023705 000676      CMP      #SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
007310 001403              BEQ      65$      ;BR IF OK
007312 010637 000700      MOV      SP,#BADRF6 ;STORE BAD SP VALUE
007316 104002              ERROR      2      ;*****TEST 24 - ERROR 2*****
                                           ;STACK POINTER NOT RESTORED BY INSTRUCTION
                                           ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                                           ;ERRONEOUS VALUE IS AT "BADRF6"

007320
007320 005700 65$:      .ST      R0          ;CHECK R0=0
007322 001401              BEQ      1$      ;
007324 104003              ERROR      3      ;*****TEST 24 - ERROR 3*****
                                           ;SOURCE1 LENGTH ERROR
                                           ;R0 SHOULD BE ZERO
                                           ;CALCULATE ADDRESS OF LSB+1

007326 016700 171314 1$:      MOV      SRCAD,R0
007332 066700 171306      ADD      SRCLN,R0
007336 020001              CMP      PC,R1      ;CHECK R1, SOURCE1 ADDRESS
007340 001401              BEQ      2$      ;BR IF OK
007342 104004              ERROR      4      ;*****TEST 24 - ERROR 4*****

```

007344	005702		25:	1ST	R2	:SOURCE1 ADDRESS ERROR
007346	001401			BEG	35	: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	171274	33:	MOV	DESTAD,R2	:CALCULATE ADDRESS OF LSB+1
007356	066702	171266		ADD	DESTAD,R2	
007362	020203			CMP	R2,R3	:CHECK R3,SOURCE2 ADDRESS
007364	001401			BEG	45	:BR IF OK
007366	104005			ERROR	6	:*****TEST 24 - ERROR 6*****
						:SOURCE2 ADDRESS ERROR
						:R2 CONTAINS THE EXPECTED ADDRESS POINTER
						:R3 CONTAINS THE BAD VALUE
007370			43:			:CHECK R4 UNCHANGED
007370	026704	171260		CMP	FILL,R4	:BR IF OK
007374	001401			BEG	65	:*****TEST 24 - ERROR 7*****
007376	104007			ERROR	7	:R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
007400	026705	171252	55:	CMP	TABLE,R5	:CHECK R5 UNCHANGED
007404	001401			BEG	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
*****
+ S'25:  SCOPE
      JSR      R5,PREP      ;SET UP INSTRUCTION ARGJMENTS
                          4 ;SOURCE1 LENGTH
                          BUF2 ;SOURCE1 ADDRESS
                          0 ;SOURCE2 LENGTH
                          NXM ;SOURCE2 ADDRESS
                          377 ;FILL CHARACTER
      MOV      #BUF2,R0      ;GENERATE SOURCE1 STRING STARTING
      MOV      #177777,(R0)+ ;WITH TWO FILL BYTES
      CLR      (R0)+         ;THEN TWO "ZERO" BYTES
      JSR      R5,XPSW       ;STORE EXPECTED PSW VALUE
      .WORD    201
      JSR      PC,GENR       ;SET UP GENERAL REGISTERS
      SCC      ;SET ALL CONDITION CODES, EXCEPT
      CLN      ;CLEAR N=1
                          ;EXECUTE "CMPC" INSTRUCTION
      CMPC
                          ;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 25 - ERROR 1*****
                          ;PSW ERROR
                          ;EXPECTED PSW IS STORED AT "EXPPSW"
                          ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      @#SAVR6,SP    ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;BR IF OK
      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$:
      CMP      #2,R0         ;CHECK R0=2
      BEQ      1$           ;
      ERROR    3            ;*****TEST 25 - ERROR 3*****
                          ;SRC1 LENGTH ERROR
      1$:
      CMP      #BUF2+2,R1    ;CHECK R1=BUF2+2
      BEQ      2$           ;
      ERROR    4            ;*****TEST 25 - ERROR 4*****
                          ;SRC1 ADDRESS ERROR
      2$:
      TST      R2            ;CHECK R2=0
      BEQ      3$           ;

```

MAIN. MACY11 2710061 22-DEC-76 10:49 PAGE 63
 CKRARA.F11 22-DEC-76 10:47 T25

TEST "CMPC" WITH SOURCE2 LENGTH EQUAL TO ZERO

SEG 0004

2735	007546	104005		ERROR	5	*****TEST 25 - ERROR 5*****
2736						:SRC2 LENGTH ERROR
2737	007550	026703	171075	3\$:	CMP	DSTAD,R3
2738	007554	001401			BEG	4\$
2739	007556	104006			ERROR	6
2740						*****TEST 25 - ERROR 6*****
2741						:SRC2 ADDRESS ERROR
2742	007560			4\$:		
2743	007560	026704	171070		CMP	FILL,R4
2744	007564	001401			BEG	66\$
2745	007566	104007			ERROR	7
2746						*****TEST 25 - ERROR 7*****
2747	007570	026705	171062	56\$:	CMP	TABLE,R5
2748	007574	001401			BEG	67\$
2749	007576	104010			ERROR	10
2750						*****TEST 25 - ERROR 10*****
2751	007600			67\$:		:R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"

M05

MAIN MACY11 27(1006) 22-DEC-76 10:49 PAGE 64
 CYKARA.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
      BUF2          ;SOURCE1 LENGTH
      3             ;SOURCE1 ADDRESS
      BUF1          ;SOURCE2 LENGTH
      377           ;SOURCE2 ADDRESS
      JSR      R5,GENSRC        ;FILL CHARACTER
      7             ;GENERATE SOURCE1 STRING
      .WORD    BUF2           ;STRING LENGTH
      .WORD    2*BUF2+2       ;STRING ADDRESS
      CLR      2*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!C,N              ;CLEAR CONDITION CODES C&N
      +SEZ!SEV             ;SET CONDITION CODES Z&V
                           ;EXECUTE "CMPC" INSTRUCTION
      CMPC
                           ;CHECK RESULT
      MFPS      CCODES        ;STORE THE PSW
      BIC      2*EXPPSW,CCODES ;CLEAR ALL UNUSED BITS
      CMP      2*EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$            ;BR, IF EQUAL
      *****TEST 26 - 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
      EPROR    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	Op Code	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
---------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

```

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
        10          ;SOURCE1 LENGTH
        BUF1        ;SOURCE1 ADDRESS
        12          ;SOURCE2 LENGTH
        BUF2        ;SOURCE2 ADDRESS
        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              ;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
        BEQ      2$           ;CHECK R1, SOURCE1 ADDRESS
        ERROR    4             ;*****TEST 27 - ERROR 4*****

```

MAIN. MACY11 27, 1006) 22-DEC-76 10:49 PAGE 67
SVKRAIA.F11 22-DEC-76 10:47 T27 TE

SEP 206E

2896					;SOURCE1 ADDRESS ERROR
2897	010200	005702	2\$:	TST R2	;CHECK R2=0
2898	010202	001401		BEG 3\$	
2999	010204	104005		ERROR 5	:*****TEST 27 - ERROR 5*****
2900					:SOURCE2 LENGTH ERROR
2901	010206	016702	3\$:	MOV DSTAD,R2	:CALCULATE ADDRESS OF LSB+1
2902	010212	066702		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		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	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\$:		

```

2917
2918
2919
2920
2921
2922
2923
2924
2925
2926 010244 000004
2927 010246 004567 00474C
2928 010252 016434
2929 010254 000010
2930 010256 017034
2931 010260 000010
2932 010262 000377
2933 010264 012737 076045 010312
2934 010272 013767 000010 170406
2935 010300 012737 010322 000010
2936 010306 004767 004634
2937
2938 010312 076045
2939
2940 010314 016700 177772
2941 010320 104001
2942
2943
2944
2945 010322 012626
2946 010324 005257 177762
2947 010330 022767 076050 177754
2948 010336 001363
2949 010340 016737 170342 000010
2950

```

```

*****
*TEST 3C      TEST THAT BAD "SEARCH" OPCODES TRAP
*****
*THIS TEST VERIFIES THAT OPCODES 076045-->076047 TRAP TO
*LOCATION 10.
*****
*****
TST30:  SCOPE
        JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
                                ;SOURCE LENGTH
                                ;SOURCE ADDRESS
                                ;DESTINATION LENGTH
                                ;DESTINATION ADDRESS
                                ;FILL CHARACTER
                                ;STORE THE FIRST BAD MOVE OPCODE
                                ;SAVE ILLEGAL INSTRUCTION TRAP VECTOR
                                ;POINT ILLEGAL INSTRUCTION VECTOR TO CONTINUE TEST
                                ;SET UP GENERAL REGISTERS
        MOV      #076045,2*BD30
        MOV      2*10,TEMP1
        MOV      #T30CONT,2*10
REF30:  JSR      PC,GENR
        BD30:    .WORD 076045      ;EXECUTE BAD SEARCH INSTRUCTION
                                ;STORE BAD OPCODE THAT DID NOT TRAP
                                ;*****TEST 30 - ERROR 1*****
                                ;BAD SEARCH OPCODE DID NOT TRAP
                                ;RD CONTAINS THE BAD OPCODE
                                ;RESTORE THE STACK POINTER AFTER THE TRAP
                                ;INCREMENT INSTRUCTION OPCODE
                                ;FINISHED WITH BAD SEARCH OPCODES?
                                ;BR IF NOT
                                ;RESTORE ILLEGAL INSTRUCTION TRAP VECTOR
        T30CONT:MOV  (SP)+,(SP)+
        INC      BD30
        CMP      #076050,BD30
        BNE     REF30
        MOV      TEMP1,2*10

```

```

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      QSWR      ;TEST BIT 7 OF SWR
        BMI      TST32     ;SKIP TO NEXT TEST IF SET
        CMP      $TPS,TCR  ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
        BNE      T31CONT   ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
        BIT      #BITC,$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      RS,PREP   ;SET UP INSTRUCTION ARGUMENTS
        400      ;SOURCE1 LENGTH
        BUF1     ;SOURCE1 ADDRESS
        400      ;SOURCE2 LENGTH
        BUF1     ;SOURCE2 ADDRESS
        377     ;FILL CHARACTER
        MOV      #CMC,PCI  ;STORE PC OF TEST INSTRUCTION
        MOV      #INTA,$TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
        CLR      $TPSW     ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
        JSR      PC,TDONE
        MOV      Q,$NULL,$TBUF ;SEND CARRIAGE RETURN
        JSR      RS,XPSW   ;STORE EXPECTED PSW VALUE
        .WORD    04
        MTPS     #0        ;SET PSW TO ALLOW INTERRUPTS
        BIS      #100,$TCR ;ENABLE TTY INTERRUPTS
        JSR      PC,GENR   ;SET UP GENERAL REGISTERS
        MOV      SP,$SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
        SCC      ;SET ALL CONDITION CODES, EXCEPT
        CLZ      ;CLEAR Z=1

CMC:    CMPC      ;EXECUTE "CMPC" INSTRUCTION

        MFPS     CCODES   ;CHECK RESULTS
        BIT      #100,$TCR ;STORE THE PSW
        BNE      REPCMC   ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
        BIC      #177400,CCODES ;BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
        CMP      Q,$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"

```

```

E-4:

```

3007	010546	023706	000676		CMP	2#SAVR6,SP	:VERIFY STACK POINTER IS RESTORED
3009	010552	001403			BEG	65\$	:BR IF OK
3009	010554	010637	000700		MOV	SP,2#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	*PSW,STVECT	:RESTORE TRAP CATCHER

```

*****
*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.
*****
+*32:  SCOPE
      JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
      10      ;SOURCE LENGTH
      BCF:     ;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
      ;EXECUTE "LOCC"
      LOCC
      ;CHECK RESULTS
      MFPS     CCODES      ;STORE THE PSW
      BVC      177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      2*EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$
      ERROR    :
      ;*****TEST 32 - 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 32 - ERROR 2*****
      ;STACK POINTER NOT RESTORED BY INSTRUCTION
      ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
      ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      CMP      FILL,R4      ;CHECK R4 UNCHANGED
      BEQ      66$
      ERROR    3
      ;*****TEST 32 - ERROR 3*****
      ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
      66$:
      CMP      TABLE,R5   ;CHECK R5 UNCHANGED
      BEQ      67$
      ERROR    4
      ;*****TEST 32 - ERROR *****
      ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
      MOV      SRCAD,R3     ;LOCATE "CHARACTER"
      MOV      SPCLN,R2
      68$:
      CMPB     R3,FILL
      BEQ      69$

```

MAIL ROOM 27-1006 22-DEC-76 10:49 PAGE 72
MAIL ROOM 22-DEC-76 10:47 732 TE

55: 2273

035:	035B	033
	COMP	02
	BEG	69\$
	END	02
	REQ	70\$
	ERROR	5
035:	COMP	03
	BEG	71\$
	ERROR	6
035:		

```
:POINT TO NEXT BYTE
:DECREMENT BYTE COUNT
:BR IF NOT FINISHED
:CHECK RD=REMAINING SOURCE LENGTH

:*****TEST 32 - ERROR 5*****
:LOCATE LENGTH ERROR
:CHECK RI=ADDRESS OF CHARACTER

:*****TEST 32 - ERROR 5*****
:LOCATE ADDRESS ERROR
```

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 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. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 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. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840.

```

:*****
: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      RS,PREP      :SET UP INSTRUCTION ARGUMENTS
        100010      :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    210
        JSR      PC,GENR      :SET UP GENERAL REGISTERS
        SOC      :SET ALL CONDITION CODES
        :EXECUTE "LOCC"
        LOCC
        :CHECK RESULTS
        MFPS      CCODES      :STORE THE PSW
        BYC      #177400,CCODES :CLEAR ALL UNUSED BITS
        CMP      2*EXPPSW,CCODES :CHECK PSW AGAINST EXPECTED VALUE
        BEQ      545          :BR, IF EQUAL
        ERROR    :
        :*****TEST 33 - ERROR 1*****
        :PSW ERROR
        :EXPECTED PSW IS STORED AT "EXPPSW"
        :ACTUAL PSW IS STORED AT "CCODES"
643:    CMP      2*SAVR6,SP      :VERIFY STACK POINTER IS RESTORED
        BEQ      555          :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"
655:    CMP      575--R4      :CHECK R4 UNCHANGED
        BEQ      665          :BR IF OK
        ERROR    3
        :*****TEST 33 - ERROR 3*****
        :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
665:    CMP      TABLE,R5      :CHECK R5 UNCHANGED
        BEQ      675          :BR IF OK
        ERROR    4
        :*****TEST 33 - ERROR 4*****
        :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
675:    MOV      SRCAD,R3      :LOCATE "CHARACTER"
        MOV      SRCLN,R2
685:    CMPB     (R3),FILL
        BEQ      695

```

MAIN MAGY11 27:1006 22-DEC-76 10:49 PAGE 74
 C:\KATA.F11 22-DEC-76 10:47 T33

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

SEG 007E

| | | | | | | |
|------|--------|--------|-------|-------|-------|-----------------------------------|
| 3169 | 011166 | 105723 | | TSTB | (R3)+ | :POINT TO NEXT BYTE |
| 3170 | 011170 | 005302 | | DEC | R2 | :DECREMENT BYTE COUNT |
| 3171 | 011172 | 001372 | | BNE | 68\$ | :BR IF NOT FINISHED |
| 3172 | 011174 | 020200 | 59\$: | CMR | R2,R0 | :CHECK R0=REMAINING SOURCE LENGTH |
| 3173 | 011176 | 001401 | | BEC | 70\$ | |
| 3174 | 011200 | 104005 | | ERROR | 5 | :*****TEST 33 - ERROR 5***** |
| 3175 | | | | | | :LOCATE LENGTH ERROR |
| 3176 | 011202 | 020301 | 70\$: | CMR | R3,R1 | :CHECK R1=ADDRESS OF CHARACTER |
| 3177 | 011204 | 001401 | | BEC | 71\$ | |
| 3178 | 011206 | 104005 | | ERROR | 6 | :*****TEST 33 - ERROR 6***** |
| 3179 | | | | | | :LOCATE ADDRESS ERROR |
| 3180 | 011210 | | 71\$: | | | |

32193
32194
32195
32196
32197
32198
32199
32200
32201
32202
32203
32204
32205
32206
32207
32208
32209
32210
32211
32212
32213
32214
32215
32216
32217
32218
32219
32220
32221
32222
32223
32224
32225
32226
32227
32228
32229
32230
32231
32232
32233
32234
32235
32236
32237
32238
32239
32240
32241
32242
32243
32244
32245
32246
32247
32248
32249
32250
32251
32252
32253
32254
32255
32256
32257
32258
32259
32260
32261
32262
32263
32264
32265
32266
32267
32268
32269
32270
32271
32272
32273
32274
32275
32276
32277
32278
32279
32280
32281
32282
32283
32284
32285
32286
32287
32288
32289
32290
32291
32292
32293
32294
32295
32296
32297
32298
32299
32300
32301
32302
32303
32304
32305
32306
32307
32308
32309
32310
32311
32312
32313
32314
32315
32316
32317
32318
32319
32320
32321
32322
32323
32324
32325
32326
32327
32328
32329
32330
32331
32332
32333
32334
32335
32336
32337
32338
32339
32340
32341
32342
32343
32344
32345
32346
32347
32348
32349
32350
32351
32352
32353
32354
32355
32356
32357
32358
32359
32360
32361
32362
32363
32364
32365
32366
32367
32368
32369
32370
32371
32372
32373
32374
32375
32376
32377
32378
32379
32380
32381
32382
32383
32384
32385
32386
32387
32388
32389
32390
32391
32392
32393
32394
32395
32396
32397
32398
32399
32400
32401
32402
32403
32404
32405
32406
32407
32408
32409
32410
32411
32412
32413
32414
32415
32416
32417
32418
32419
32420
32421
32422
32423
32424
32425
32426
32427
32428
32429
32430
32431
32432
32433
32434
32435
32436
32437
32438
32439
32440
32441
32442
32443
32444
32445
32446
32447
32448
32449
32450
32451
32452
32453
32454
32455
32456
32457
32458
32459
32460
32461
32462
32463
32464
32465
32466
32467
32468
32469
32470
32471
32472
32473
32474
32475
32476
32477
32478
32479
32480
32481
32482
32483
32484
32485
32486
32487
32488
32489
32490
32491
32492
32493
32494
32495
32496
32497
32498
32499
32500
32501
32502
32503
32504
32505
32506
32507
32508
32509
32510
32511
32512
32513
32514
32515
32516
32517
32518
32519
32520
32521
32522
32523
32524
32525
32526
32527
32528
32529
32530
32531
32532
32533
32534
32535
32536
32537
32538
32539
32540
32541
32542
32543
32544
32545
32546
32547
32548
32549
32550
32551
32552
32553
32554
32555
32556
32557
32558
32559
32560
32561
32562
32563
32564
32565
32566
32567
32568
32569
32570
32571
32572
32573
32574
32575
32576
32577
32578
32579
32580
32581
32582
32583
32584
32585
32586
32587
32588
32589
32590
32591
32592
32593
32594
32595
32596
32597
32598
32599
32600
32601
32602
32603
32604
32605
32606
32607
32608
32609
32610
32611
32612
32613
32614
32615
32616
32617
32618
32619
32620
32621
32622
32623
32624
32625
32626
32627
32628
32629
32630
32631
32632
32633
32634
32635
32636
32637
32638
32639
32640
32641
32642
32643
32644
32645
32646
32647
32648
32649
32650
32651
32652
32653
32654
32655
32656
32657
32658
32659
32660
32661
32662
32663
32664
32665
32666
32667
32668
32669
32670
32671
32672
32673
32674
32675
32676
32677
32678
32679
32680
32681
32682
32683
32684
32685
32686
32687
32688
32689
32690
32691
32692
32693
32694
32695
32696
32697
32698
32699
32700
32701
32702
32703
32704
32705
32706
32707
32708
32709
32710
32711
32712
32713
32714
32715
32716
32717
32718
32719
32720
32721
32722
32723
32724
32725
32726
32727
32728
32729
32730
32731
32732
32733
32734
32735
32736
32737
32738
32739
32740
32741
32742
32743
32744
32745
32746
32747
32748
32749
32750
32751
32752
32753
32754
32755
32756
32757
32758
32759
32760
32761
32762
32763
32764
32765
32766
32767
32768
32769
32770
32771
32772
32773
32774
32775
32776
32777

```

:*****
:TEST 34      TEST "LOCC" WITH SEARCH CHARACTER NOT IN SOURCE
:*****
:PROPER TERMINATION OF THIS TEST IS RD EQUALS ZERO, R1
:EQUALS ADDRESS OF LSB OF SOURCE PLUS ONE, AND ALL
:CONDITION CODES CLEAR EXCEPT Z=1.
:*****
:*****
†ST34:  SCOPE
        JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
        377          ;SOURCE LENGTH
        BJF1        ;SOURCE ADDRESS
        NXM          ;STORE NON-ZERO VALUES TO TEST R2 & R3
        NXM          ;UNAFFECTED BY DIS STRING INSTRUCTIONS
        C            ;SEARCH IMASK FOR SCANC SPANC: CHARACTER
        JSR      RS,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
                                ;EXECUTE "LOCC"
        LOCC
                                ;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$        ;BR, IF EQUAL
        ERROR      1          ;*****TEST 34 - 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*BADP6 ;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 "BADP6"
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"
66$:
        CMP        TABLE,R5 ;CHECK R5 UNCHANGED
        BEQ        67$        ;BR IF OK
        ERROR      4          ;*****TEST 34 - ERROR 4*****
                                ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
67$:
        MOV        SRCAD,R3    ;LOCATE "CHARACTER"
        MOV        SRCLN,R2
68$:
        CMPB       (R3),FILL
        BEQ        69$

```

| | | | | | |
|------|--------|--------|-------|-------|-----------------------------------|
| 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 | 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 | 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 | | | | |

```

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
*R0&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        ;UNAFFECTED 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     C00DES       ;CHECK RESULTS
      BIC      #177400,C00DES ;STORE THE PSW
      CMP      @#EXPPSW,C00DES ;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 "C00DES"
      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$:
      TSTB     R0            ;CHECK R0
      BEQ      1$           ;RO CHANGED
      ERROR    3            ;*****TEST 35 - ERROR 3*****
      1$:
      CMP      SRCAD,R1      ;CHECK R1
      BEQ      40$          ;R1 CHANGED
      ERROR    4            ;*****TEST 35 - ERROR 4*****
      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 $TPS,TC SR ;IS SLU USED FOR INTERRUPTS THE CONSOLE?
      BNE T36CONT ;BR, IF NOT & PERFORM INTERRUPTABILITY TEST
      BIT @BIT0,$ENV ;IF YES, CHECK IF ON APT
      BEQ T36CONT ;BT 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,PCI ;STORE PC OF TEST INSTRUCTION
          MOV @INTA,@TVECT ;POINT TTY VECTOR TO INTERRUPT ROUTINE
          CLR @TPSW ;ALLOW INTERRUPTS AFTER TTY INTERRUPT
          JSR PC,TDCNE
          MOV @#$NULL,@TBUF ;SEND CARRIAGE RETURN
          JSR R5,XPSW ;STORE EXPECTED PSW VALUE
          .WORD 00
          MTPS #0 ;SET PSW TO ALLOW INTERRUPTS
          BIS #100,@TC SR ;ENABLE TTY INTERRUPTS
          JSR PC,GENR ;SET UP GENERAL REGISTERS
          MOV SP,@$SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
          SCC ;SET ALL CONDITION CODES

LOC: LOCC ;EXECUTE "LOCC"

MFFS CCODES ;STORE THE PSW
BIT #100,@TC SR ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
BNE REPLC ;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 R1=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 | #2C0 | ;RESTORE PSW TO PRIORITY 7 |
| 3388 | 012004 | 016777 | 166656 | | MOV | TPSW,STVECT | ;RESTORE TRAP CATCHER |
| 3389 | | | 166652 | | | | |

```

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
*****
*ST37: 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
      JSR      R5,XPSW
      .WORD    200
      JSR      PC,GENR      ;SET UP GENERAL RESISTERS
      SCC                      ;SET ALL CONDITION CODES
                          ;EXECUTE "SKPC"
      SKPC
                          ;CHECK RESULTS
                          ;STORE THE PSW
      MFPS     CCODES
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      3*EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
      ERROR    1
                          ;*****TEST 37 - 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 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$:
      MOV      SRCAD,R3      ;SKIP "CHARACTER"
      MOV      SRCCLN,R2
      68$:
      CMPB     (R3),FILL

```

MAIN: MACY11 27(1006) 22-DEC-76 10:49 PAGE 81
 CVKATA.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 C03014
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 012252 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 RD EQUALS ZERO
*R1 EQUALS THE ADDRESS+1 OF THE LSB IN THE SOURCE STRING,
*AND ALL CONDITION CODES ARE CLEAR EXCEPT Z=1.
*****
TS40:  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)+
      1$      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     ;CHECK RESULTS
      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 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
      MCV      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"

```

MACY11 27(1006) 22-DEC-76 10:49 PAGE 83
 SYRRIA.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 | SRCLN,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 4C - ERROR 6***** |
| 3528 | 012366 | 104005 | | | | :SKIP ADDRESS ERROR |
| 3529 | | | 71\$: | | | |
| 3530 | 012370 | | | | | |

```

3532
3533
3534
3535
3536
3537
3538
3539
3540
3541 012370 000004
3542 012372 004567 002614
3543 012376 000000
3544 012400 177777
3545 012402 177777
3546 012404 177777
3547 012406 000000
3548 012410 004567 002672
3549 012414 000204
3550 012416 004767 003524
3551 012422 000244
3552 012424 000267
3553
3554 012426 076041
3555
3556 012430 106767 166236
3557 012434 042767 177400 166230
3558 012442 023767 000674 166222
3559 012450 001401
3560 012452 104001
3561
3562
3563
3564 012454
3565 012454 023706 000676
3566 012460 001403
3567 012462 010537 000700
3568 012466 104002
3569
3570
3571 012470
3572 012470 105700
3573 012472 001401
3574 012474 104003
3575
3576 012476 026701 166144
3577 012502 001401
3578 012504 104004
3579
3580 012506

```

```

*****
*TEST 41 TEST "SKPC" WITH ZERO SOURCE LENGTH
*****
*PROPER TERMINATION OF TEST IS NO CHANGE IN CONTENTS OF
*RO3R1, AND ALL CONDITION CODES CLEAR EXCEPT Z=1.
*****
*ST41: SCOPE
      JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS
      0 ;SOURCE LENGTH
      NXM ;SOURCE ADDRESS
      NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3
      0 ;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 "Z" CONDITION CODE
      +SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES
                   ;EXECUTE "SKPC"
      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 41 - 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 41 - ERROR 2*****
                   ;STACK POINTER NOT RESTORED BY INSTRUCTION
                   ;EXPECTED VALUE OF SP IS STORED AT "SAVR6"
                   ;ERRONEOUS VALUE IS AT "BADR6"
      65$:
      TSTB R0 ;CHECK R0
      BEQ 1$
      ERROR 3 ;*****TEST 41 - ERROR 3*****
                   ;R0 CHANGED
      1$:
      CMP SRCAD,R1 ;CHECK R1
      BEQ 40$
      ERROR 4 ;*****TEST 41 - ERROR 4*****
                   ;R1 CHANGED
      40$:

```

```

012506 000004
012510 105777 166024
012514 100550
012516 026767 166026 166124
012524 001007
012526 032767 000001 166052
012534 001403
012536 005767 166032
012542 001135
012544
012544 004567 002442
012550 000400
012552 017034
012554 177777
012556 177777
012560 000377
012562 016700 166060
012566 016701 166052
012572 005301
012574 116720 166054
012600 005301
012602 001374
012604 105020
012606 012767 012672 166054
012614 012777 015356 166042
012622 005077 166040
012626 004767 002500
012632 013777 000554 166022
012640 004567 002442
012644 000000
012646 106427 000000
012652 052777 000100 166000
012660 004767 002262
012664 010637 000676
012670 002277
012672 076041
012674 106767 165772
012700 032777 000100 165752
012706 001364
012710 042767 177400 165754
012716 023767 000674 165746

*****
*TEST 42 TEST INTERRUPTABILITY OF "SKPC" INSTRUCTION
*****
*THIS TEST INTERRUPTS THE EXECUTION OF "SKPC"
*INSTRUCTION, RESUMES EXECUTION AFTER THE INTERRUPT,
*AND CHECKS THE RESULTS. RESULTS INDICATE THE NON-SKIP
*CHARACTER FOUND
*****
TST42: SCOPE
TSTB 2SWR :TEST BIT 7 OF SWR
BMI TST43 :SKIP TO NEXT TEST IF SET
CMP STPS, TCSR :IS SLU USED FOR INTERRUPTS THE CONSOLE?
BNE T42CONT :BR, IF NOT & PERFORM INTERRUPTABILITY TEST
BIT 2BIT0, SENV :IF YES, CHECK IF ON APT
BEQ T42CONT :BR IF NOT UNDER APT, AND DO THIS TEST
ST SPASS :CHECK IF ON FIRST PASS
BNE TST43 :IF NOT ON FIRST PASS, BR & SKIP THIS TEST

T42CONT: JSR RS, PREP :SET UP INSTRUCTION ARGUMENTS
400 :SOURCE LENGTH
BCF2 :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, RC :GENERATE SOURCE STRING OF ALL
MOV SRCLN, R1 :SKIP CHARACTERS EXCEPT THE LAST BYTE
DEC R1 :ADJUST BYTE COUNT=SRCLN-1
CONTG: MOV8 FILL, (RC)+
DEC R1
BNE CONTG
CLR8 (RC)+ :PUT NON-SKIP CHARACTER IN LAST BYTE
MOV 2SKP, PCI :STORE PC OF TEST INSTRUCTION
MOV 2INTR, 2*VECT :POINT TTY VECTOR TO INTERRUPT ROUTINE
CLR 2TPSW :ALLOW INTERRUPTS AFTER TTY INTERRUPT
JSR PC, TDONE
MOV 22$NULL, 2TBUF :SEND CARRIAGE RETURN
JSR RS, XPSW :STORE EXPECTED PSW VALUE
.WORD DC
MTPS 20 :SET PSW TO ALLOW INTERRUPTS
BIS 2100, 2TCSR :ENABLE TTY INTERRUPTS
REPSKP: JSR PC, GENR :SET UP GENERAL REGISTERS
MOV SP, 2$SAVR6 :COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
S0C :SET ALL CONDITION CODES

SKP: SKPC :EXECUTE "SKPC"

MFPS 0000ES :CHECK RESULTS
BIT 2100, 2TCSR :STORE THE PSW
BNE REPSKP :IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
BIT 217400, 0000ES :BR & DO TEST AGAIN, IF INSTRUCTION WAS NOT INTERRUPTED
CMP 22EYPPSW, 0000ES :CHECK PSW AGAINST EXPECTED VALUE

```

| | | | | | | |
|------|--------|--------|---------------|-------|-------------|---|
| 3643 | 012734 | 001401 | | BEG | 64\$ | :BR, IF EQUAL |
| 3644 | 012736 | 104001 | | ERROR | 1 | :*****TEST 42 - ERROR 1***** |
| 3645 | | | | | | :PSW ERROR |
| 3646 | | | | | | :EXPECTED PSW IS STORED AT "EXPPSW" |
| 3647 | | | | | | :ACTUAL PSW IS STORED AT "CCODES" |
| 3648 | 012730 | | 64\$: | | | |
| 3649 | 012730 | 023706 | 000676 | COMP | 2#SAVR6, SP | :VERIFY STACK POINTER IS RESTORED |
| 3650 | 012734 | 001403 | | BEG | 65\$ | :BR IF OK |
| 3651 | 012736 | 000700 | | MOV | SP, 2#BADR6 | :STORE BAD SP VALUE |
| 3652 | 012736 | 104002 | | ERROR | 2 | :*****TEST 42 - ERROR 2***** |
| 3653 | | | | | | :STACK POINTER NOT RESTORED BY INSTRUCTION |
| 3654 | | | | | | :EXPECTED VALUE OF SP IS STORED AT "SAVR6" |
| 3655 | | | | | | :ERRONEOUS VALUE IS AT "BADR6" |
| 3656 | 012744 | | 65\$: | | | |
| 3657 | 012744 | 025704 | 165704 | COMP | 57--R4 | :CHECK R4 UNCHANGED |
| 3658 | 012750 | 001401 | | BEG | 66\$ | :BR IF OK |
| 3659 | 012752 | 104003 | | ERROR | 3 | :*****TEST 42 - ERROR 3***** |
| 3660 | | | | | | :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL" |
| 3661 | 012754 | 025705 | 165676 | COMP | TABLE, R5 | :CHECK R5 UNCHANGED |
| 3662 | 012760 | 001401 | | BEG | 67\$ | :BR IF OK |
| 3663 | 012762 | 104004 | | ERROR | 4 | :*****TEST 42 - ERROR 4***** |
| 3664 | | | | | | :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE" |
| 3665 | 012764 | | 67\$: | | | |
| 3666 | 012764 | 016703 | 165656 | MOV | SRCAD, R3 | :SKIP "CHARACTER" |
| 3667 | 012770 | 016702 | 165650 | MOV | SRCIN, R2 | |
| 3668 | 012774 | 121367 | 165654 | COMP | (R3), FILL | |
| 3669 | 013000 | 001003 | | BNE | 69\$ | |
| 3670 | 013002 | 105723 | | TSTB | (R3)+ | :POINT TO NEXT BYTE |
| 3671 | 013004 | 005302 | | DEC | R2 | :DECREMENT BYTE COUNT |
| 3672 | 013006 | 001372 | | BNE | 68\$ | :BR, IF NOT FINISHED |
| 3673 | 013010 | 020200 | | COMP | R2, R0 | :CHECK R0=REMAINING SOURCE LENGTH |
| 3674 | 013012 | 001401 | | BEG | 70\$ | |
| 3675 | 013014 | 104005 | | ERROR | 5 | :*****TEST 42 - ERROR 5***** |
| 3676 | | | | | | :SKIP LENGTH ERROR |
| 3677 | 013016 | 020301 | | COMP | R3, R1 | :CHECK R1=ADDRESS OF CHARACTER |
| 3678 | 013020 | 001401 | | BEG | 71\$ | |
| 3679 | 013022 | 104006 | | ERROR | 6 | :*****TEST 42 - ERROR 6***** |
| 3680 | | | | | | :SKIP ADDRESS ERROR |
| 3681 | 013024 | | 71\$: | | | |
| 3682 | | | | | | |
| 3683 | 013024 | 106427 | 000200 | MTPS | #200 | :RES ARE PSW TO PRIORITY 7 |
| 3684 | 013030 | 016777 | 165632 165626 | MOV | TPSW, 3TVEC | :RESTORE TRAP CATCHER |

```

3687 013036 000004
3688 013040 004567 002146
3689 013044 000003
3690 013046 016434
3691 013050 177777
3692 013052 177777
3693 013054 000002
3694 013056 012767 016434 165572
3695 013064 004567 002216
3696 013070 000200
3697 013072 004767 002050
3698 013076 016705 165554
3699 013102 000277
3700
3701
3702
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 AND'ED WITH MASK=1 R0 EQUALS REMAINDER
*OF SOURCE LENGTH, AND ALL CONDITION CODES CLEAR.
*****
*543:  SCOPE
      JSR      RS,PREP      ;SET UP INSTRUCTION ARGUMENTS
                          3 ;SOURCE LENGTH
                          BLF1 ;SOURCE ADDRESS
                          NYM  ;STORE NON-ZERO VALUES TO TEST R2 & R3
                          NYM  ;UNAFECTED BY DIS STRING INSTRUCTIONS
                          2 ;SEARCH IMASK FOR SCANC/SPANC: CHARACTER
      MOV      #BUF1, TABLE
      JSR      RS,XPSW      ;STORE EXPECTED PSW VALUE
      .WORD    200
      JSR      PC,GENR      ;SET UP GENERAL REGISTERS
      MOV      TABLE,RS
      SCC
      ;SET ALL CONDITION CODES
      ;EXECUTE "SCANC"
      SCANC
      ;CHECK RESULTS
      MFPS      CCODES      ;STORE THE PSW
      BIC      #177400,CCODES ;CLEAR ALL UNUSED BITS
      CMP      3*EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALUE
      BEQ      64$          ;BR, IF EQUAL
      ;*****TEST 43 - ERROR 1*****
      ;PSW ERROR
      ;EXPECTED PSW IS STORED AT "EXPPSW"
      ;ACTUAL PSW IS STORED AT "CCODES"
      64$:
      CMP      3*SAVRE,SP    ;VERIFY STACK POINTER IS RESTORED
      BEQ      65$          ;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 "SAVRE"
      ;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
      66$:
      CMP      TABLE,R5
      BEQ      67$          ;BR IF OK
      ERROR    4
      ;*****TEST 43 - ERROR 4*****
      ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
      67$:
      MOV      SRCAD,R3      ;POINT R3 TO SOURCE STRING
      MOV      SRCLEN,R2     ;STORE SOURCE LENGTH IN R2
      68$:
      MOVB     (R3),R4       ;USE SOURCE BYTE AS INDEX

```

22-DEC-76 10:49 PAGE 08
22-DEC-76 10:47 43 TE

550 2089

| Address | Hex | Assembly | Comment |
|---------|--------|----------|---------------|
| 013200 | 042704 | BIC | #177400,R4 |
| 013204 | 136467 | BITB | BUF1(R4),FILL |
| 013208 | 001003 | BNE | 69\$ |
| 013212 | 105723 | *STB | (R3)+ |
| 013216 | 005302 | DEC | R2 |
| 013220 | 001355 | BNE | 68\$ |
| 013224 | 020200 | CMP | R2,R0 |
| 013228 | 001401 | BEG | 70\$ |
| 013232 | 104005 | ERROR | 5 |
| 013236 | 000000 | | |
| 013240 | 000000 | | |
| 013244 | 000000 | | |
| 013248 | 000000 | | |
| 013252 | 000000 | | |
| 013256 | 000000 | | |
| 013260 | 000000 | | |
| 013264 | 000000 | | |
| 013268 | 000000 | | |
| 013272 | 000000 | | |
| 013276 | 000000 | | |
| 013280 | 000000 | | |
| 013284 | 000000 | | |
| 013288 | 000000 | | |
| 013292 | 000000 | | |
| 013296 | 000000 | | |
| 013300 | 000000 | | |
| 013304 | 000000 | | |
| 013308 | 000000 | | |
| 013312 | 000000 | | |
| 013316 | 000000 | | |
| 013320 | 000000 | | |
| 013324 | 000000 | | |
| 013328 | 000000 | | |
| 013332 | 000000 | | |
| 013336 | 000000 | | |
| 013340 | 000000 | | |
| 013344 | 000000 | | |
| 013348 | 000000 | | |
| 013352 | 000000 | | |
| 013356 | 000000 | | |
| 013360 | 000000 | | |
| 013364 | 000000 | | |
| 013368 | 000000 | | |
| 013372 | 000000 | | |
| 013376 | 000000 | | |
| 013380 | 000000 | | |
| 013384 | 000000 | | |
| 013388 | 000000 | | |
| 013392 | 000000 | | |
| 013396 | 000000 | | |
| 013400 | 000000 | | |
| 013404 | 000000 | | |
| 013408 | 000000 | | |
| 013412 | 000000 | | |
| 013416 | 000000 | | |
| 013420 | 000000 | | |
| 013424 | 000000 | | |
| 013428 | 000000 | | |
| 013432 | 000000 | | |
| 013436 | 000000 | | |
| 013440 | 000000 | | |
| 013444 | 000000 | | |
| 013448 | 000000 | | |
| 013452 | 000000 | | |
| 013456 | 000000 | | |
| 013460 | 000000 | | |
| 013464 | 000000 | | |
| 013468 | 000000 | | |
| 013472 | 000000 | | |
| 013476 | 000000 | | |
| 013480 | 000000 | | |
| 013484 | 000000 | | |
| 013488 | 000000 | | |
| 013492 | 000000 | | |
| 013496 | 000000 | | |
| 013500 | 000000 | | |
| 013504 | 000000 | | |
| 013508 | 000000 | | |
| 013512 | 000000 | | |
| 013516 | 000000 | | |
| 013520 | 000000 | | |
| 013524 | 000000 | | |
| 013528 | 000000 | | |
| 013532 | 000000 | | |
| 013536 | 000000 | | |
| 013540 | 000000 | | |
| 013544 | 000000 | | |
| 013548 | 000000 | | |
| 013552 | 000000 | | |
| 013556 | 000000 | | |
| 013560 | 000000 | | |
| 013564 | 000000 | | |
| 013568 | 000000 | | |
| 013572 | 000000 | | |
| 013576 | 000000 | | |
| 013580 | 000000 | | |
| 013584 | 000000 | | |
| 01 | | | |

```

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
        200          ;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
        MOV      #BUF1,TABLE
        JSR      R5,XPSW          ;STORE EXPECTED PSW VALUE
        .WORD    204
        JSR      PC,GENR          ;SET UP GENERAL REGISTERS
        MOV      TABLE,R5
        SCC
        CLZ          ;SET ALL CONDITION CODES, EXCEPT
                   ;CLEAR "Z" CONDITION CODE
        SCANC          ;EXECUTE "SCANC"
                   ;CHECK RESULTS
                   ;STORE THE PSW
        MFPS      CCODES          ;CLEAR ALL UNUSEC BITS
        BIC      #177400,CCODES  ;CHECK PSW AGAINST EXPECTED VALUE
        CMP      3*EXPPSW,CCODES ;BR, IF EQUAL
        BEQ      64$
        ERROR    1
                   ;*****TEST 44 - 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       ;STORE BAD SP VALUE
        ERROR    2
                   ;*****TEST 44 - 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 44 - ERROR 3*****
                   ;R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL"
                   ;CHECK R5 UNCHANGED
        66$:
        CMP      TABLE,R5
        BEQ      67$
        ERROR    4
                   ;*****TEST 44 - ERROR 4*****
                   ;R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE"
        67$:
        MOV      SRCAD,R3          ;POINT R3 TO SOURCE STRING
        MOV      SRCLEN,R2        ;STORE SOURCE LENGTH IN R2

```

M07

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

TEST "SCAN" FAILS. ALL NON-MEMBER CHARACTERS IN STRING

SEG CC9:

| | | | | | | | |
|------|--------|--------|---------------|-------|-------|---------------|-----------------------------------|
| 3807 | 013400 | 111304 | | 69\$: | 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\$: | | | |

3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878

| | | | |
|--------|--------|--------|--------|
| 013440 | 000004 | | |
| 013442 | 105777 | 165072 | |
| 013446 | 100550 | | |
| 013450 | 026767 | 165074 | 165202 |
| 013456 | 001007 | | |
| 013460 | 032767 | 000001 | 165120 |
| 013466 | 001403 | | |
| 013470 | 005767 | 165100 | |
| 013474 | 001135 | | |
| 013476 | | | |
| 013476 | 004567 | 001510 | |
| 013502 | 000400 | | |
| 013504 | 016434 | | |
| 013506 | 177777 | | |
| 013510 | 177777 | | |
| 013512 | 000000 | | |
| 013514 | 012767 | 016434 | 165134 |
| 013522 | 012767 | 013614 | 165140 |
| 013530 | 012777 | 015356 | 165126 |
| 013536 | 005077 | 165124 | |
| 013542 | 004767 | 001564 | |
| 013546 | 013777 | 000554 | 165106 |
| 013554 | 004567 | 001526 | |
| 013560 | 000004 | | |
| 013562 | 106427 | 000000 | |
| 013566 | 052777 | 000100 | 165064 |
| 013574 | 004767 | 001346 | |
| 013600 | 010637 | 000676 | |
| 013604 | 016705 | 165046 | |
| 013610 | 000244 | | |
| 013612 | 000267 | | |
| 013614 | 076042 | | |
| 013616 | 106767 | 165050 | |
| 013622 | 032777 | 000100 | 165030 |
| 013630 | 001361 | | |
| 013632 | 042767 | 177400 | 165032 |
| 013640 | 023767 | 000674 | 165024 |
| 013646 | 001401 | | |
| 013650 | 104001 | | |

```

*****
;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
        MOVB     400           ;SOURCE LENGTH
        MOVB     BUF1          ;SOURCE ADDRESS
        MOVB     NXM           ;STORE NON-ZERO VALUES TO TEST R2 & R3
        MOVB     NXM           ;UNAFFECTED BY DIS STRING INSTRUCTIONS
        MOVB     0             ;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      @#$NULL,@TBUF  ;SEND CARRIAGE RETURN
        JSR      R5,XPSW        ;STORE EXPECTED PSW VALUE
        .WORD    04
        MTPS     #0            ;SET PSW TO ALLOW INTERRUPTS
        BIS      @100,@TCSR     ;ENABLE TTY INTERRUPTS
REPSCN: JSR      PC,GENR        ;SET UP GENERAL REGISTERS
        MOV      SP,@$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
        MFPS     CCODES         ;STORE THE PSW
        BIT      @100,@TCSR     ;IF INTERRUPT ARE DISABLED, THE INSTRUCTION WAS NOT INTE
        BNE      REPSCN        ;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 45 - ERROR 1*****
        ;PSW ERROR
        ;EXPECTED PSW IS STORED AT "EXPPSW"

```

```

3879                                     ;ACTUAL PSW IS STORED AT "CCODES"
3880 013652                                     64$:
3881 013652 023706 000676                   CMP      2#SAVR6,SP      ;VERIFY STACK POINTER IS RESTORED
3882 013656 001403                   BEQ      65$           ;BR IF OK
3883 013660 010637 000700                   MOV      SP,2#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                   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$:
3898 013706 016703 164734                   MOV      SRCAD,R3      ;POINT R3 TO SOURCE STRING
3899 013712 016702 164726                   MOV      SRCLN,R2     ;STORE SOURCE LENGTH IN R2
3900 013716 111304                                     68$:
3901 013720 042704 177400                   MOVSB   (R3),R4      ;USE SOURCE BYTE AS INDEX
3902 013724 136467 016434 164722           BIC      #177400,R4   ;CLEAR ANY BYTE SIGN EXTENSION
3903 013732 001003                   BITB     BUF1(R4),FILL ;"AND" TABLE ENTRY WITH MASK
3904 013734 105723                   BNE      69$           ;POINT TO NEXT SOURCE BYTE
3905 013736 005302                   TSTB     (R3)+       ;DECREMENT BYTE COUNT
3906 013740 001366                   DEC      R2          ;BR IF NOT FINISHED
3907 013742 020200                                     69$:
3908 013744 001401                   CMP      R2,R0      ;CHECK R0=REMAINING SOURCE LENGTH
3909 013746 104005                   BEQ      70$           ;*****TEST 45 - ERROR 5*****
3910                                     ;SCAN LENGTH ERROR
3911 013750 020301                                     70$:
3912 013752 001401                   CMP      R3,R1      ;CHECK R1=ADDRESS OF CHARACTER
3913 013754 104006                   BEQ      71$           ;*****TEST 45 - ERROR 5*****
3914                                     ;SCAN ADDRESS ERROR
3915 013756                                     71$:
3916                                     ;
3917 013756 106427 000200                                     70$:
3918 013762 016777 164700 164674           MTPS    #200      ;RESTORE PSW TO PRIORITY 7
                                     MOV      TPSW,3TVECT    ;RESTORE TRAP CATCHER

```

| PC | Instruction | Source | Target | Comment |
|------|---|--------|--------|--|
| 3919 | | | | |
| 3920 | | | | |
| 3921 | | | | |
| 3922 | TEST 46 | | | TEST "SPANC" WITH NON-MEMBER CHARACTER IN SOURCE, SL .GT. 0 AT CHARACTER |
| 3923 | | | | |
| 3924 | THIS TEST VERIFIES THAT "SPANC" INDICATES A NON-MEMBER | | | |
| 3925 | FOUND. | | | |
| 3926 | THE RESULT IS R1 EQUALS ADDRESS OF NON-MEMBER CHARACTER, R0 | | | |
| 3927 | EQUALS THE REMAINING SOURCE LENGTH, AND ALL | | | |
| 3928 | CONDITION CODES CLEAR. | | | |
| 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 164946 | | | |
| 3966 | 014106 001401 | | | |
| 3967 | 014110 104003 | | | |
| 3968 | | | | |
| 3969 | 014112 026705 164540 | | | |
| 3970 | 014116 001401 | | | |
| 3971 | 014120 104004 | | | |
| 3972 | | | | |
| 3973 | 014122 | | | |
| 3974 | 014122 016703 164520 | | | |

MAIN. MACY11 27(1006) 22-DEC-76 10:49 PAGE 94
 CVKATA.P11 22-DEC-76 10:47 745 TEST "SPANC" WITH NON-MEMBER CHARACTER IN SOURCE. SL .GT. 0 AT CHARACTER SEQ 0095

| | | | | | | | |
|------|--------|--------|---------------|-------|-------|---------------|-----------------------------------|
| 3975 | 014126 | 016702 | 164512 | | MOV | SRCLN,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 | BJF1(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 | 02020C | | 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\$: | CMP | 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 | | | | | | ;TEST 47 TEST "SPANC" FAILS ONLY MEMBER CHARACTERS IN SOURCE |
| 3994 | | | | | | ::***** |
| 3995 | | | | | | *PROPER TERMINATION OF THIS TEST IS R0=0, R1 EQUALS |
| 3996 | | | | | | *ADDRESS+1 OF LSB OF SOURCE, AND ALL CONDITION CODES |
| 3997 | | | | | | *CLEAR EXCEPT Z=1 |
| 3998 | | | | | | ::***** |
| 3999 | | | | | | ::***** |
| 4000 | | | | | | +ST47: SCOPE |
| 4001 | 014172 | 000004 | | | | JSR R5,PREP ;SET UP INSTRUCTION ARGUMENTS |
| 4002 | 014174 | 004567 | 001012 | | | 100 ;SOURCE LENGTH |
| 4003 | 014200 | 000100 | | | | BUF1 ;SOURCE ADDRESS |
| 4004 | 014202 | 016434 | | | | NXM ;STORE NON-ZERO VALUES TO TEST R2 & R3 |
| 4005 | 014204 | 177777 | | | | NXM ; UNAFFECTED BY DIS STRING INSTALCTIONS |
| 4006 | 014206 | 177777 | | | | 377 ;SEARCH [MASK FOR SCANC/SPANC] CHARACTER |
| 4007 | 014210 | 000377 | | | | |
| 4008 | 014212 | 012767 | 016434 | 164436 | | MOV #BUF1,TABLE |
| 4009 | 014220 | 004567 | 001062 | | | JSR R5,XPSW ;STORE EXPECTED PSW VALUE |
| 4010 | 014224 | 000204 | | | | .WORD 204 |
| 4011 | 014226 | 004767 | 000714 | | | JSR PC,GENR ;SET UP GENERAL REGISTERS |
| 4012 | 014232 | 016705 | 164420 | | | MOV TABLE,R5 |
| 4013 | 014236 | 000244 | | | | CLZ ;CLEAR "Z" CONDITION CODE |
| 4014 | 014240 | 000267 | | | | +SEV!SEZ!SEC ;SET ALL OTHER CONDITION CODES |
| 4015 | | | | | | ;EXECUTE "SPANC" |
| 4016 | | | | | | |
| 4017 | | | | | | |
| 4018 | 014242 | 076043 | | | | SPANC |
| 4019 | | | | | | ;CHECK RESULTS |
| 4020 | 014244 | 106767 | 164422 | | | MFPS CCODES ;STORE THE PSW |
| 4021 | 014250 | 042767 | 177400 | 164414 | | BIC #177400,CCODES ;CLEAR ALL UNUSED BITS |
| 4022 | 014256 | 023767 | 000674 | 164406 | | CMP 3#EXPPSW,CCODES ;CHECK PSW AGAINST EXPECTED VALLE |
| 4023 | 014264 | 001401 | | | | BEQ 64\$;BR, IF EQUAL |
| 4024 | 014266 | 104001 | | | | ERROR 1 ;*****TEST 47 - ERROR 1***** |
| 4025 | | | | | | PSW ERROR |
| 4026 | | | | | | ;EXPECTED PSW IS STORED AT "EXPPSW" |
| 4027 | | | | | | ;ACTUAL PSW IS STORED AT "CCODES" |
| 4028 | 014270 | | | | | |
| 4029 | 014270 | 023706 | 000676 | | | 64\$: CMP 3#SAVR6,SP ;VERIFY STACK POINTER IS RESTORED |
| 4030 | 014274 | 001403 | | | | BEQ 65\$;BR IF OK |
| 4031 | 014276 | 010637 | 000700 | | | MOV SP,3#BADR6 ;STORE BAD SP VALUE |
| 4032 | 014302 | 104002 | | | | ERROR 2 ;*****TEST 47 - ERROR 2***** |
| 4033 | | | | | | STACK POINTER NOT RESTORED BY INSTRUCTION |
| 4034 | | | | | | ;EXPECTED VALUE OF SP IS STORED AT "SAVR6" |
| 4035 | | | | | | ;ERRONEOUS VALUE IS AT "BADR6" |
| 4036 | 014304 | | | | | |
| 4037 | 014304 | 026704 | 164344 | | | 65\$: CMP FILL,R4 ;CHECK R4 UNCHANGED |
| 4038 | 014310 | 001401 | | | | BEQ 66\$;BR IF OK |
| 4039 | 014312 | 104003 | | | | ERROR 3 ;*****TEST 47 - ERROR 3***** |
| 4040 | | | | | | R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL" |
| 4041 | 014314 | 026705 | 164336 | | | 66\$: CMP TABLE,R5 ;CHECK R5 UNCHANGED |
| 4042 | 014320 | 001401 | | | | BEQ 67\$;BR IF OK |
| 4043 | 014322 | 104004 | | | | ERROR 4 ;*****TEST 47 - ERROR 4***** |
| 4044 | | | | | | R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE" |
| 4045 | 014324 | | | | | |
| 4046 | 014324 | 016703 | 164316 | | | MOV SRCAD,R3 ;POINT R3 TO SOURCE STRING |
| 4047 | 014330 | 016702 | 164310 | | | MOV SRCLEN,R2 ;STORE SOURCE LENGTH IN R2 |

MAIN: MACY11 27(1006) 22-DEC-76 10:49 PAGE 96
 CYRIL.F11 22-DEC-76 10:47 T47

TEST "SPANC" FAILS ONLY MEMBER CHARACTERS IN SOURCE

SEC 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 | | SEQ | 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 | | BEG | 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 | | BEG | 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
4120

014374 000004
014376 105777 164136
014402 100547
014404 026767 164140 164246
014412 001007
014414 032767 000001 164154
014422 001403
014424 005767 164144
014430 001134
014432
014432 004567 000554
014436 000400
014440 016434
014442 17777
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
        STB      $SWR      ;TEST BIT 7 OF SWR
        BMT      $EOP      ;SKIP TO NEXT TEST IF SET
        CMP      $TPS,TCR  ;IS SLL USED FOR INTERRUPTS THE CONSOLE?
        BNE      $PCCONT   ;BR. IF NOT & PERFORM INTERRUPTABILITY TEST
        BIT      $BITC,$ENV ;IF YES, CHECK IF ON APT
        BEQ      $PCCONT   ;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
        LCC      ;SOURCE LENGTH
        BUF1     ;SOURCE ADDRESS
        NXM      ;DESTINATION LENGTH
        NXM      ;DESTINATION ADDRESS
        L77      ;FILL CHARACTER

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

```

| | | | | | | | |
|------|--------|--------|--------|--------|-------|----------------|---|
| 4:20 | 014604 | | | 645: | | | :ACTUAL PSW IS STORED AT "CCODES" |
| 4:21 | 014604 | 023706 | 000676 | | CMP | 2#SAVR6, RP | :VERIFY STACK POINTER IS RESTORED |
| 4:22 | 014610 | 001403 | | | BEQ | 655 | :BR IF OK |
| 4:23 | 014612 | 010637 | 000700 | | MOV | SP, 2#BADRS | :STORE BAD SP VALUE |
| 4:24 | 014616 | 104002 | | | ERROR | 2 | :*****TEST 50 - ERROR 2***** |
| 4:25 | | | | | | | :STACK POINTER NOT RESTORED BY INSTRUCTION |
| 4:26 | | | | | | | :EXPECTED VALUE OF SP IS STORED AT "SAVR6" |
| 4:27 | | | | | | | :ERRONEOUS VALUE IS AT "BADRS" |
| 4:28 | 014620 | | | 655: | | | |
| 4:29 | 014620 | 026704 | 164030 | | CMP | FILL, R4 | :CHECK R4 UNCHANGED |
| 4:30 | 014624 | 001401 | | | BEQ | 665 | :BR IF OK |
| 4:31 | 014626 | 104003 | | | ERROR | 3 | :*****TEST 50 - ERROR 3***** |
| 4:32 | | | | | | | :R4 SHOULD STILL EQUAL CONTENTS OF LOCATION "FILL" |
| 4:33 | 014630 | 026705 | 164022 | 665: | CMP | TABLE, R5 | :CHECK R5 UNCHANGED |
| 4:34 | 014634 | 001401 | | | BEQ | 675 | :BR IF OK |
| 4:35 | 014636 | 104004 | | | ERROR | 4 | :*****TEST 50 - ERROR 4***** |
| 4:36 | | | | | | | :R5 SHOULD STILL EQUAL CONTENTS OF LOCATION "TABLE" |
| 4:37 | 014640 | | | 675: | | | |
| 4:38 | 014640 | 016703 | 164002 | | MOV | SRCAD, R3 | :POINT R3 TO SOURCE STRING |
| 4:39 | 014644 | 016702 | 163774 | | MOV | SRCLEN, R2 | :STORE SOURCE LENGTH IN R2 |
| 4:40 | 014650 | 111304 | | 685: | MOV | (R3), R4 | :USE SOURCE BYTE AS INDEX |
| 4:41 | 014652 | 042704 | 177400 | | BITB | #177400, R4 | :CLEAR ANY BYTE SIGN EXTENSION |
| 4:42 | 014656 | 136467 | 016434 | 163770 | BITB | BUF1, R4, FILL | :AND "TABLE" ENTRY WITH MASK |
| 4:43 | 014664 | 001403 | | | BEQ | 695 | |
| 4:44 | 014666 | 105723 | | | TEST | (R3)+ | :POINT TO NEXT SOURCE BYTE |
| 4:45 | 014670 | 005302 | | | DEC | R2 | :DECREMENT BYTE COUNT |
| 4:46 | 014672 | 001366 | | | BNE | 685 | :BR IF NOT FINISHED |
| 4:47 | 014674 | 020200 | | 695: | CMP | R2, R0 | :CHECK R0=REMAINING SOURCE LENGTH |
| 4:48 | 014676 | 001401 | | | BEQ | 705 | |
| 4:49 | 014700 | 104005 | | | ERROR | 5 | :*****TEST 50 - ERROR 5***** |
| 4:50 | | | | | | | :SPAN LENGTH ERROR |
| 4:51 | 014702 | 020301 | | 705: | CMP | R0, R1 | :CHECK R1=ADDRESS OF CHARACTER |
| 4:52 | 014704 | 001401 | | | BEQ | 715 | |
| 4:53 | 014706 | 104006 | | | ERROR | 6 | :*****TEST 50 - ERROR 6***** |
| 4:54 | | | | | | | :SPAN ADDRESS ERROR |
| 4:55 | 014710 | | | 715: | | | |
| 4:56 | | | | | | | |
| 4:57 | 014710 | 106427 | 000200 | 405: | MTPS | #200 | :RESTORE PSW TO PRIORITY 7 |
| 4:58 | 014714 | 163777 | 163746 | 163742 | MOV | TPSW, 37400 | :RESTORE TRAP CATCHER |

```

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

```

```

SCOPE
CLR          $STSNM          :: ZERO THE TEST NUMBER
INC          $PASS           :: INCREMENT THE PASS NUMBER
BIC          #100000,$PASS   :: DON'T ALLOW A NEG. NUMBER
DEC          (PC)+           :: LOOP
SEOPCT:      .WORD          1
BGT          $DOAGN          :: YES
MOV          (PC)+,2(PC)+    :: RESTORE COUNTER
SENDCT:      .WORD          1
SEOPCT
TYPE.        MSG            ; TYPE "END PASS"
SGETH2:      MOV          #2,PC :: GET MONITOR ADDRESS
BEG          $DOAGN          :: BRANCH IF NO MONITOR
CLR          -(SP)           :: INSURE THE "T" BIT IS CLEAR
MOV          #$(CLR.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     SAVE,RO      ::INSURE RO CONTAINS THE MONITORS
          BEQ     SDO,AGN      ::RETURN ADDRESS
          RESET                    ::CLEAR THE WORLD
SENDAC:  JSR     PC,RO        ::GO TO MONITOR
          NOP                     ::SAVE ROOM
          NOP                     ::FOR
          NOP                     ::ACT!!

```

```

SDOAGN:      NOT
              TRAP
              BTRC      #20,SP
              BIT       #BIT+12,CSWR
              BNE       IS
              COM       $*BIT
              BTRC      IS
              BTRC      #20,SP
              BTRC      #SLOOP,-,SP
              JUMP      TO START OF TEST
              RETURN--THIS IS CHANGED TO
              AN "RTT" IF "RTT" IS A LEGAL
              INSTRUCTION

```

```

SLOOP:      JMP      BPC0+          ;;INCRMS FOR
SRTNAD:     .WORD    BEGIN         ;;RETURN
STBIT:      .WORD    0              ;;"1" BIT STATE INDICATOR
SENUL:      .BYTE    -1,-1,0       ;;NULL CHARACTER STRING

```

ENDMSG: .REC12 (15) 12. END PAGE .

J08

MAIN MACY11 2711006 22-DEC-76 10:49 PAGE 100
END OF PASS ROUTINE

SEC 0101

10:16 01:04 000
10:16 01:04 000

.EVEN

```

4218
4219
4220
4221 015106 011637 000702      TZERO: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 0
4222 015112 104200              MOV      (SP),3*OLDPC ;GET PC+2 WHERE UNEXPECTED TRAP OCCURRED
4223                                ERROR    200 ;*****ERROR 200*****
4224                                ;UNEXPECTED TRAP TO LOCATION 0
4225                                ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4226 015114 000000              HALT ;PROGRAM MUST BE RESTARTED AT THIS POINT
4227
4228 015116 011637 000702      TIMTRP: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 4
4229 015122 104204              MOV      (SP),3*OLDPC ;GET PC+2 WHERE UNEXPECTED TIMEOUT TRAP OCCURRED
4230                                ERROR    204 ;*****ERROR 204*****
4231                                ;UNEXPECTED TRAP TO LOCATION 4
4232 015124 000000              HALT ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4233                                ;PROGRAM MUST BE RESTARTED AT THIS POINT
4234
4235 015126 011637 000702      ILLTRP: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 10
4236 015132 104210              MOV      (SP),3*OLDPC ;GET PC+2 WHERE UNEXPECTED ILLEGAL INSTRUCTION TRAP OCCURRED
4237                                ERROR    210 ;*****ERROR 210*****
4238                                ;UNEXPECTED TRAP TO LOCATION 10
4239 015134 000000              HALT ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4240                                ;PROGRAM MUST BE RESTARTED AT THIS POINT
4241
4242 015136 011637 000702      IOTTRP: ;ROUTINE TO REPORT UNEXPECTED TRAPS TO LOCATION 20
4243 015142 104220              MOV      (SP),3*OLDPC ;GET PC+2 WHERE UNEXPECTED IOT INSTRUCTION TRAP OCCURRED
4244                                ERROR    220 ;*****ERROR 220*****
4245                                ;UNEXPECTED TRAP TO LOCATION 20
4246 015144 000000              HALT ;"OLDPC" CONTAINS THE PC+2 OF THE TRAP OCCURRENCE
4247                                ;PROGRAM MUST BE RESTARTED AT THIS POINT
4248
4249
4250 015146 016700 163472      GENR: ;SUBROUTINE TO SETUP GENERAL REGISTERS FOR INSTRUCTION EXECUTION
4251 015152 016701 163470      MOV      SRCLN,R0 ;STORE SOURCE LENGTH
4252 015156 016702 163468      MOV      SRCAD,R1 ;STORE SOURCE ADDRESS
4253 015162 016703 163466      MOV      DSTLN,R2 ;STORE DESTINATION LENGTH
4254 015166 016704 163464      MOV      DSTAD,R3 ;STORE DESTINATION ADDRESS
4255 015172 016705 163462      MOV      FILL,R4 ;STORE FILL CHARACTER
4256 015176 010637 000676      MOV      TABLE,R5 ;
4257 015202 062737 000002 000676 MOV      SP,3*SAVR6 ;COPY STACK POINTER BEFORE INSTRUCTION EXECUTION
4258 015210 000207              ADD      #2,3*SAVR6 ;ADJUST SAVED SP BECAUSE OF JSR TO THIS ROUTINE
4259                                RTS      PC
4260
4261 015212 012567 163426      PREP: ;SUBROUTINE TO PREPARE INSTRUCTION PARAMETERS
4262 015216 012567 163424      MOV      (R5)+,SRCLN ;SET SOURCE LENGTH
4263 015222 012567 163422      MOV      (R5)+,SRCAD ;SET SOURCE ADDRESS
4264 015226 012567 163420      MOV      (R5)+,DSTLN ;SET DESTINATION LENGTH
4265 015232 012567 163418      MOV      (R5)+,DSTAD ;SET DESTINATION ADDRESS
4266 015236 000203              MOV      (R5)+,FILL ;SET FILL CHARACTER
4267                                RTS      R5 ;RETURN

```

MAIN: MAC111 27(1006) 22-DEC-76 10:49 PAGE 102
 C.KATA.F11 22-DEC-76 10:47 END OF PASS ROUTINE

SEG 0103

```

4267
4268
4269
4270 015240 016700 163406 CLOST: ;SUBROUTINE TO CLEAR DESTINATION & BOUNDARY BYTES
4271 015244 005300          MOV    DSTAD,RO      ;POINT RO TO DESTINATION AREA
4272 015246 105320          DEC    RO          ;CLEAR LOWER BOUNDARY BYTE OF DEST
4273 015250 016701 163374  MOV    DSTLN,R1      ;STORE DEST. BYTE COUNT IN R1
4274 015254 105020          CLAB   (RO)+        ;CLEAR DESTINATION
4275 015256 005301          DEC    R1          ;DECREMENT DEST. BYTE COUNT
4276 015260 001375          BNE    CONTC        ;BR IF NOT FINISHED
4277 015262 105020          CLAB   (RO)+        ;CLEAR UPPER BYTE BOUNDARY OF DEST
4278 015264 000207          RTS     PC          ;RETURN
4279
4280
4281 015266 012700 016434 GENSRC: ;SUBROUTINE TO GENERATE A SOURCE STRING
4282 015272 012502          MOV    #BUF1,RO      ;POINT RO TO THE PATTERN STORED
4283 015274 012501          MOV    (R5)+,R2      ;STORE SOURCE BYTE COUNT IN R2
4284 015276 112021          MOV    (R5)+,R1      ;POINT R1 TO THE SOURCE ADDRESS
4285 015300 005302          CON:  MOVB  (RO)+,(R1)+ ;TRANSFER PATTERN TO SOURCE
4286 015302 001375          DEC    R2          ;DECREMENT BYTE COUNT
4287 015304 000205          BNE    GENCON       ;BR IF NOT FINISHED
4288
4289
4290
4291 015306 012537 000674      ;SUBROUTINE TO RECORD EXPECTED PSW
4292 015312 106700          XPSW:  MOV    (R5)+,Q#EXPPSW ;STORE EXPECTED PSW VALUE
4293 015314 032700          MFPS   RO          ;
4294 015320 001403          BIT     #TBIT,RO      ;
4295 015322 052737 000020 000674 BEQ     IS          ;
4296 015330 000205          BIS     #TBIT,Q#EXPPSW ;OTHERWISE SET T-BIT IN EXPECTED PSW VALUE
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

```

4299
4300
4301      ;SUBROUTINE TO TEST FOR TRANSMIT DONE FLAG
4302 015332 005077 163346      TDONE: CLR      @TEMP      ;CLEAR A TIMER
4303 015336 105777 163316      IS:   TSTB     @TCR      ;IS SLU READY?
4304 015342 100404              BMI      RETN      ;BR IF READY
4305 015344 005277 163334      INC      @TEMP      ;OTHERWISE INCREMENT TIMER
4306 015350 001372              BNE      IS        ;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:   CMP      (SP),PC      ;WAS PC AT INSTRUCTION UNDER TEST?
4315 015356 021667 163306      BNE      SEND      ;BR, IF NO
4316 015362 001003              CKR4:  BIT      #177400,R4 ;IF YES, CHECK UPPER BYTE OF R4
4317 015364 032704 177400      BNE      CLRINT     ;IF ZERO, INSTRUCTION WAS NOT INTERRUPTED-TRY AGAIN
4318 015370 001004              SEND:   MOV      @#NULL,@TBUF ;SEND ANOTHER CHARACTER
4319 015372 013777 000554 163262      RTI              ;RETURN
4320 015400 000002              CLRINT: BIC      #100,@TCR   ;IF NON-ZERO, CLEAR INTERRUPT ENABLE
4321 015402 042777 000100 163250      RTI              ;CONTINUE INSTRUCTION
4322 015410 000002

```

```

4323
4324
4325 .SBTTL SCOPE HANDLER ROUTINE
4326
4327 *****
4328 ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
4329 ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
4330 ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
4331 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4332 ;*SW14=1 LOOP ON TEST
4333 ;*SW09=1 LOOP ON ERROR
4334 ;*SW08=1 LOOP ON TEST IN SWR<5:0>
4335 ;*CALL
4336 ;*
4337 SCOPE ;;SCOPE=IOT
4338
4339 $SCOPE:
4340 1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
4341 BNE $OVER ;;YES IF SW14=1
4342 ;*****START OF CODE FOR THE XOR TESTER*****
4343 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
4344 015412 032777 040000 163120 000004 MOV $ERRVEC,-(SP) ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
4345 015412 001065 000004 000004 MOV $,$ERRVEC ;;SAVE THE CONTENTS OF THE ERROR VECTOR
4346 015422 000416 000004 TST $177060 ;;SET FOR TIMEOUT
4347 015424 013746 000004 MOV (SP)+,$ERRVEC ;;TIME OUT ON XOR?
4348 015430 012737 015450 000004 BR $SVLAD ;;RESTORE THE ERROR VECTOR
4349 015436 005737 177060 000004 BR $SVLAD ;;GO TO THE NEXT TEST
4350 015442 012637 000004 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
4351 015446 000434 000004 MOV (SP)+,$ERRVEC ;;RESTORE THE ERROR VECTOR
4352 015450 022626 000004 BR 7$ ;;LOOP ON THE PRESENT TEST
4353 015452 012637 000004 6$: *****END OF CODE FOR THE XOR TESTER*****
4354 015456 000422 000004 BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
4355 015460 032777 000400 163052 BEQ 2$ ;;BR IF NO
4356 015466 001407 000004 MOV $SWR,-(SP) ;;SET DESIRED TEST NUM. FROM SWR
4357 015470 017746 163044 BIC $SWRMK,(SP) ;;STRIP AWAY UNDESIRED BITS
4358 015474 042716 000300 CMPB (SP)+,$TSTNM ;;ON THE RIGHT TEST?
4359 015500 122667 162775 BEQ $OVER ;;BR IF YES
4360 015504 001433 000004 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
4361 015506 105767 162771 BEQ $SVLAD ;;BR IF NO
4362 015512 001412 000004 BIT #BIT09,$SWR ;;LOOP ON ERROR?
4363 015514 032777 001000 163016 BEQ 4$ ;;BR IF NO
4364 015522 001404 000004 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
4365 015524 016767 162760 162754 BR $OVER
4366 015532 000420 000004 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
4367 015534 105067 162743 $SVLAD: INCB $TSTNM ;;COUNT TEST NUMBERS
4368 015540 105267 162736 MOV $TSTNM,$ESTN ;;SET TEST NUMBER IN APT MAILBOX
4369 015544 116767 162732 163020 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
4370 015552 011667 162730 MOV (SP),$LPERR ;;SAVE ERROR LOOP ADDRESS
4371 015556 011667 162726 CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
4372 015562 005067 162772 MOVB #1,$EMAX ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4373 015566 112767 000001 162721 $OVER: MOV $TSTNM,$DISPLAY ;;DISPLAY TEST NUMBER
4374 015574 016777 162702 162740 MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
4375 015602 016716 162700 RTI ;;FIXES PS
4376 015606 000002

```

```

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 $STSTM
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

```

```

4425
4426      .SBTTL  ERROR HANDLER ROUTINE
4427
4428      ;*****
4429      ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT.
4430      ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
4431      ;*AND TYPE OUT THE PC OF THE ERROR INSTRUCTION
4432      ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
4433      ;*SW15=1      HALT ON ERROR
4434      ;*SW13=1      INHIBIT ERROR TYPEOUTS
4435      ;*SW09=1      LOOP ON ERROR
4436      ;*CALL
4437      ;*      ERROR      N      ;ERR0P=EMT AND N=ERROR ITEM NUMBER
4438
4439      ;:*****
4440
4441      016010      $ERROR:
4442      016010      105267      162467      7$:      INCB      $ERFLG      ;SET THE ERROR FLAG
4443      016014      001775      BEQ      7$      ;DON'T LET THE FLAG GO TO ZERO
4444      016016      016777      162460      162516      MOV      $TSTNM,2DISP      ;DISPLAY TEST NUMBER AND ERROR FLAG
4445      016024      005267      162462      INC      $ERTTL      ;INC THE ERROR COUNT
4446      016030      011667      162462      MOV      (SP), $ERRPC      ;GET ADDRESS OF ERROR INSTRUCTION
4447      016034      162767      000002      162454      SUB      #2, $ERRPC
4448      016042      117767      162450      162444      MOVB      2$ERRPC, $ITEMB      ;STRIP AND SAVE THE ERROR ITEM CODE
4449      016050      032777      020000      162462      BIT      #BIT13, 2$SWR      ;SKIP TYPEOUT IF SET
4450      016056      001010      BNE      20$      ;SKIP TYPEOUTS
4451      016060      104401      000563      TYPE      $CRLF
4452      016064      016746      162426      MOV      $ERRPC, -(SP)      ;SAVE $ERRPC FOR TYPEOUT
4453      ;ERROR ADDRESS
4454      016070      004767      000160      JSR      PC, TYPECT      ;GO TYPE--OCTAL ASCII(ALL DIGITS,
4455      016074      104401      000563      TYPE      $CRLF
4456      016100      122767      000001      162500      20$:      CMPB      #APTENV, $ENV      ;RUNNING IN APT MODE
4457      016106      001011      BNE      2$      ;NO, SKIP APT ERROR REPORT
4458      016110      005767      162452      21$:      TST      $MSGTYPE      ;FINISHED LAST MESSAGE?
4459      016114      001375      BNE      21$      ;IF NOT, WAIT
4460      016116      116767      162372      162444      MOVB      $ITEMB, $FATAL      ;REPORT ERROR NUMBER TO APT
4461      016124      005267      162436      INC      $MSGTYPE      ;TELL APT TO TAKE ERROR
4462      016130      000777      22$:      BR      22$      ;APT ERROR LOOP
4463      016132      005777      162402      2$:      TST      2$SWR      ;HALT ON ERROR
4464      016136      100001      BPL      3$      ;SKIP IF CONTINUE
4465      016140      000000      HALT      ;HALT ON ERROR!
4466      016142      032777      001000      162370      3$:      BIT      #BIT09, 2$SWR      ;LOOP ON ERROR SWITCH SET
4467      016150      001402      BEQ      4$      ;BR IF NO
4468      016152      016715      162332      4$:      MOV      $LPERR, (SP)      ;FUDGE RETURN FOR LOOPING
4469      016156      005767      162376      TST      $ESCAPE      ;CHECK FOR AN ESCAPE ADDRESS
4470      016162      001402      BEQ      5$      ;BR IF NONE
4471      016164      016716      162370      MOV      $ESCAPE, (SP)      ;FUDGE RETURN ADDRESS FOR ESCAPE
4472      016170      022737      015010      000042      5$:      CMP      #SENAD, 2#42      ;ACT-11 AUTO-ACCEPT?
4473      016176      001001      BNE      6$      ;BRANCH IF NC
4474      016200      000000      HALT      ;YES
4475      016202      000002      6$:      RTI      ;RETURN

```

```

4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495 016204 010046
4496 016206 017600 000002
4497 016212 132767 000040 162357
4498 016220 001003
4499 016222 112046
4500 016224 001005
4501 016226 005726
4502 016230 012600
4503 016232 062716 000002
4504 016236 000002
4505 016240 105777 162304
4506 016244 100375
4507 016246 112677 162300
4508 016252 000763

      .SETTL TYPE ROUTINE

      ;*****
      ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
      ;*
      ;*CALL:
      ;*1) USING A TRAP INSTRUCTION
      ;*   TYPE      .MESADR      ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
      ;*OR
      ;*   TYPE
      ;*   MESADR
      ;*
      ;*****

$TYPE:  MOV      RO, -(SP)      ;SAVE RO
        MOV      @2(SP), RO    ;GET ADDRESS OF ASCIZ STRING
        BITB     #APCSUP, $ENVM ;IS CONSOLE SUPPRESSED?
        BNE      60$           ;YES, SKIP TYPE OUT
2$:     MOVB      (RO)+, -(SP)   ;PUSH CHARACTER TO BE TYPED ONTO STACK
        BNE      4$           ;BR IF IT ISN'T THE TERMINATOR
        TST      (SP)+         ;IF TERMINATOR POP IT OFF THE STACK
60$:    MOV      (SP)+, RO      ;RESTORE RO
3$:     ADD      #2, (SP)       ;ADJUST RETURN PC
        RTI
4$:     TSTB      @5TPS
        BR       4$
        MOVB      (SP)+, @5TPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
        BR       2$

```

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

SEG 0109

```

4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521 016254 112767 000005 000070
4522 016262 010446
4523 016264 010546
4524 016266 016605 000006
4525 016272 005004
4526 016274 006105
4527 016276 000404
4528 016300 006105
4529 016302 006105
4530 016304 006105
4531 016306 010504
4532 016310 006104
4533 016312 042704 177770
4534 016316 052704 000060
4535 016322 110467 000022
4536 016326 104401 016350
4537 016332 105367 000014
4538 016336 002360
4539 016340 012605
4540 016342 012604
4541 016344 012616
4542 016346 000207
4543 016350 000
4544 016351 000
4545 016352 000000
4546

;SBTTL BINARY TO OCTAL (ASCII) AND TYPE
;*****
;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;OCTAL (ASCII) NUMBER AND TYPE IT.
;CALL:
;* MOV NUM, -(SP) ;NUMBER TO BE TYPED
;* JSR PC, TYPOCT ;CALL FOR TYPEOUT
;*****
;*****
TYPOCT: MOVB #5, $OCNT ;SET ITERATION COUNT
        MOV R4, -(SP) ;SAVE R4
        MOV R5, -(SP) ;SAVE R5
        MOV 6(SP), R5 ;PICKUP THE INPUT NUMBER
        CLR R4 ;CLEAR THE OUTPUT WORD
1$: ROL R5 ;ROTATE MSB INTO "C"
    BR 3$ ;GO DO MSB
2$: ROL R5 ;FORM THIS DIGIT
    ROL R5
    ROL R5
    MOV R5, R4
3$: ROL R4 ;GET LSB OF THIS DIGIT
    BIC #177770, R4 ;GET RID OF JUNK
    BIS #0, R4 ;MAKE THIS DIGIT ASCII
    MOVB R4, 8$ ;SAVE FOR TYPING
    TYPE 8$ ;GO TYPE THIS DIGIT
7$: DECB $OCNT ;COUNT BY 1
    BGE 2$ ;BR IF MORE TO DO
6$: MOV (SP)+, R5 ;RESTORE R5
    MOV (SP)+, R4 ;RESTORE R4
    MOV (SP)+, (SP) ;SET THE STACK FOR RETURNING
    RTS PC ;RETURN
8$: .BYTE 0 ;STORAGE FOR ASCII DIGIT
    .BYTE 0 ;TERMINATOR FOR TYPE ROUTINE
$OCNT: .WORD 0 ;OCTAL DIGIT COUNTER
  
```

```

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
4591

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

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

```

[illegible]

MAIN. MACY:1 27010061 22-DEC-76 10:49 PAGE :12
CYRIA.P11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

MAY 27 1966 22-DEC-76 10:49 PAGE 114
 22-DEC-76 10:47

CROSS REFERENCE TABLE -- USER SYMBOLS

SEC 0114

| | | | | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 907# | 910 | 911# | 917 | 919# | 923 | 924# | 928 | 929# | 935# | 956 | 967# | 974 |
| 975# | 981 | 982# | 985 | 986# | 992 | 993# | 997 | 998# | 1001 | 1002# | 1005 | 1008# |
| 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 | 1229 | 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 | 1439# | 1442 |
| 1445 | 1446# | 1455 | 1456# | 1461 | 1462# | 1468# | 1500 | 1501# | 1508 | 1509# | 1515 | 1516# |
| 1519 | 1520# | 1520 | 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 |
| 1790# | 1793 | 1794# | 1799 | 1791# | 1795 | 1796# | 1799 | 1800# | 1803 | 1804# | 1810 | 1811# |
| 1817 | 1820 | 1821# | 1828 | 1829# | 1835 | 1866 | 1867# | 1874 | 1875# | 1881 | 1882# | 1886 |
| 1897# | 1890 | 1891# | 1894 | 1895# | 1899 | 1900# | 1903 | 1904# | 1909 | 1910# | 1915 | 1919 |
| 1919# | 1925 | 1928 | 1929# | 1936 | 1937# | 1942 | 1975 | 1976# | 1976 | 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# | 2435 |
| 2426# | 2430 | 2463 | 2464# | 2471 | 2472# | 2478 | 2479# | 2482 | 2483# | 2496 | 2497# | 2500 |
| 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# | 2632 |
| 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 | 3181 | 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 | 3369# | 3379 | 3380# |
| 3382 | 3384# | 3392 | 3420 | 3421# | 3429 | 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 | 3589 |
| 3609# | 3646 | 3647# | 3653 | 3654# | 3657 | 3658# | 3669 | 3670# | 3673 | 3674# | 3682 | 3683# |
| 3711# | 3718 | 3719# | 3725 | 3726# | 3729 | 3730# | 3743 | 3744# | 3747 | 3748# | 3753 | 3754# |
| 3784 | 3791 | 3792# | 3798 | 3799# | 3802 | 3803# | 3816 | 3817# | 3820 | 3821# | 3826 | 3827# |
| 3877 | 3884 | 3885# | 3891 | 3892# | 3895 | 3896# | 3904 | 3910# | 3917 | 3914# | 3921 | 3922# |
| 3953 | 3960 | 3961# | 3967 | 3968# | 3971 | 3972# | 3985 | 3986# | 3989 | 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# | | |
| 4223 | 4583# | | | | | | | | | | | |
| 476# | 758 | 854 | 2299 | 2301 | 2371 | 2611 | 2695 | 3061 | 3062 | 3129 | 3130 | 3196 |
| 3219# | 3262 | 3263# | 3264 | 3324 | 3325 | 3406 | 3407 | 3472 | 3474 | 3543 | 3544 | 3545 |
| 3606 | 3607# | 3694 | 3695 | 3765 | 3766 | 3848 | 3849 | 3925 | 3926 | 4007 | 4008 | 4090 |

 10000
 10000

CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

DATA MACY11 27(1005) 22-DEC-76 10:49 PAGE 117
SVKATA.F11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

[illegible]

STKB = 000546
STKS = 000544
STN = 000051

| | | | | | | | | | | | | |
|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 574* | | | | | | | | | | | | |
| 573* | | | | | | | | | | | | |
| 474* | 748 | 756* | 775 | 783 | 790 | 796 | 800 | 804 | 809 | 813 | 920 | 924 |
| 827 | 832 | 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 | 2275 |
| 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* | 3875 | 3884 |
| 3891 | 3895 | 3909 | 3913 | 3921 | 3932* | 3952 | 3960 | 3967 | 3971 | 3985 | 3989 | 3995 |
| 4004* | 4024 | 4032 | 4039 | 4043 | 4057 | 4061 | 4067 | 4078* | 4117 | 4125 | 4132 | 4135 |

```

STPB      000552
STPFLG    000557
STPS      000550
STRAP     016354
STRAP2    016376
STRP      = 000002
STRPAD    016410
STSTM     000404
STSTNM    000502
STYPBN=   *****
STYPCS=   *****
STYPE     016204
STYPOC=   *****
SJUNIT    000600
SJUNITM   000410
JSWR      000612
SVECT1    000636
SVECT2    000640
$XTETR    C15422
$GET4=     000001

```

| | | | | | | | | | |
|-------|-------|------|------|-------|------|------|------|-------|------|
| 4130 | 4134 | | | | | | | | |
| 576# | 4507* | | | | | | | | |
| 580# | | | | | | | | | |
| 575# | 1618 | 2171 | 2966 | 3314 | 3596 | 3838 | 4080 | 4505 | |
| 688 | 4556# | | | | | | | | |
| 4567# | 4578 | | | | | | | | |
| 4571# | 4580# | | | | | | | | |
| 4561 | 4578# | | | | | | | | |
| 538# | | | | | | | | | |
| 553# | 4172* | 4330 | 4357 | 4366* | 4367 | 4372 | 4375 | 4401* | 4444 |
| 4580 | | | | | | | | | |
| 4580 | | | | | | | | | |
| 4495# | 4571 | 4579 | | | | | | | |
| 4580 | | | | | | | | | |
| 596# | | | | | | | | | |
| 540# | | | | | | | | | |
| 603# | | | | | | | | | |
| 628# | 730 | | | | | | | | |
| 629# | | | | | | | | | |
| 4342# | | | | | | | | | |
| 4184# | 4189 | | | | | | | | |

.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

\$40CAT= ***** 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 | | | | | | | | | | | | |

.\$X = 000400

| | | | | | | | | | | | | | | | | |
|------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|--|
| CATCH | 330* | 4221 | 4228 | 4235 | 4242 | | | | | | | | | | | |
| COMMON | 467* | | | | | | | | | | | | | | | |
| HL | 332* | 2297 | 2369 | 2441 | 2535 | 2607 | 2691 | 2765 | 2851 | 2973 | | | | | | |
| | 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 | 1129 | |
| | 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 | 2569 | 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 | 3279 | |
| | 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 | | | | | | | | | |
| END COMMON | 467* | | | | | | | | | | | | | | | |
| HL | 332* | 4181 | | | | | | | | | | | | | | |
| | 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 | 1129 | |
| | 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 | 2569 | 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 | 3279 | |
| | 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 | 1129 | |
| | 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 | |

CROSS REFERENCE TABLE -- MACRO NAMES

| | | | | | | | | | | | | | | | |
|--------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| | 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 | 2317 | 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 | 3999 | 4024 | 4032 | 4039 | 4043 | 4057 |
| | 4061 | 4117 | 4125 | 4132 | 4136 | 4150 | 4154 | 4222 | 4229 | 4236 | 4243 | 4307 | | | |
| ESCAPE | 467# | | | | | | | | | | | | | | |
| GETPRI | 467# | 4197 | | | | | | | | | | | | | |
| GETSWR | 467# | | | | | | | | | | | | | | |
| INT | 348# | 4314 | | | | | | | | | | | | | |
| BYTE | 346# | 917 | 914 | 1009 | 1106 | 1217 | 1434 | 1543 | 1695 | 1807 | 1906 | 2018 | 2119 | 2248 | |
| MLOC | 352# | 3097 | 3165 | 3233 | 3370 | 3442 | 3515 | 3660 | | | | | | | |
| MSGPN | 353# | 3732 | 3805 | 3898 | 3974 | 4046 | 4139 | | | | | | | | |
| MSG1 | 748# | 750 | | | | | | | | | | | | | |
| MSG10 | 1834# | 1837 | | | | | | | | | | | | | |
| MSG10A | 1942# | 1944 | | | | | | | | | | | | | |
| MSG10B | 2046# | 2048 | | | | | | | | | | | | | |
| MSG11 | 2156# | 2159 | | | | | | | | | | | | | |
| MSG11A | 2287# | 2289 | | | | | | | | | | | | | |
| MSG11B | 2356# | 2359 | | | | | | | | | | | | | |
| MSG11C | 2430# | 2432 | | | | | | | | | | | | | |
| MSG14 | 2523# | 2526 | | | | | | | | | | | | | |
| MSG14A | 2597# | 2599 | | | | | | | | | | | | | |
| MSG15 | 2753# | 2756 | | | | | | | | | | | | | |
| MSG15A | 2842# | 2844 | | | | | | | | | | | | | |
| MSG15B | 2919# | 2921 | | | | | | | | | | | | | |
| MSG16 | 2680# | 2683 | | | | | | | | | | | | | |
| MSG17 | 2953# | 2956 | | | | | | | | | | | | | |
| MSG2 | 841# | 844 | | | | | | | | | | | | | |
| MSG22 | 3047# | 3050 | | | | | | | | | | | | | |
| MSG22A | 3115# | 3117 | | | | | | | | | | | | | |
| MSG23 | 3183# | 3186 | | | | | | | | | | | | | |
| MSG24 | 3251# | 3254 | 3535 | | | | | | | | | | | | |
| MSG25 | 3302# | 3305 | | | | | | | | | | | | | |
| MSG26 | 3391# | 3394 | | | | | | | | | | | | | |
| MSG27 | 3460# | 3463 | | | | | | | | | | | | | |
| MSG28 | 3583# | 3586 | | | | | | | | | | | | | |
| MSG29 | 3681# | 3684 | | | | | | | | | | | | | |
| MSG29A | 3753# | 3755 | | | | | | | | | | | | | |
| MSG3 | 935# | 938 | | | | | | | | | | | | | |
| MSG30 | 3825# | 3828 | | | | | | | | | | | | | |
| MSG30A | 3921# | 3923 | | | | | | | | | | | | | |
| MSG31 | 3994# | 3997 | | | | | | | | | | | | | |
| MSG32 | 4066# | 4069 | | | | | | | | | | | | | |
| MSG34 | 1034# | 1037 | | | | | | | | | | | | | |
| MSG35 | 1111# | 1114 | | | | | | | | | | | | | |
| MSG36 | 1252# | 1255 | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | |
|---------|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 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 | | | | |
| REG | 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 |
| | 2158 | 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# | | | | | | | | | | | | | | |
| SKIPINT | 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 | 2691 | 2692 |
| | 2688 | 2689 | 2754 | 2756 | 2762 | 2763 | 2842 | 2844 | 2849 | 2849 | 2919 | 2921 | 2924 | 2925 | 2934 |
| | 2956 | 2961 | 2962 | 3048 | 3050 | 3055 | 3056 | 3115 | 3117 | 3123 | 3124 | 3184 | 3196 | 3197 | 3198 |
| | 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 |
| SWRSL | 467# | | | | | | | | | | | | | | |
| TGRZ | 340# | 794 | 891 | 995 | 1083 | 1195 | 1314 | 1411 | 1513 | 1672 | 1793 | 1884 | 2004 | 2097 | 2226 |
| TRMTRP | 4571# | | | | | | | | | | | | | | |
| TRUNC | 345# | 1522 | 1785 | 1997 | | | | | | | | | | | |
| TSLOC | 344# | 771 | 869 | 962 | 1061 | 1173 | 1281 | 1399 | 1496 | 1647 | 1760 | 1862 | 1971 | 2075 | 2201 |
| | 2311 | 2385 | 2459 | 2549 | 2624 | 2708 | 2793 | 2870 | 2996 | 3071 | 3139 | 3207 | 3274 | 3342 | 3416 |
| | 3489 | 3555 | 3632 | 3706 | 3779 | 3870 | 3948 | 4020 | 4111 | | | | | | |
| TESTP | 335# | 790 | 878 | 971 | 1070 | 1182 | 1290 | 1398 | 1505 | 1659 | 1769 | 1871 | 1980 | 2084 | 2213 |
| | 2320 | 2394 | 2468 | 2558 | 2632 | 2717 | 2792 | 2879 | 3007 | 3090 | 3149 | 3216 | 3293 | 3353 | 3425 |
| | 3498 | 3564 | 3643 | 3715 | 3798 | 3881 | 3957 | 4029 | 4122 | | | | | | |
| TRAP | 355# | 1577 | 2927 | | | | | | | | | | | | |
| TRY | 349# | 1632 | 2185 | 2979 | 3327 | 3616 | 3852 | 4094 | | | | | | | |

MAIN. MACY11 27(1005) 22-DEC-76 10:49 PAGE 126
 DYKAI.A.P11 22-DEC-76 10:47 CROSS REFERENCE TABLE -- MACRO NAMES

SEG 0125

| | | | | | | | | | | | | | | | |
|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| TYPBIN | 4670 | | | | | | | | | | | | | | |
| TYPDEC | 4670 | | | | | | | | | | | | | | |
| TYPNAM | 4670 | | | | | | | | | | | | | | |
| TYPNUM | 4670 | | | | | | | | | | | | | | |
| TYPJCS | 4670 | | | | | | | | | | | | | | |
| TYPJCT | 4670 | | | | | | | | | | | | | | |
| TYPJXT | 4670 | | | | | | | | | | | | | | |
| JBYTE | 3470 | 833 | 925 | 1026 | 1134 | 1245 | 1459 | 1560 | 1720 | 1826 | 1934 | 2037 | 2147 | 2276 | |
| JDEST | 3360 | 824 | 1017 | 1114 | 1125 | 1225 | 1236 | 1442 | 1551 | 1703 | 1817 | 1915 | 1925 | 2028 | 2129 |
| | 2138 | 2257 | 2267 | | | | | | | | | | | | |
| SSCHRE | 5440 | | | | | | | | | | | | | | |
| SSCHTM | 5440 | | | | | | | | | | | | | | |
| SSSCHN | 4670 | | | | | | | | | | | | | | |
| SSNEWY | 4670 | 748 | 842 | 936 | 1035 | 1141 | 1253 | 1359 | 1469 | 1559 | 1604 | 1733 | 1835 | 1942 | 2046 |
| | 2137 | 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 | 4570 | | | | | | | | | | | | | | |
| SSK10 | 4570 | | | | | | | | | | | | | | |
| SSK11 | 3230 | 357 | | | | | | | | | | | | | |
| SSK12 | 3230 | 500 | | | | | | | | | | | | | |
| SSK13 | 3230 | 513 | | | | | | | | | | | | | |
| SSK14 | 3230 | | | | | | | | | | | | | | |
| SSK15 | 3230 | 520 | | | | | | | | | | | | | |
| SSK16 | 3230 | 481 | | | | | | | | | | | | | |
| SSK17 | 3230 | 544 | | | | | | | | | | | | | |
| SSK18 | 3230 | 4162 | | | | | | | | | | | | | |
| SSK19 | 3230 | 4376 | | | | | | | | | | | | | |
| SSK20 | 3230 | 4325 | | | | | | | | | | | | | |
| SSK21 | 3230 | 4548 | | | | | | | | | | | | | |

. ABS. 017434 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DYKAI.A.DYKAI.A/SOL/CRF=DYKAI.A.P11
 RUN-TIME: 62 52 5 SECONDS
 RUN-TIME RATE: 745/121=6.1
 CORE USED: 11K 141 PAGES