PDP11

INSTRUCTION EXERCISER

MD-11-DZQKC-F

EP-DZQKC-F-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A

.REM !

IDENTIFICATION

PRODUCT CODE: [illegible]
PRODUCT NAME: 11 FAMILY INSTRUCTION EXERCISER
DATE: [illegible]
MAINTAINER: DIAGNOSTIC GROUP

1.0 ABSTRACT

THIS DIAGNOSTIC PROGRAM IS DESIGNED TO BE A COMPREHENSIVE CHECK OF THE PDP11/05 AND PDP11/20 PROCESSORS. THE PROGRAM EXECUTES EACH INSTRUCTION IN ALL ADDRESS MODES AND INCLUDES TESTS FOR TRAPS AND THE TELETYPE INTERRUPT SEQUENCE. THE PROGRAM DOES NOT TEST INSTRUCTIONS NOT COMMON TO THE 11/20 OR 11/05. THE PROGRAM RELOCATES THE TEST CODE THROUGHOUT MEMORY 0-28K.

2.0 REQUIREMENTS

2.1 EQUIPMENT

PDP11 FAMILY CENTRAL PROCESSOR
OPTIONAL - KW11-L (LINE CLOCK)

2.2 STORAGE

THE PROGRAM USES ALL OF THE FIRST 4K OF MEMORY (EXCLUDING THAT AREA OF MEMORY RESERVED FOR THE LOADERS).

2.3 PRELIMINARY PROGRAMS

NONE, HOWEVER, THE EMT AND TRAP INSTRUCTION SHOULD BE VERIFIED BEFORE RUNNING.

3.0 LOADING AND STARTING PROCEDURE

LOAD PROGRAM USING ABS LOADER
IF THE CONSOLE TTY IS A SERIAL LA30 OR A VT05 FILLER CHARACTERS MAY BE REQUIRED. DEPOSIT INTO LOCATION 1002 (FILLS) A 4400.
LOAD ADDRESS = 200
PRESS START
SET OPERATING SWITCHES
PASS COUNT IS PRINTED AFTER EACH PASS (SEE SEC 5.4)
"DZ3KC DONE" IS PRINTED WHEN DONE (SEE SEC 7.1)

4.0 SWITCH SETTINGS

SW15 HALT ON ERROR... THIS SWITCH WHEN SET WILL HALT THE PROCESSOR WHEN AN ERROR IS DETECTED. THE PC+2 AND THE CURRENT STATUS AT THE TIME OF THE ERROR IS STORED ON THE STACK (R6). IF THIS SWITCH IS SET BEFORE AN ERROR IS DETECTED THE PROGRAM HALTS AS DESCRIBED ABOVE. THE PROGRAM MAY BE HALTED AFTER THE ERROR TYPEOUT OCCURS BY SETTING SW15 AFTER THE TYPEOUT BEGINS.

SW14 LOOP SUBTEST... THIS SWITCH WHEN SET LOOPS THE CURRENT SUBTEST RUNNING REGARDLESS OF ERROR.

SW13 INHIBIT ERROR PRINTOUT - THIS SWITCH WHEN SET INHIBITS THE ERROR PRINTOUT.

SW12 INHIBIT RELOCATION... THIS SWITCH WHEN SET CAUSES THE PROGRAM TO BE EXECUTED ONLY IN THE FIRST 4K OF MEMORY. THIS SWITCH CANNOT BE SET WHEN THE PROGRAM IS RUNNING.

SW11 INHIBIT SUBTEST ITERATION... THIS SWITCH WHEN SET INHIBITS SUBTEST REITERATION. NORMALLY EACH SUBTEST IS EXECUTED 8 TIMES BEFORE THE NEXT SUBTEST IS RUN.

SETTING SW11 CAUSES EACH TEST TO BE EXECUTED ONCE BEFORE STARTING THE NEXT SUBTEST.

SW10 RING BELL ON ERROR... THIS SWITCH WHEN SET WILL RING THE BELL WHEN AN ERROR IS DETECTED.

SW7 INHIBIT ALL BUT ERROR TYPEOUTS...THIS SWITCH WHEN RESET (0) INHIBITS THE END OF PASS TYPEOUT (ICNT=XXXX) AND THE END OF PROGRAM TYPEOUT (DZQKC DONE)

5.0 ERRORS

IF AN ERROR IS DETECTED THE PROGRAM WILL TRAP TO THE ERROR HANDLING ROUTINE (ERRCR). IF ENABLED THIS ROUTINE WILL BYTE THE PC AND THE PROCESSER STATUS AT THE TIME OF THE ERROR. ALSO (IF REQUIRED) THE ORIGINAL PC (WHERE THE PC WAS RELOCATED FROM).

5.0.1 ERROR PRINTOUT FORMAT

ICNT=AAAA PC=BBBBBB PSW=DDDDDD
OR
ICNT=AAAA PC=BBBBBB PSW=DDDDDD PC RELOCATED FROM CCCCCC

WHERE: AAAA=PASS COUNT
BBBBBB=PC AT THE TIME OF THE ERROR
CCCCCC=PC OF THE ORIGINAL CODE RELOCATED
DDDDDD=PSW AT THE TIME OF THE ERROR.

5.1 PARITY ERROR DECTECTION

IF A PARITY ERROR IS DETECTED THE PROGRAM WILL TYPE A MESSAGE"PARITY ERROR" AND SCAN MEMORY FOR THE PARITY ERROR. WHEN THE FAILING ADDRESS IS LOCATED THE PROGRAM WILL HALT WITH THE VALUE OF THE ADDRESS+2 IN R0.

5.2 ERROR LOOPING

THE SUBTEST DETECTING THE ERROR MAY BE LOOPED INDEFINITELY BY SETTING SW14. SETTING SW13 WILL INHIBIT THE TYPEOUT AND ALLOW SCOPING THE FAULTY SIGNAL(S).

5.3 UNPREDICTED ERRORS

THE PROGRAM MAY ON OCCASSION DETECT A MEMORY ERROR THE RESULTS OF WHICH WERE NOT PREDICTABLE IN WHICH CASE THE PROGRAM MAY BEHAVE UNPREDICTABLY. WHEN THIS HAPPENS THE USER MUST RETRACE THE PROGRAM STEPS TO RESOLVE WHERE THE ERROR OCCURRED. THE FOLLOWING ITEMS SHOULD BE CONSIDERED AND MAY BE OF USE WHEN RETRACING A FAILURE OF THIS NATURE.

1. HALT THE PROGRAM (IF NECESSARY)

2. EXAMINE RELR1
   ADDRESS RELR1 (1006) CONTAINS THE UNRELOCATED VALUE OF
   THE PC OF THE LAST TEST THAT WAS SUCCESSFULLY EXECUTED.

3. EXAMINE FACTOR

ADDRESS FACTOR (1004) CONTAINS THE RELOCATION FACTOR.

4. EXAMINE ALL LOCATIONS STARTING WITH THE ADDRESS SPECIFIED IN R1/R11(IF PSW BIT11 =0/1) COMPARING THEIR CONTENTS WITH THE CONTENTS OF THE CORRESPONDING UNRELOCATED CODE (SPECIFIED IN 1006) AS SHOWN IN THE LISTING. EXAMINE AND COMPARE UNTIL EITHER A DIFFERENCE IN INSTRUCTION (I.E., THE ERROR) OR THE NEXT 'SCOPE' IS SEEN.

IF THE PROGRAM TRAPS AND HALTS AT A TRAP/INTERRUPT VECTOR+2 (NOTE: THE PDP-11/45 WILL DISPLAY THE ADDRESS OF THE HALT+2 I.E., A FALSE TRAP TO 4 WILL DISPLAY 10).

1A. EXAMINE THE STACK (R6)

THE TOP WORD ON THE STACK CONTAINS THE PC AT THE TIME OF THE TRAP. IF THE PC IS GREATER THAN 20000, THEN

2A. EXAMINE LOCATION 1002 (FACTOR)

THIS LOCATION CONTAINS THE PROGRAM RELOCATION FACTOR WHICH, WHEN SUBTRACTED FROM THE PC GIVES THE PC OF THE ORIGINAL CODE.

6.0 SUBROUTINE ABSTRACTS

6.1 SCOPEA

THE SCOPEA ROUTINE IS ENTERED BY THE SCOPE (EMT) INSTRUCTION AND IS EXECUTED AT THE START OF EACH SUBTEST. THE ROUTINE MONITORS SW14, SW11 AND SW 8 AND TAKES APPROPRIATE ACTION. ALSO, THIS ROUTINE STORES IN R1/R11 THE FIRST ADDRESS OF THE SUBTEST BEING ENTERED.

6.2 ERROR

THE ERROR ROUTINE IS ENTERED BY THE HLT (TRAP) INSTRUCTION, AND IS EXECUTED WHEN A PREDICTABLE ERROR IS DETECTED. THIS ROUTINE MONITORS SW15, SW13, AND SW10.

6.3 RELOC

THE RELOC ROUTINE IS ENTERED BY A MOV RELOC,PC INSTRUCTION. THIS ROUTINE RELOCATES THE PROGRAM CODE THROUGHOUT MEMORY, AND 'JUMPS' TO THE RELOCATED CODE AFTER IT HAS BEEN MOVED SUCCESSFULLY. IF THE CODE CANNOT BE RELOCATED (BECUASE OF INSUFFICIENT MEMORY) THE ROUTINE 'JUMPS' TO THE NEXT SECTION OF UNRELOCATED PROGRAM CODE. THE CODE MOVED IS LESS THAN 1K (4000) BYTES). AT THE START AND END OF EACH SECTION OF CODE TO BE MOVED ARE A SECTION OF CODE WHICH ESTABLISHES THE FIRST ADDRESS OF THE CODE TO BE MOVED, AND SETS A SCOPE POINTER (R1/R11) AND, ALSO A SECTION WHICH ESTABLISHES THE LAST ADDRESS AND 'JUMPS' TO THE

```
   230                    RELOCATION  (RELOC)  ROUTINE.  EACH  SECTION  OF  CODE  IS
   231                    IDENTIFIED AS SHOWN BELOW:
```

```
                        ;OOOOOOOOOOOOFIRST ADDRESS TO BE RELOCATEDOOOOOOOO

                             CODE TO BE MOVED AND EXECUTED

                        ;OOOOOOOOOOOOLAST ADDRESS OF CODE TO BE RELOCATED OOOOOOOO

                        THE RELOC ROUTINE DOES NOT RELOCATE PROGRAM  CODE  INTO  THE
                        LAST 1000(8) BYTES OF MEMORY, THUS PRESERVING THE LOADERS.

                6.4     END
                        THIS ROUTINE IS ENTERED AT THE COMPLETION OF  EACH  PASS  IT
                        SETS UP (LOADS NEW PROCESSOR STATUS) FOR THE NEXT PASS;  AND
                        PRINTS THE PASS COUNT:

                             ICNT=XXXX

                7.0     MISCELLANEOUS

                7.1     EXECUTION TIME
                        THE  EXECUTION  TIME  IS  HIGHLY  VARIABLE  (DEPENDENT  ON
                        PROCESSOR,  TYPE OF MEMORY, AND AMOUNT OF MEMORY).  HOWEVER,
                        WHEN  THE  PROGRAM  IS  RUNNING  SUCCESSFULLY  THERE  IS  A
                        NOTICEABLE  'FLICKER' DISPLAYED IN THE CONSOLE LIGHT PATTERN
                        THE 'FLICKER' WILL DIM WHEN 'T' BIT TRAP PASSES  (EVERY  ODD
                        PASS)  ARE  RUNNING, THE PROGRAM SHOULD BE RUN FOR A MINIMUM
                        OF:
                                2 PASSES ICNT=2 11/05 OR 11/20
                        SOME TYPICAL TIMES FOLLOW:

                8.0     PROGRAM DESCRIPTION
                        THE PROGRAM  IS  DIVIDED  INTO  FOUR  SECTIONS  OF  POSITION
                        INDEPENDENT  RELOCATABLE  TEST  CODE.  EACH  SECTION  IS
                        APPROXIMATELY 1K WORDS LONG.  (EXCEPT SECTION A).

                        SECTION 0  THIS  SECTION  TEST  THE  UNARY  INSTRUCTION  SET
                                   EXECUTING  EACH UNARY INSTRUCTION IN EACH ADDRESS
                                   MODE (EXCLUDING UNARY INSTRUCTIONS USING  ADDRESS
                                   MODE 7).

                        SECTION 1  THIS SECTION TESTS THE UNARY  INSTRUCTIONS  USING
                                   ADDRESS  MODE  7 AND BINARYS IN ALL ADDRESS MODES
                                   (EXCLUDING BINARY BYTE OPS USING ADDRESS MODE 7).

                        SECTION 2  THIS SECTION TEST BINARY BYTE OPS  USING  ADDRESS
                                   MODE  7, JMP, JSR AND PROGRAM TRAP (IOT, TRAP AND
                                   EMT) INSTRUCTIONS.

                        SECTION A  FOLLOWING SECTION 2 IS  A  ROUTINE  TO  ASCERTAIN
                                   WHICH  CP THE PROGRAM IS RUNNING ON.  THE RESULTS
                                   ARE USED BY  THE  FOLLOWING  CODE  TO  CHECK  THE
                                   ADDITIONAL INSTRUCTIONS/FEATURES OF THE 11/40 AND
                                   11/45.
```

```
                        SECTION 3  THIS  SECTION  CHECKS  THAT  EACH  BIT  IN  THE
                                   PROCESSOR  STATUS  WORD (PSW) CAN BE SET CLEARED,
```

RESERVED INSTRUCTION, AND ODD ADDRESS TRAPS.

FOLLOWING SECTION 3 ARE TWO ROUTINES TO CHECK THE TELETYPE PRINTER LOGIC AND A ROUTINE TO START THE KW11-L LINE CLOCK. IF THE KW11-L IS AVAILABLE THE PRIORITY ARBITRATION LOGIC IS TESTED.

AFTER EACH INDIVIDUAL SECTION HAS BEEN EXECUTED THE "RELOC" ROUTINE WILL RELOCATE THE SECTION THROUGHOUT ALL MEMORY UP TO 28K. WHEN THE SECTION HAS BEEN RELOCATED AND EXECUTED IN ALL MEMORY THE "RELOC" ROUTINE WILL RETURN THE PROGRAM TO THE NEXT UNRELOCATED SECTION.

RELOCATION AND EXECUTION OF ALL SECTIONS THROUGHOUT ALL MEMORY CONSTITUTES A SINGLE PASS.

UPON COMPLETION OF A PASS OF THE PROGRAM THE PROGRAM RESTARTS USING A NEW PROCESSOR STATUS DEPENDING ON THE TYPE OF PROCESSOR AND THE PASS COUNT.

8.1 STACK POINTER
THE STACK POINTER IS SET AT 500.
NOTE: IF THE PROGRAM IS RUNNING IN EITHER USER OR SUPERVISOR MODE (NOT APPLICABLE IF 11/20 OR 11/05) THE USER/SUPERVISOR STACK POINTER IS SET TO 500 AND THE KERNEL STACK POINTER IS SET TO 600. THE KERNEL STACK POINTER IS USED ONLY FOR THE SCOPE, HLT, TTY, AND KW11-L (IF AVAILABLE TRAP/INTERUPT ROUTINES.

8.2 POWER FAILURE
A POWER FAIL SERVICE ROUTINE IS INCORPORATED IN THE TEST. WHEN USING THIS PROGRAM THE POWER SHOULD BE TURNED OFF WHEN RUNNING TO CHECK THE POWER FAIL LOGIC. WHEN THE POWER FAILS THE PROGRAM WILL TYPE:

POWER FAILED

AND RESTART THE PROGRAM AT THE BEGINNING. (START)

9.0 USER DEFINED RELOCATION LIMITS
THE PROGRAM WILL REQUEST A LOWER AND UPPER LIMIT FOR RELOCATION. THE LIMITS MUST BE BETWEEN 20000 AND 157776.
THE PROGRAM WILL EXECUTE IN THE LOWER 4K (0-17776) AND THE LIMITS SPECIFIED.
THE STARTING ADDRESS IS 204.
TO RETAIN PREVIOUSLY SPECIFIED LIMITS START AT 210.

!

```
                                .NLIST  SEQ,MD,MC
                                .LIST   ME
                                .ABS
                                .TITLE  FRONT END
                        ;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

                        ;GENERAL REGISTER ASSIGNMENTS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7
000000                          R10=%0
000001                          R11=%1
000002                          R12=%2
000003                          R13=%3
000004                          R14=%4
000005                          R15=%5


                        ;STATUS REGISTER (PSW) BIT ASSIGNMENTS
000001                          C=1                     ;C BIT
000002                          V=2                     ;V BIT
000004                          Z=4                     ;Z BIT
000010                          N=10                    ;N BIT
000020                          T=20                    ;'T' BIT
000340                          PRTY7=340               ;PRIORITY LEVEL 7
000300                          PRTY6=300               ;PRIORITY LEVEL 6
000200                          PRTY4=200               ;PRIORITY LEVEL 4

                        ;VECTOR ADDRESSES
000004                          ERRVEC=4                ;ADDRESS OF ERROR VECTOR
000010                          RESVEC=10               ;ADDRESS OF RESERVED INST. TRAP VECTOR
000014                          TBITVEC=14              ;ADDRESS OF 'T' BIT TRAP VECTOR
000014                          TRTVEC=14               ;ADDRESS OF 'TRACE' TRAP VECTOR
000014                          BPTVEC=14               ;ADDRESS OF 'BREAKPOINT' TRAP VECTOR
000020                          IOTVEC=20               ;ADDRESS OF IOT TRAP VECTOR
000024                          PFVEC=24                ;ADDRESS OF POWER FAIL TRAP VECTOR
000030                          EMTVEC=30               ;ADDRESS OF EMT VECTOR
000034                          TRAPVEC=34              ;ADDRESS OF TRAP VECTOR
000064                          TPVEC=64                ;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
000100                          LKVEC=100               ;ADDRESS KW11-L LINE CLOCK INT. VECTOR
000240                          PIRVEC=240              ;ADDRESS OF PIRQ VECTOR
000244                          FPEVEC=244              ;ADDRESS OF FLOATING POINT INT. VECTOR
000250                          MMVEC=250               ;ADDRESS OF MEM MGMT ERROR TRAP VECTOR

                        ;REGISTER ADDRESSES
177776                          PSW=    177776          ;ADDRESS OF STATUS REGISTER
177774                          SLR=    177774          ;ADDRESS OF STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;ADDRESS OF PROGRAM INTERRUPT REQUEST
177770                          UBREAK= 177770          ;ADDRESS OF MICRO BREAK REGISTER
177546                          LKS=    177546          ;ADDRESS OF KW11-L STATUS REG.
```

```
                177560                          TKS=    177560          ;ADDRESS OF KEYBOARD CSR
                177562                          TKB=    177562          ;ADDRESS OF KEYBOARD BUFFER
                177564                          TPS=    177564          ;ADDRESS OF TELEPRINTER CSR
                177566                          TPB=    177566          ;ADDRESS OF TELEPRINTER BUFFER
                177572                          SR0=    177572          ;ADDRESS OF MEM MGMT REGISTER SR0
                177570                          SWR=    177570          ;ADDRESS OF CONSOL SWITCH REGISTER
                177570                          DISPLAY=177570          ;ADDRESS OF CONSOL DISPLAY REGISTER
                177514                          LPS=    177514          ;ADDRESS OF LINE PRINTER STATUS REG
                177516                          LPB=    177516          ;ADDRESS OF LINE PRINTER DATA BUFFER

                                        ;INITIAL STACK POINTER SETTING
                000500                          STKPTR= 500             ;PROGRAM STACK PTR
                000600                          KPTR=600                ;KERNEL STACK PTR (USED BY KERNEL WHEN
                                                                        ;PROGRAM IS RUNNING IN OTHER THAN KERNEL
                                                                        ;MODE (NOT APPLICABLE TO 11/05,11/20)

                                        ;MISCELLANEOUS BIT ASSIGNMENTS
                100000                          BIT15=100000
                040000                          BIT14=40000
                020000                          BIT13=20000
                000400                          BIT8=400
                000100                          BIT6=100

                                        ;INSTRUCTION EQUATES
                104400                          HLT=TRAP                ;HLT IS A TRAP INST TO THE ERROR ROUTINE
                104000                          SCOPE=EMT               ;SCOPE IS AN EMT TRAP
```

```
                000046                          .=46
000046  016544                                  LOGICAL
                000052                          .=52
000052  040000                                  BIT14
                000200                          .=200
000200  012707  002066                          MOV     #START,PC       ;GO TO START OF TEST
000204  012707  002160                          MOV     #START1,PC      ;GO GET LOWER/UPPER RELOCATION BOUNDARY
000210  012707  002224                          MOV     #START3,PC      ;START WITH LAST TYPED BOUNDARY LIMITS

                                        ;ROUTINE TO SAVE REGISTERS ON THE STACK
                                        ;CALLED BY SAVE MACRO OR JSR    PC,$SAVR
000214  012667  000016          $SAVR:  MOV     (SP)+,1$                ;SAVE RETURN PC
000220  010546                          MOV     %5,-(SP)
000222  010446                          MOV     %4,-(SP)
```

```
000224  010346                          MOV     %3,-(SP)
000226  010246                          MOV     %2,-(SP)
000230  010146                          MOV     %1,-(SP)
000232  010046                          MOV     %0,-(SP)
000234  012707                          MOV     (PC)+,PC        ;RETURN
000236  000000                  1$:     0                       ;CONTAINS RETURN ADDRESS

                                ;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
                                ;CALLED BY RESTORE MACRO OR JSR  PC,$RESTR
000240  012667  000016          $RESTR: MOV     (SP)+,1$        ;SAVE RETURN PC
000244  012600                          MOV     (SP)+,%0
000246  012601                          MOV     (SP)+,%1
000250  012602                          MOV     (SP)+,%2
000252  012603                          MOV     (SP)+,%3
000254  012604                          MOV     (SP)+,%4
000256  012605                          MOV     (SP)+,%5
000260  012707                          MOV     (PC)+,PC        ;RETURN
000262  000000                  1$:     0                       ;CONTAINS RETURN ADDRESS

        000610                          .=610
                                ;POWER FAIL SUBROUTINE
000610  012737  000620  000024  PDWN:   MOV     #PUP,@#PFVEC
000616  000000                          HALT

                                ;POWER UP SUBROUTINE
000620  012737  000610  000024  PUP:    MOV     #PDWN,@#PFVEC   ;RESTORE POWER FAIL TRAP TO POWER
                                                                ;DOWN ROUINE ABOVE
000626  012706  000600                  MOV     #KPTR,SP        ;SET STACK PTR
000632  005027                          CLR     (PC)+
000634  000000                  1$:     .WORD   0               ;KILL TIME
000636  005267  177772          2$:     INC     1$
000642  001375                          BNE     2$
000644  004767  000362                  JSR     PC,.PRINT       ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
000650  000656                          PFAIL
000652  000137  002066                  JMP     @#START                 ;RESTART TEST

000656  005015  047520  042527  PFAIL:  .ASCIZ  <15><12>'POWER FAILED'<15><12>
000664  020122  040506  046111
000672  042105  005015     000

        000740                          .=740
                                ;NOTE: THIS CODE USED ONLY BY THE XOR TESTER.
                                ;TO USE CODE PLACE 776 (BR  .-2) IN SCOPEA
000740  012737  000002  000006  FORXOR: MOV     #RTI,@#ERRVEC+2 ;SET TIME OUT TRAP TO RETURN
000746  000261                          SEC                     ;SET C
000750  005737  177060                  TST     @#177060        ;IF A TIME OUT OCCURS THEN WHEN NEXT
                                                                ;INSTRUCTION IS EXECUTED 'C' WILL BE SET
                                                                ;AND IF NO TIME OUT 'C' WILL BE CLEARED
000754  103401                          BCS     1$              ;BRANCH IF 'C' SET (TIMED OUT)
000756  011601                          MOV     (SP),R1         ;ADDRESS OF NEXT SUBTEST TO R1
000760  005037  000006          1$:     CLR     @#ERRVEC+2      ;RESTORE TIME OUT TRAP
000764  010116                          MOV     R1,(SP)         ;GET RETURN ADDRESS BACK TO SUBTESTS
000766  000240                          NOP
000770  000002                          RTI                     ;RETURN EITHER TO LAST OR NEXT SUBTEST

        000776                          .=776
```

```
000776  000000                  TICKS:  .WORD   0               ;CONTAINS CLOCK TICK COUNT
        001000                          .=1000
001000  000000                  ICNT:   0                       ;CONTAINS PASS COUNT
001002  000000                  $FILLS: .WORD   0               ;CONTAINS FILLS COUNT IN ODD BYTE
                                                                ;AND FILLER CHARACTER IN EVEN BYTE
001004  000000                  FACTOR: 0                       ;CONTAINS RELOCATION FACTOR
                                ;SUBTRACT # IN FACTOR FROM PC TO GET PC OF ORIGINAL CODE
001006  000000                  RELR1:  0                       ;CONTAINS RELOCATED R1 (THE R1 OF THE
                                                                ;ORIGINAL CODE MOVED)
001010  000000                  FRSTAD: .WORD   0               ;CONTAINS FIRST ADRS OF CODE TO BE MOVED
001012  000000                  FRSTMEM:.WORD   0               ;CONTAINS LOWER RELOCATION BOUNDARY ADDRESS
001014  000751                          BR      FORXOR          ;BRANCH TO XOR TESTER CODE
                                ;SCOPE (EMT) SERVICE ROUTINE
                                ;THIS ROUTINE ALLOWS THE SUBTEST TO BE CONTINUOUSLY LOOPED, ITERATED
                                ;(OR NOT ITERATED) BEFORE BEGINNING NEXT SUBTEST
001016  000240                  SCOPEA: NOP
001020  032766  004000  000002          BIT     #4000,2(SP)     ;WAS REGISTER SET BIT SET ON TRAP
001026  001403                          BEQ     2$              ;BRANCH IF NOT
001030  052737  004000  177776          BIS     #4000,@#PSW     ;RETAIN REGISTER SET
001036  032737  040000  177570  2$:     BIT     #40000,@#SWR    ;CHECK BIT 14 (CONTINUOUS LOOP)
001044  001416                          BEQ     SCOPEC
001046  010116                  SCOPEB: MOV     R1,(SP)         ;LOAD RETURN ADDRESS
001050  010137  001006                  MOV     R1,@#RELR1
001054  163737  001004  001006          SUB     @#FACTOR,@#RELR1        ;RELR1 CONTAINS UNRELOCATED R1
001062  032737  000400  177570          BIT     #400,@#SWR      ;LOAD PDP11/45 MICRO BREAK REG?
001070  001403                          BEQ     1$
001072  113737  177570  177770          MOVB    @#SWR,@#UBREAK  ;LOAD MICRO BREAK REG WITH SR0-7
001100  000002                  1$:     RTI                     ;RETURN TO SUBTEST
001102  032737  004000  177570  SCOPEC: BIT     #4000,@#SWR     ;SUBTEST ITERATION DESIRED?
001110  001006                          BNE     SCOPEE          ;BRANCH IF NO ITERATION DESIRED?
001112  005327                          DEC     (PC)+           ;DECREMENT SUBTEST ITERATION COUNT
001114  000040                  SCOPED: 40                      ;CONTAINS SUBTEST ITERATION COUNT
001116  001353                          BNE     SCOPEB
001120  012767  000040  177766  SCOPEF: MOV     #40,SCOPED      ;RESET ITERATION COUNT
001126  011601                  SCOPEE: MOV     (SP),R1         ;GET ADDRESS OF NEXT TEST
001130  000746                          BR      SCOPEB

                                ;ROUTINE TO RELOCATE PROGRAM CODE
001132  032737  010000  177570  RELOC:  BIT     #10000,@#SWR    ;CHECK IF RELOCATION DESIRED (BIT12)
001140  001031                          BNE     3$              ;BRANCH IF NO RELOCATION DESIRED
001142  013700  001010                  MOV     @#FRSTAD,R0     ;GET FIRST ADDRESS OF CODE TO BE MOVED
001146  010005                          MOV     R0,R5           ;SAVE
001150  010204                          MOV     R2,R4           ;GET LAST ADDRESS OF CODE TO BE MOVED
001152  160504                          SUB     R5,R4           ;R4 CONTAINS # OF WORDS TO RELOCATE
001154  010203                          MOV     R2,R3           ;SAVE LAST ADDRESS OF CODE TO BE MOVED
001156  005737  001004                  TST     @#FACTOR        ;FIRST RELOCATION IS TO 20000
001162  001004                          BNE     10$
001164  010237  001230                  MOV     R2,@#RETPC      ;SAVE RETURN PC TO NEXT SECTION OF CODE
001170  013702  001012                  MOV     @#FRSTMEM,R2    ;SET FIRST ADDDRESS
001174  060204                  10$:    ADD     R2,R4           ;R4 CONTAINS LAST MEMORY ADDRESS
                                                                ;TO BE USED
001176  020437  002140                  CMP     R4,@#LSTMEM     ;CHECK IF SUFFICIENT MEMORY REMAINS
001202  101011                          BHI     4$
001204  012022                  1$:     MOV     (R0)+,(R2)+     ;RELOCATE PROGRAM CODE
001206  020003                          CMP     R0,R3           ;CHECK IF DONE
001210  001375                          BNE     1$
```

```
001212  024042                  2$:     CMP     -(R0),-(R2)     ;CHECK THAT CODE WAS RELOCATED
001214  001401                          BEQ     .+4             ;PROPERLY
001216  104400                          HLT                     ;ERROR! CODE NOT RELOCATED PROPERLY
001220  020005                          CMP     R0,R5           ;CHECK IF FINISHED CHECKING
001222  001373                          BNE     2$
001224  010207                  3$:     MOV     R2,PC           ;GO EXECUTE RELOCATED CODE
001226  011707                  4$:     MOV     (PC)+,PC        ;RETURN TO NEXT SECTION OF CODE
001230  000000                  RETPC:  0                       ;CONTAINS PC OF NEXT SECTION OF CODE

                                ;ROUTINE TO PRINT ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
001232  010046                  .PRINT: MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
001234  017600  000002                  MOV     @2(SP),R0       ;GET MESSAGE ADDRESS
001240  062766  000002  000002          ADD     #2,2(SP)        ;ADJUST RETURN PC

001246  112046                  1$:     MOVB    (R0)+,-(SP)     ;PUSH CHAR ON THE STACK
001250  001003                          BNE     2$              ;BRANCH IF NOT TERMINATOR
001252  005726                          TST     (SP)+           ;POP TERMINATOR OFF THE STACK
001254  012600                          MOV     (SP)+,R0        ;RESTORE R0
001256  000207                          RTS     PC              ;RETURN

001260  004767  000026          2$:     JSR     PC,5$           ;TYPE CHARACTER
001264  122726  000012          3$:     CMPB    #12,(SP)+       ;CHECK IF CHAR WAS A LINE FEED
001270  001366                          BNE     1$              ;BRANCH IF NOT LINE FEED

001272  016746  177504                  MOV     $FILLS,-(SP)    ;GET # OF FILLERS REQUIRED AFTER
                                                                ;LINE FEED AND FILLER CHARACTER
001276  105366  000001          4$:     DECB    1(SP)           ;DECREMENT FILLERS COUNT
001302  002770                          BLT     3$              ;BRANCH IF NO MORE FILLERS NEEDED
001304  004767  000002                  JSR     PC,5$           ;TYPE FILLER CHARACTER
001310  000772                          BR      4$

001312  105737  177564          5$:     TSTB    @#TPS           ;WAIT FOR OUTPUT DEVICE
001316  100375                          BPL     .-4             ;TO BECOME READY
001320  116637  000002  177566          MOVB    2(SP),@#TPB     ;TYPE CHARACTER
001326  000207                          RTS     PC

        000000                          NULL=0
                                ;ROUTINE TO PLACE ASCII VALUE OF AN ADDRESS IN TO ADDRESS MESSAGE
001330                          $FORM0:
001330  004767  176660                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001334  012704  001662                  MOV     #DIGITS,R4      ;ADDRESS WHERE ASCII VALUES ARE STORED
001340  005003                          CLR     R3              ;WORKING 3 INDEX REGISTER
001342  010201                          MOV     R2,R1           ;SAVE
001344  006302                  1$:     ASL     R2              ;FIRST DIGIT TO R3
001346  006103                          ROL     R3
001350  012700  000006                  MOV     #6,R0           ;DIGIT COUNT
001354  000404                          BR      3$              ;PRINT FIRST DIGIT
001356  006302                  2$:     ASL     R2
001360  006103                          ROL     R3
001362  005301                          DEC     R1
001364  001374                          BNE     2$
001366  012701  000003          3$:     MOV     #3,R1           ;DIGIT SHIFT COUNT
001372  116324  001652                  MOVB    DIGTAB(3),(4)+  ;LOAD DIGIT INTO MESSAGE
001376  005003                          CLR     R3              ;CLEAR INDEX
001400  005300                          DEC     R0              ;DEC DIGIT COUNT
001402  001365                          BNE     2$
```

```
001404  004767  176630                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001410  000207                          RTS     PC              ;RETURN

                                ;ERROR SERVICE CALLED BY TRAP (HLT) INSTRUCTION
001412  005737  177570          ERROR:  TST     @#SWR           ;HALT ON ERROR?
001416  100002                          BPL     .+6
001420  000000                          HALT                    ;ERROR PC IS THE TOP WORD
001422  000002                          RTI                     ;ON THE STACK
001424  032737  020000  177570          BIT     #20000,@#SWR    ;PRINT OUT DESIRED?
001432  001073                          BNE     1$              ;BRANCH IF NO PRINTOUT
001434  011627                          MOV     (SP),(PC)+      ;SAVE PC
001436  000000                  11$:    .WORD   0               ;CONTAINS SAVED PC
001440  016627  000002                  MOV     2(SP),(PC)+     ;GET STATUS ON TRAP
001444  000000                  12$:    .WORD   0               ;CONTAINS STATUS (PSW) AT TIME OF TRAP
001446  004767  176542                  JSR     PC,$SAVR        ;GO SAVE REGISTERS ON THE STACK
001452  013702  001000                  MOV     @#ICNT,R2       ;GET PASS COUNT
001456  004767  177646                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001462  016767  000176  000212          MOV     DIGITS+2,PASSES ;LOAD ASCII VALUES
001470  016767  000172  000206          MOV     DIGITS+4,PASSES+2
001476  004767  177530                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001502  001672                          PASCNT
001504  016702  177726                  MOV     11$,R2          ;GET PC OF ERROR CALL
001510  005742                          TST     -(R2)           ;DECREMENT PC TO HLT
001512  004767  177612                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001516  004767  177510                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001522  001707                          ERRPC
001524  004767  177502                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001530  001662                          DIGITS
001532  004767  177474                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001536  001714                          STATUS
001540  016702  177700                  MOV     12$,R2          ;GET STAUS AT TIME OF ERROR
001544  004767  177560                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001550  004767  177456                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001554  001662                          DIGITS
001556  016702  177654                  MOV     11$,R2          ;GET PC OF ERROR
001562  005742                          TST     -(R2)
001564  005737  001004                  TST     @#FACTOR
001570  001412                          BEQ     10$
001572  163702  001004                  SUB     @#FACTOR,R2     ;FORM PC OF ORIGINAL CODE
001576  004767  177526                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
001602  004767  177424                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001606  001721                          ERRPCO
001610  004767  177416                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001614  001662                          DIGITS
001616                          10$:
001616  004767  176416                  JSR     PC,$RESTR       ;RESTORE REGISTERS FROM STACK
001622  032737  002000  177570  1$:     BIT     #2000,@#SWR     ;RING BELL ON ERROR
001630  001403                          BEQ     2$
001632  004767  177374                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001636  001747                          BELL
001640  005737  177570          2$:     TST     @#SWR           ;HALT AFTER PRINT OUT
001644  100001                          BPL     .+4
001646  000000                          HALT
001650  000002                          RTI

                                ;DIGIT TABLE
```

```
001652  030460                  DIGTAB: "01
001654  031462                          "23
001656  032464                          "45
001660  033466                          "67
001662  030060  030060  030060  DIGITS: .ASCIZ  '000000 '
001670  000040
001672  005015                  PASCNT: .ASCII  <15><12>
001674  044440  047103  036524          .ASCII  '  ICNT='
001702  030060  030060     000  PASSES: .ASCIZ  '0000'
001707     040  041520  000075  ERRPC:  .ASCIZ  '  PC='
001714  051520  036527     000  STATUS: .ASCIZ  'PSW='
001721     120  020103  042522  ERRPCD: .ASCIZ  'PC RELOCATED FROM '
001726  047514  040503  042524
001734  020104  051106  046517
001743  000040
001744  005015     000          $CRLF:  .ASCIZ  <15><12>
001747     000     000          BELL:   .ASCIZ  <7>
        001752                          .EVEN

                                ;ROUTINE TO GET TYPED OCTAL ADDRESS AND CONVERT TO OCTAL. CALL:
                                ;       JSR     R5,RECD
                                ;       .WORD   0               ;CONVERTED DATA IS PLACED HERE
001752  010046                  RECD:   MOV     R0,-(SP)        ;SAVE R0 ON THE STACK
001754  005015                          CLR     (R5)            ;CLEAR OLD DATA
001756  105737  177560          1$:     TSTB    @#TKS           ;WAIT FOR USER TO TYPE CHARACTER
001762  100375                          BPL     1$
001764  113700  177562                  MOVB    @#TKB,R0        ;GET CHARACTER
001770  042700  000200                  BIC     #200,R0         ;STRIP MSB
001774  122700  000177                  CMPB    #177,R0         ;CHECK IF RUBOUT
002000  001010                          BNE     2$              ;BRANCH IF NOT RUBOUT
002002  112737  000134  177566          MOVB    #'\,@#TPB       ;TYPE \
002010  000241                          CLC                     ;CLEAR CARRY
002012  006015                          ROR     (R5)            ;SHIFT LAST TYPED CHARACTER
002014  006215                          ASR     (R5)            ;OUT OF DATA WORD
002016  006215                          ASR     (R5)
002020  000756                          BR      1$              ;GO WAIT FOR NEXT CHARACTER

002022  110037  177566          2$:     MOVB    R0,@#TPB        ;ECHO CHARACTER TYPED
002026  122700  000015                  CMPB    #15,R0          ;CHECK IF CARRIAGE RETURN
002032  001005                          BNE     3$              ;BRANCH IF NOT CARRIAGE RETURN
002034  004767  177172                  JSR     PC,.PRINT       ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
002040  001744                          $CRLF
002042  005725                          TST     (R5)+           ;STEP RETURN ADDRESS
002044  000205                          RTS     R5              ;RETURN

002046  042700  177770          3$:     BIC     #177770,R0      ;STRIP NON-ESSENTIAL BITS
002052  006315                          ASL     (R5)            ;SHIFT LAST CHARACTER 3 PLACES
002054  006315                          ASL     (R5)            ;LEFT
002056  006315                          ASL     (R5)
002060  050015                          BIS     R0,(R5)         ;AND INSERT NEW CHARACTER
002062  000735                          BR      1$              ;WAIT FOR NEXT CHARACTER


002064  000002                          RTI                             ;RETURN
```

```
                                                .TITLE  DZQKC-F BASIC 11 FAMILY INSTRUCTION EXER.

002066  005037  177776          START:  CLR     @#PSW           ;KERNEL MODE
002072  005000                          CLR     R0              ;CLEAR R0-R5
002074  005001                          CLR     R1
002076  005002                          CLR     R2
002100  005003                          CLR     R3
002102  005004                          CLR     R4
002104  005005                          CLR     R5
002106  012706  000600                  MOV     #KPTR,SP        ;SET KERNEL STACK PTR

                                        ;ROUTINE TO DETERMINE LAST MEMORY ADDRESS
002112  012737  002132  000004          MOV     #1$,@#ERRVEC
002120  005037  000006                  CLR     @#ERRVEC+2
002124  005000                          CLR     R0
002126  005720                          TST     (R0)+           ;WILL TIME OUTWHEN END OF MEMORY
002130  000776                          BR      .-2
002132  162700  000002          2$:     SUB     #2,R0
002136  010027                          MOV     R0,(PC)+        ;SET VALUE INTO LSTMEM
002140  000000                  LSTMEM: .WORD   0               ;CONTAINS VALUE OF LAST MEMORY ADDRESS
002142  162737  004000  002140          SUB     #4000,@#LSTMEM  ;SET PROTECTION FOR LOADERS
002150  012737  020000  001012          MOV     #20000,@#FRSTMEM;SET LOWER BOUNDARY AT 20000
002156  000422                          BR      START3          ;GO TO START 3
002160                          START1:
002160  004767  177046                  JSR     PC,.PRINT       ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
002164  016600                          MSG1
002166  004567  177560                  JSR     R5,RECD         ;GET LOWER LIMIT
002172  000000                  1$:     .WORD   0               ;CONTAINS TYPED LOWER LIMIT
002174  016737  177772  001012          MOV     1$,@#FRSTMEM    ;SET IN LOWER LIMIT
002202  004767  177024                  JSR     PC,.PRINT       ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
002206  016615                          MSG2
002210  004567  177536                  JSR     R5,RECD         ;GET UPPER LIMIT
002214  000000                  2$:     .WORD   0               ;CONTAINS UPPER LIMIT
002216  016737  177772  002140          MOV     2$,@#LSTMEM

002224  005037  001000          START3: CLR     @#ICNT          ;CLEAR PASS COUNT
002230  012737  000006  000004  START2: MOV     #ERRVEC+2,@#ERRVEC      ;SET ERROR TRAP TO HALT AT 6
002236  012706  000500                  MOV     #STKPTR,SP      ;SET STACK PTR
002242  013737  001000  177570          MOV     @#ICNT,@#DISPLAY;DISPLAY PASS COUNT
002250  012737  001016  000030          MOV     #SCOPEA,@#EMTVEC;SET EMT(SCOPE) TRAP VECTOR
002256  012737  001412  000034          MOV     #ERROR,@#TRAPVEC        ;SET TRAP (HLT) VECTOR
002264  012737  000340  000036          MOV     #340,@#TRAPVEC+2        ;PRIORITY LEVEL 4 ON TRAP

                                        ;000000000000 FIRST ADDRESS TO BE RELOCATED 000000000
002272  010700                  RELO:   MOV     PC,R0           ;GET PC
002274  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF RELO
002276  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
002302  010700                          MOV     PC,R0           ;GET CURRENT PC
002304  162700  002304                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
002310  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
002314  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR
                                        ;CHECK BRANCH INSTRUCTIONS
002316  000257                          CCC                     ;CC'S=0000
002320  103407                          BCS     CC0             ;SAME AS BLO
002322  102406                          BVS     CC0
002324  001405                          BEQ     CC0
```

```
002326  100404                          BMI     CC0
002330  002403                          BLT     CC0
002332  003402                          BLE     CC0
002334  101401                          BLOS    CC0
002336  101001                          BHI     .+4
002340  104400                  CC0:    HLT                     ;ONE OF THE ABOVE BRANCHES FAILED

                                ;CONTINUE
002342  000270                          SEN                     ;CC'S=1000
002344  100003                          BPL     CC1
002346  002002                          BGE     CC1
002350  003001                          BGT     CC1
002352  002401                          BLT     .+4
002354  104400                  CC1:    HLT                     ;ONE OF THE ABOVE BRANCHES FAILED

                                ;CONTINUE
002356  000262                          SEV                     ;CC'S=1010
002360  102003                          BVC     CC2
002362  002402                          BLT     CC2
002364  003401                          BLE     CC2
002366  002001                          BGE     .+4
002370  104400                  CC2:    HLT                     ;ERROR! ONE OF THE ABOVE BRANCHES FAILED

                                ;CONTINUE
002372  000261                          SEC                     ;CC'S=1011
002374  103002                          BCC     CC3
002376  101001                          BHI     CC3
002400  003001                          BGT     .+4
002402  104400                  CC3:    HLT                     ;ERROR! ONE OF THE ABOVE BRANCHES FAILED

                                ;CONTINUE
002404  000264                          SEZ                     ;CC'S=1111
002406  001003                          BNE     CC4
002410  003002                          BGT     CC4
002412  101001                          BHI     CC4
002414  003401                          BLE     .+4
002416  104400                  CC4:    HLT                     ;ERROR! ONE OF THE ABOVE BRANCHES FAILED
002420  104000                          SCOPE

                                ;TEST UNARY CONDITION CODES
                                ;CLR    R0
002422  000277                          SCC
002424  000244                          CLZ
002426  005000                          CLR     R0              ;R0=0,CC'S=0100
002430  103404                          BCS     CLR0
002432  102403                          BVS     CLR0
002434  001002                          BNE     CLR0
002436  100401                          BMI     CLR0
002440  003401                          BLE     .+4
002442  104400                  CLR0:   HLT                     ;ERROR! INCORRECT CC'S AFTER CLR

002444  000277                          SCC
002446  000244                          CLZ
002450  005700                          TST     R0              ;R0=0,CC'S=0100
002452  103404                          BCS     TST0
002454  102403                          BVS     TST0
```

```
002456  001002                  BNE     TSTO
002460  100401                  BMI     TSTO
002462  101401                  BLOS    .+4
002464  104400          TSTO:   HLT                     ;ERROR! INCORRECT CC'S AFTER TST

002466  000257                  CCC
002470  000266                  +SEZ!SEV
002472  005100                  COM     R0              ;R0=-1,CC'S=1001
002474  103004                  BCC     COMO
002476  102403                  BVS     COMO
002500  001402                  BEQ     COMO
002502  100001                  BPL     COMO
002504  002401                  BLT     .+4
002506  104400          COMO:   HLT                     ;ERROR! INCORRECT CC'S AFTER COM

002510  000261                  SEC
002512  005500                  ADC     R0              ;R0=000000,CC'S=0101
002514  103003                  BCC     ADCO
002516  102402                  BVS     ADCO
002520  001001                  BNE     ADCO
002522  002001                  BGE     .+4
002524  104400          ADCO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ADC

002526  000261                  SEC
002530  006000                  ROR     R0              ;R0=100000,CC'S=1010
002532  103404                  BCS     RORO
002534  102003                  BVC     RORO
002536  001402                  BEQ     RORO
002540  100001                  BPL     RORO
002542  003001                  BGT     .+4
002544  104400          RORO:   HLT                     ;ERROR! INCORRECT CC'S AFTER ROR
002546  000277                  SCC
002550  000242                  CLV
002552  005300                  DEC     R0                      ;R0=077777,CC'S=0011
002554  103004                  BCC     DECO
002556  102003                  BVC     DECO
002560  001402                  BEQ     DECO
002562  100401                  BMI     DECO
002564  003401                  BLE     .+4
002566  104400          DECO:   HLT                     ;ERROR! INCORRECT CC'S AFTER DEC

002570  000257                  CCC
002572  005200                  INC     R0              ;R0=100000,CC'S=1010
002574  103404                  BCS     INCO
002576  102003                  BVC     INCO
002600  001402                  BEQ     INCO
002602  100001                  BPL     INCO
002604  003001                  BGT     .+4
002606  104400          INCO:   HLT                     ;ERROR! INCORRECT CC'S AFTER INC

002610  000277                  SCC
002612  000242                  CLV
002614  005400                  NEG     R0              ;R0=100000,CC'S=1011
002616  103003                  BCC     NEGO
002620  102002                  BVC     NEGO
002622  001401                  BEQ     NEGO
```

```
002624  002001                          BGE     .+4
002626  104400          NEGO:           HLT                     ;ERROR! INCORRECT CC'S AFTER NEG

002630  000261                          SEC
002632  006300                          ASL     R0              ;R0=000000,CC'S=0111
002634  103004                          BCC     ASLO
002636  102003                          BVC     ASLO
002640  001002                          BNE     ASLO
002642  100401                          BMI     ASLO
002644  101401                          BLOS    .+4
002646  104400          ASLO:           HLT                     ;ERROR! INCORRECT CC'S AFTER ASL

002650  006100                          ROL     R0              ;R0=000001,CC'S=0000
002652  103402                          BCS     ROLO
002654  003401                          BLE     ROLO
002656  002001                          BGE     .+4
002660  104400          ROLO:           HLT                     ;ERROR! INCORRECT CC'S AFTER ROL

002662  006200                          ASR     R0              ;R0=000000,CC'S=0111
002664  103003                          BCC     ASRO
002666  102002                          BVC     ASRO
002670  001001                          BNE     ASRO
002672  002401                          BLT     .+4
002674  104400          ASRO:           HLT                     ;ERROR! INCORRECT CC'S AFTER ASR

002676  000277                          SCC
002700  005600                          SBC     R0              ;R0=-1,CC'S=1001
002702  103002                          BCC     SBCO
002704  102401                          BVS     SBCO
002706  003401                          BLE     .+4
002710  104400          SBCO:           HLT                     ;ERROR! INCORRECT CC'S AFTER SBC

002712  005400                          NEG     R0              ;R0=000001,CC'S=00001
002714  000300                          SWAB    R0              ;R0=000400 ,CC'S=0100
002716  103403                          BCS     SWABO
002720  102402                          BVS     SWABO
002722  001001                          BNE     SWABO
002724  002001                          BGE     .+4
002726  104400          SWABO:          HLT                     ;ERROR! INCORRECT CC'S AFTER SWAB
002730  104000                          SCOPE

                        ;CHECK REGISTER SELECTION
002732  005000                          CLR     R0
002734  000277                          SCC
002736  006100                          ROL     R0              ;R0=1
002740  010002                          MOV     R0,R2
002742  006302                          ASL     R2              ;R2=2
002744  010203                          MOV     R2,R3
002746  006303                          ASL     R3              ;R3=4
002750  010304                          MOV     R3,R4
002752  006304                          ASL     R4              ;R4=10
002754  010405                          MOV     R4,R5
002756  006305                          ASL     R5              ;R5=20
002760  010546                          MOV     R5,-(SP)        ;SET BITS SET IN REGISTERS
002762  050416                          BIS     R4,(SP)         ;INTO STACK ADDRESS
002764  050316                          BIS     R3,(SP)
```

```
002766  050216                          BIS     R2,(SP)
002770  050016                          BIS     R0,(SP)
002772  022726  000037                  CMP     #37,(SP)+
002776  001401                          BEQ     .+4                     ;WERE SET
003000  104400                          HLT                             ;MISSING BIT(S) REPRESENT
                                                                        ;INCORRECT REGISTER SELECTION

                                ;CHECK THAT ALL BITS CAN BE SET & CLEARED IN ALL REGISTERS
003002  000257                          CCC
003004  112700  000377                  MOVB    #377,R0         ;SET ALL BITS (MOVB EXTENDS SIGN)
003010  006100                  1$:     ROL     R0                      ;ROTATE A 0 THROUGH ALL BIT
003012  103776                          BCS     1$                      ;POSITIONS
003014  005200                          INC     R0                      ;FINAL RESULT IS -1
003016  001401                          BEQ     .+4
003020  104400                          HLT                             ;ERROR!

003022  012700  000020                  MOV     #16.,R0         ;SET SHIFT COUNT
003026  005002                          CLR     R2
003030  000261                  2$:     SEC
003032  006002                          ROR     R2              ;ROTATE 1 THROUGH ALL BIT POSITS
003034  005300                          DEC     R0              ;DECREMENT SHIFT COUNT
003036  001374                          BNE     2$
003040  005102                          COM     R2              ;R2 SHOULD CONTAIN -1
003042  001401                          BEQ     .+4
003044  104400                          HLT                     ;ERROR! CHECK R2 SHOULD = 0

003046  012703  100000                  MOV     #100000,R3
003052  006203                  3$:     ASR     R3              ;EXTEND 1 BIT THROUGH ALL POSITIONS
003054  103376                          BCC     3$
003056  005203                          INC     R3
003060  001401                          BEQ     .+4
003062  104400                          HLT                             ;ERROR!

003064  112704  177401                  MOVB    #177401,R4      ;R4=!
003070  060404                  4$:     ADD     R4,R4           ;HAS THE AFFECT OF SHIFTING A BIT
003072  103376                          BCC     4$              ;THROUGH ALL POSITIONS
003074  005704                          TST     R4              ;RESULT SHOULD BE 0
003076  001401                          BEQ     .+4
003100  104400                          HLT

003102  012705  000001                  MOV     #1,R5
003106  006305                  5$:     ASL     R5
003110  102376                          BVC     5$
003112  006305                          ASL     R5
003114  103002                          BCC     6$
003116  005705                          TST     R5
003120  001401                          BEQ     .+4
003122  104400                  6$:     HLT

                                ;CHECK REGISTER VOLITILITY
003124  005002                          CLR     R2
003126  005102                          COM     R2              ;R2=-1
003130  010203                          MOV     R2,R3
003132  000257                          CCC
003134  006002                          ROR     R2              ;R2=LOOP COUNT
003136  006202                          ASR     R2
```

```
003140  010304                  7$:     MOV     R3,R4
003142  005302                          DEC     R2              ;DECREMENT LOOP COUNT
003144  001375                          BNE     7$
003146  005203                          INC     R3              ;CHECK R3
003150  001002                          BNE     9$
003152  005204                          INC     R4              ;CHECK R4
003154  001401                          BEQ     .+4
003156  104400                  8$:     HLT

                                ;CHECK TRANSFER OF REGISTER DATA BETWEEN THE GS AND GD REGISTERS (11/45)
003160  032737  000020  177776  GSTST:  BIT     #20,@#PSW       ;CHECK IF 'T' BIT IS SET
003166  001052                          BNE     7$              ;SKIP TEST IF 'T' BIT SET
003170  010146                          MOV     R1,-(SP)        ;SAVE SCOPE PTR
003172  010627                          MOV     SP,(PC)+        ;SAVE STACK PTR
003174  000000                  1$:     .WORD   0               ;CONTAINS SAVED STACK PTR
003176  010727                          MOV     PC,(PC)+        ;LOAD DATA. THE CURRENT PC IS USED AS
003200  000000                  2$:     .WORD   0               ;DATA. IF THIS TEST FAILS 2$ CON-
                                                                ;TAINS THE DATA BEING USED.
003202  005267  177772                  INC     2$              ;MAKE ODD TO CHECK BIT 0
003206  016700  177766          3$:     MOV     2$,R0           ;LOAD GD REGISTER 0
003212  010001                          MOV     R0,R1           ;TRANSFER GS REG 0 TO GD REG 1
003214  010102                          MOV     R1,R2           ;AND GS REG 1 TO GD REG 2
003216  010203                          MOV     R2,R3           ;ETC...
003220  010304                          MOV     R3,R4
003222  010405                          MOV     R4,R5
003224  152737  000340  177776          BISB    #340,@#PSW      ;SET PRIORITY LEVEL 7
003232  010506                          MOV     R5,SP           ;TRANSFER GS REG 5 TO GD STK PTR
003234  010627                          MOV     SP,(PC)+        ;TRANSFER GS STK PTR TO MEMORY
003236  000000                  4$:     .WORD   0               ;CONTAINS GS STACK PTR
003240  016706  177730                  MOV     1$,SP           ;RESTORE STK PTR NEEDED FOR HLT/SCOPE
003244  142737  000340  177776          BICB    #340,@#PSW      ;SET PRIORITY LEVEL 0
003252  026700  177760                  CMP     4$,R0           ;COMPARE GS/GD STKPTR WITH GS REG 0
003256  001004                          BNE     5$              ;BRANCH IF THEY WERE NOT =
003260  006367  177714                  ASL     2$              ;SHIFT TEST DATA UNTIL = 000000
003264  001350                          BNE     3$
003266  000411                          BR      6$
003270  010046                  5$:     MOV     R0,-(SP)        ;GET GS REG 0
003272  010146                          MOV     R1,-(SP)        ;ETC...
003274  010246                          MOV     R2,-(SP)
003276  010346                          MOV     R3,-(SP)
003300  010446                          MOV     R4,-(SP)
003302  010546                          MOV     R5,-(SP)
003304  104400                          HLT                     ;ERROR! DATA IN GS STK PTR NOT = GS REG 0
                                                                ;GS REG 0-GS REG 5 ARE ON THE STACK
003306  016706  177662                  MOV     1$,SP           ;RESTORE STACK PTR
003312  012601                  6$:     MOV     (SP)+,R1        ;RESTORE SCOPE PTR
003314  104000                  7$:     SCOPE

                                ;TEST UNARY WORD INSTRUCTIONS USING ADDRESS MODE 1
003316  000401                          BR      .+4
003320  000000                          .WORD   0               ;RESERVE ADDRESS FOR TESTS
003322  010702                          MOV     PC,R2
003324  162702  000004                  SUB     #4,R2           ;R2 POINTS TO RESERVED WORD
003330  005012                          CLR     (R2)            ;PRESET (R2)

003332  000261                          SEC
```

```
003334  006012                  ROR     (R2)            ;(R2)=100000,CC=1010
003336  101402                  BLOS    ROR1
003340  100001                  BPL     ROR1
003342  002001                  BGE     .+4
003344  104400          ROR1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003346  000257                  CCC
003350  000261                  SEC
003352  005312                  DEC     (R2)            ;(R2)=077777,CC=0011
003354  103001                  BCC     DEC1
003356  003401                  BLE     .+4
003360  104400          DEC1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003362  000257                  CCC
003364  000261                  SEC
003366  005512                  ADC     (R2)            ;(R2)=100000,CC=1010
003370  103403                  BCS     ADC1
003372  102002                  BVC     ADC1
003374  100001                  BPL     ADC1
003376  001001                  BNE     .+4
003400  104400          ADC1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003402  006112                  ROL     (R2)            ;(R2)=000000,CC=0111
003404  103003                  BCC     ROL1
003406  102002                  BVC     ROL1
003410  001001                  BNE     ROL1
003412  100001                  BPL     .+4
003414  104400          ROL1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003416  006112                  ROL     (R2)            ;(R2)=000001,CC=0000
003420  101402                  BLOS    ROL1A           ;BRANCH IF C OR Z IS SET
003422  102401                  BVS     ROL1A
003424  100001                  BPL     .+4
003426  104400          ROL1A:  HLT

003430  006212                  ASR     (R2)            ;(R2)=000000,CC=0111
003432  103003                  BCC     ASR1
003434  102002                  BVC     ASR1
003436  001001                  BNE     ASR1
003440  100001                  BPL     .+4
003442  104400          ASR1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003444  006012                  ROR     (R2)            ;(R2)=100000,CC=1010
003446  103403                  BCS     ROR1A
003450  102002                  BVC     ROR1A
003452  001401                  BEQ     ROR1A
003454  100401                  BMI     .+4
003456  104400          ROR1A:  HLT

003460  000261                  SEC
003462  005212                  INC     (R2)            ;(R2)=100001,CC=1001
003464  103003                  BCC     INC1
003466  102402                  BVS     INC1
003470  001401                  BEQ     INC1
003472  100401                  BMI     .+4
003474  104400          INC1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
```

```
003476  005612                  SBC     (R2)            ;(R2)=100000,CC=1000
003500  103403                  BCS     SBC1
003502  102402                  BVS     SBC1
003504  001401                  BEQ     SBC1
003506  100401                  BMI     .+4
003510  104400          SBC1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003512  000261                  SEC
003514  005612                  SBC     (R2)            ;(R2)=077777,CC=0010
003516  103403                  BCS     SBC1A
003520  102002                  BVC     SBC1A
003522  001401                  BEQ     SBC1A
003524  100001                  BPL     .+4
003526  104400          SBC1A:  HLT                     ;ERROR! INCORRECT CC'S AS SHOEN ABOVE

003530  000261                  SEC
003532  005512                  ADC     (R2)            ;(R2)=100000,CC=1010
003534  100401                  BMI     .+4
003536  104400                  HLT

003540  000261                  SEC
003542  006312                  ASL     (R2)            ;(R2)=000000,CC=0111
003544  103003                  BCC     ASL1
003546  102002                  BVC     ASL1
003550  001001                  BNE     ASL1
003552  100001                  BPL     .+4
003554  104400          ASL1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003556  005112                  COM     (R2)            ;(R2)=177777,CC=1001
003560  103002                  BCC     COM1
003562  102401                  BVS     COM1
003564  100401                  BMI     .+4
003566  104400          COM1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003570  000250                  CLN
003572  005712                  TST     (R2)            ;(R2)=177777,CC=1000
003574  103403                  BCS     TST1
003576  102402                  BVS     TST1
003600  100001                  BPL     TST1
003602  001001                  BNE     .+4
003604  104400          TST1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003606  000262                  SEV
003610  005412                  NEG     (R2)            ;(R2)=000001,CC=0000
003612  103002                  BCC     NEG1
003614  102401                  BVS     NEG1
003616  001001                  BNE     .+4
003620  104400          NEG1:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003622  005312                  DEC     (R2)            ;(R2)=000000,CC=0101
003624  103001                  BCC     DEC1A
003626  001401                  BEQ     .+4
003630  104400          DEC1A:  HLT                     :ERROR! INCORRECT CC'S AS SHOWN ABOVE
003632  104000                  SCOPE
```

```
                                ;CHECK UNARY BYTE INSTRUCTIONS USING ADDRESS MODE 1
003634  000401                  BR      .+4             ;RESERVE A WORD
003636  000000                  .WORD   0               ;ADDRESS RESERVED FOR TESTS
003640  010703                  MOV     PC,R3
003642  162703  000004          SUB     #4,R3           ;R3 POINTS TO EVEN BYTE OF WORD
003646  010304                  MOV     R3,R4           ;R4 POINTS TO ODD BYTE OF WORD
003650  005204                  INC     R4
003652  005013                  CLR     (R3)            ;PRESET DATA

003654  000261          1$:     SEC
003656  105513                  ADCB    (R3)            ;ADD CARRY TO EVEN BYTE
003660  100402                  BMI     2$              ;UNTIL EVEN BYTE BECOMES NEGATIVE
003662  105214                  INCB    (R4)            ;INCREMENT ODD BYTE
003664  000773                  BR      1$
003666  102401          2$:     BVS     .+4             ;(R3)=077600=[0774][200],CC=1010
003670  104400                  HLT
003672  000242                  CLV
003674  105214                  INCB    (R4)            ;(R3)=100200=[1000][200],CC=1010
003676  103402                  BCS     INCB1
003700  102001                  BVC     INCB1
003702  100401                  BMI     .+4
003704  104400          INCB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003706  106114                  ROLB    (R4)            ;(R3)=000200=[0000][200],CC=0111
003710  103002                  BCC     ROLB1
003712  102001                  BVC     ROLB1
003714  001401                  BEQ     .+4
003716  104400          ROLB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003720  105614                  SBCB    (R4)            ;(R3)=177600=[1774][200], CC=1001
003722  103002                  BCC     SBCB1
003724  102401                  BVS     SBCB1
003726  100401                  BMI     .+4
003730  104400          SBCB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003732  106313                  ASLB    (R3)            ;(R3)=177400,CC=0111
003734  103002                  BCC     ASLB1
003736  102001                  BVC     ASLB1
003740  001401                  BEQ     .+4
003742  104400          ASLB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003744  105413                  NEGB    (R3)            ;(R3)=177400,CC=0100
003746  103402                  BCS     NEGB1
003750  102401                  BVS     NEGB1
003752  001401                  BEQ     .+4
003754  104400          NEGB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003756  000277                  SCC
003760  105313                  DECB    (R3)            ;(R3)=177777,CC=1001
003762  103002                  BCC     DECB1
003764  102401                  BVS     DECB1
003766  001001                  BNE     .+4
003770  104400          DECB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

003772  000241                  CLC
003774  106013                  RORB    (R3)            ;(R3)=177577,CC=0011
```

```
003776  103002                  BCC     RORB1
004000  102001                  BVC     RORB1
004002  100001                  BPL     .+4
004004  104400          RORB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004006  000241                  CLC
004010  105114                  COMB    (R4)            ;(R3)=000177,CC=0101
004012  103002                  BCC     COMB1
004014  102401                  BVS     COMB1
004016  001401                  BEQ     .+4
004020  104400          COMB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004022  106213          1$:     ASRB    (R3)            ;SHIFT EVEN BYTE UNTIL V CLEARS
004024  102002                  BVC     2$
004026  105514                  ADCB    (R4)            ;AND ADD CARRY TO ODD BYTE
004030  000774                  BR      1$
004032  103401          2$:     BCS     ASRB1
004034  001401                  BEQ     .+4
004036  104400          ASRB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004040  106214                  ASRB    (R4)
004042  106214                  ASRB    (R4)            ;(R3)=000400,CC=0011
004044  103002                  BCC     ASRB1A
004046  102001                  BVC     ASRB1A
004050  001001                  BNE     .+4
004052  104400          ASRB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004054  105314                  DECB    (R4)            ;(R3)=000000,CC=0100
004056  001401                  BEQ     .+4
004060  104400                  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004062  000261                  SEC
004064  106014                  RORB    (R4)            ;(R3)=100000,CC=1010
004066  103402                  BCS     RORB1A
004070  102001                  BVC     RORB1A
004072  100401                  BMI     .+4
004074  104400          RORB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004076  000242                  CLV
004100  105314                  DECB    (R4)            ;(R3)=077400,CC=0100
004102  102401                  BVS     .+4
004104  104400                  HLT

004106  000261                  SEC
004110  105313                  DECB    (R3)            ;(R3)=077777,CC=1001
004112  103002                  BCC     DECB1A
004114  102401                  BVS     DECB1A
004116  100401                  BMI     .+4
004120  104400          DECB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004122  000277                  SCC
004124  000313                  SWAB    (R3)            ;(R3)=177577=[1774][177],CC=0000
004126  103402                  BCS     SWAB1
004130  102401                  BVS     SWAB1
004132  100001                  BPL     .+4
004134  104400          SWAB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
```

```
004136  105714                  TSTB    (R4)            ;(R3)=177777=[1774][1771],CC=1000
004140  103402                  BCS     TSTB1
004142  102401                  BVS     TSTB1
004144  100401                  BMI     .+4
004146  104400          TSTB1:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004150  105014                  CLRB    (R4)            ;(R3)=000177=[0000][1771],CC=0100
004152  001401                  BEQ     .+4
004154  104400                  HLT
004156  106313                  ASLB    (R3)            ;(R3)=000376 ,CC=1010
004160  103402                  BCS     ASLB1A
004162  102001                  BVC     ASLB1A
004164  100401                  BMI     .+4
004166  104400          ASLB1A: HLT                     ;ERROR! INCORRECT CC S AS SHOWN ABOVE

004170  105113                  COMB    (R3)            ;(R3)=000001,CC=0001
004172  103002                  BCC     COMB1A
004174  102401                  BVS     COMB1A
004176  100001                  BPL     .+4
004200  104400          COMB1A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004202  000313                  SWAB    (R3)            ;(R3)=000400, CC=0100
004204  001401                  BEQ     .+4
004206  104400                  HLT

004210  105213                  INCB    (R3)
004212  000261                  SEC
004214  105613                  SBCB    (R3)            ;(R3)=000400,CC=0100
004216  001401                  BEQ     .+4
004220  104400                  HLT
004222  027313  000400          CMP     #400,(R3)       ;CHECK REMAINING RESULT
004226  001401                  BEQ     .+4
004230  104400                  HLT
004232  104000                  SCOPE

                        ;CHECK UNARY WORD OPS USING ADDRESS MODES 2 AND 4 (AUTO INC/DEC)
004234  000401                  BR      .+4
004236  000000                  .WORD   0               ;ADDRESS RESERVED FOR TESTS
004240  010704                  MOV     PC,R4
004242  162704  000004          SUB     #4,R4           ;R4 AND R5 POINT TO
004246  010405                  MOV     R4,R5           ;RESERVED WORD
004250  005015                  CLR     (R5)            ;PRESET DATA=0

004252  000277                  SCC
004254  000244                  CLZ
004256  005725                  TST     (R5)+           ;(R5)=000000,CC=0100
004260  103402                  BCS     TST2
004262  102401                  BVS     TST2
004264  001401                  BEQ     .+4
004266  104400          TST2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004270  005145                  COM     -(R5)           ;(R5)=177777,CC=1001
004272  103001                  BCC     COM4
004274  100401                  BMI     .+4
004275  104400          COM4:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
```

```
004300  000241                  CLC
004302  006024                  ROR     (R4)+           ;(R4)=077777,CC=0011
004304  103002                  BCC     ROR2
004306  102001                  BVC     ROR2
004310  100001                  BPL     .+4
004312  104400          ROR2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004314  000257                  CCC
004316  005244                  INC     -(R4)           ;(R4)=100000,CC=1010
004320  102002                  BVC     INC4
004322  001401                  BEQ     INC4
004324  100401                  BMI     .+4
004326  104400          INC4.   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004330  000261                  SEC
004332  000324                  SWAB    (R4)+           ;(R4)=000200,CC=1000
004334  103401                  BCS     SWAB2
004336  100401                  BMI     .+4
004340  104400          SWAB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004342  005425                  NEG     (R5)+           ;(R5)=177600,CC=1001
004344  103001                  BCC     NEG2
004346  100401                  BMI     .+4
004350  104400          NEG2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004352  005C44                  CLR     -(R4)           ;(R4)=000000,CC=0100
004354  001401                  BEQ     .+4
004356  104400                  HLT

004360  000261                  SEC
004362  006045                  ROR     -(R5)           ;(R5)=100000,CC=1010
004364  000261                  SEC
004366  005525                  ADC     (R5)+           ;(R5)=100001,CC=1000
004370  102401                  BVS     ADC2
004372  100401                  BMI     .+4
004374  104400          ADC2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004376  000262                  SEV
004400  006224                  ASR     (R4)+           ;(R4)=140000,CC=1001
004402  103002                  BCC     ASR2
004404  102401                  BVS     ASR2
004406  100401                  BMI     .+4
004410  104400          ASR2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004412  000262                  SEV
004414  006144                  ROL     -(R4)           ;(R4)=100001, CC=1001
004416  103002                  BCC     ROL4
004420  102401                  BVS     ROL4
004422  100401                  BMI     .+4
004424  104400          ROL4:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004426  005645                  SBC     -(R5)           ;(R5)=100000,CC=1000
004430  103001                  BCC     .+4
004432  104400                  HLT                     ;ERROR! 'C' BIT FAILED TO CLEAR
```

```
004434  005325                          DEC     (R5)+           ;(R5)=077777,CC=0010
004436  103402                          BCS     DEC2
004440  102001                          BVC     DEC2
004442  100001                          BPL     .+4
004444  104400                  DEC2:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004446  006324                          ASL     (R4)+           ;(R4)=177776,CC=1010
004450  102401                          BVS     .+4
004452  104400                          HLT
004454  006344                          ASL     -(R4)           ;(R4)=177774,CC=1001
004456  103003                          BCC     ASL4
004460  102402                          BVS     ASL4
004462  001401                          BEQ     ASL4
004464  100401                          BMI     .+4
004466  104400                  ASL4:   HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004470  022724  177774                  CMP     #177774,(R4)+
004474  001401                          BEQ     .+4
004476  104400                          HLT
004500  020405                          CMP     R4,R5
004502  001401                          BEQ     .+4
004504  104400                          HLT
004506  104000                          SCOPE

                                ;CHECK UNARY BYTE OPS USING ADDRESS MODES 2 AND 4
004510  000401                          BR      .+4             ;RESERVE A WORD
004512  000000                          .WORD   0               ;RESERVED WORD
004514  010705                          MOV     PC,R5
004516  162705  000004                  SUB     #4,R5           ;R5 POINTS TO EVEN BYTE OF RESERVED WORD
004522  010500                          MOV     R5,R0
004524  010002                          MOV     R0,R2
004526  005202                          INC     R2              ;R2 POINTS TO ODD BYTE OF RESERVED WORD
004530  005010                          CLR     (R0)            ;PRESET

004532  000277                          SCC
004534  000241                          CLC
004536  105125                          COMB    (R5)+           ;(R0)=000377,CC=1001
004540  103002                          BCC     COMB2
004542  102401                          BVS     COMB2
004544  100401                          BMI     .+4
004546  104400                  COMB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004550  105542                          ADCB    -(R2)           ;(R0)=000000,CC=0101
004552  001401                          BEQ     .+4
004554  104400                          HLT                     ;ERROR! INCORRECT RESULT AS SHOWN ABOVE
004556  105525                          ADCB    (R5)+           ;(R0)=000400,CC=0000
004560  103401                          BCS     ADCB2
004562  001001                          BNE     .+4
004564  104400                  ADCB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004566  000261                          +SEC!SEV
004570  106045                          RORB    -(R5)           ;(R0)=100000,CC=1001
004572  103003                          BCC     RORB4
004574  102402                          BVS     RORB4
004576  001401                          BEQ     RORB4
004600  100401                          BMI     .+4
```

```
004602  104400                  RORB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004604  000277                          SCC
004606  106122                          ROLB    (R2)+           ;(R0)=100001,CC=0000
004610  103403                          BCS     ROLB2
004612  102402                          BVS     ROLB2
004614  001401                          BEQ     ROLB2
004616  100001                          BPL     .+4
004620  104400                  ROLB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004622  000257                          CCC
004624  106225                          ASRB    (R5)+           ;(R0)=140001, CC=1010
004626  103402                          BCS     ASRB2
004630  102001                          BVC     ASRB2
004632  100401                          BMI     .+4
004634  104400                  ASRB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABO.E

004636  105242                          INCB    -(R2)           ;(R0)=140002,CC=0000
004640  000277                          SCC
004642  106222                          ASRB    (R2)+           ;(R0)=140001,CC=0000
004644  103402                          BCS     ASRB2A
004646  102401                          BVS     ASRB2A
004650  100001                          BPL     .+4
004652  104400                  ASRB2A: HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004654  000266                          +SEZ!SEV                ;SET Z,V
004656  106345                          ASLB    -(R5)           ;(R0)=100001,CC=1C01
004660  103003                          BCC     ASLB4
004662  102402                          BVS     ASLB4
004664  001401                          BEQ     ASLB4
004666  100401                          BMI     .+4
004670  104400                  ASLB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004672  105322                          DECB    (R2)+           ;(R0)=077401=[0774][001] ,CC=0010
004674  103002                          BCC     DECB2
004676  102001                          BVC     DECB2
004700  100001                          BPL     .+4
004702  104400                  DECB2:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWNABOVE

004704  105645                          SBCB    -(R5)           ;(R0)=077400, CC=0100
004706  103402                          BCS     SBCB4
004710  102401                          BVS     SBCB4
004712  001401                          BEQ     .+4
004714  104400                  SBCB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004716  105442                          NEGB    -(R2)           ;(R0)=10400,CC=1001
004720  103002                          BCC     NEGB4
004722  102401                          BVS     NEGB4
004724  100401                          BMI     .+4
004726  104400                  NEGB4:  HLT                     ;ERROR! INCORRECT CC'S AS SHOWN ABOVE

004730  105725                          TSTB    (R5)+           ;(R0)=100400,CC=0100
004732  103401                          BCS     TSTB2
004734  001401                          BEQ     .+4
004736  104400                  TSTB2:  HLT
```

```
004740  105722                          TSTB    (R2)+           ;(R0)=100400,CC=1000
004742  001401                          BEQ     TST82A
004744  100401                          BMI     .+4
004746  104400              TST82A:     HLT

004750  000261                          SEC
004752  000342                          SWAB    -(R2)           ;(R0)=000201,CC=1000
004754  103401                          BCS     SWAB4
004756  100401                          BMI     .+4
004760  104400              SWAB4:      HLT

004762  000277                          SCC
004764  105225                          INCB    (R5)+           ;(R0)=000601=[0004][201],CC=0000
004766  103003                          BCC     INCB2
004770  102402                          BVS     INCB2
004772  001401                          BEQ     INCB2
004774  100001                          BPL     .+4
004776  104400              INCB2:      HLT

005000  022227  000601                  CMP     (R2)+,#000601   ;CHECK END RESULT
005004  001401                          BEQ     .+4
005006  104400                          HLT
005010  020205                          CMP     R2,R5           ;CHECK REGISTERS
005012  001401                          BEQ     .+4
005014  104400                          HLT
005016  104000                          SCOPE

                            ;CHECK UNARY WORD OPS USING ADDRESS MODES 3 AND 5
005020  000402                          BR      .+6             ;RESERVE 2 WORDS
005022  000000                          .WORD   0               ;1 FOR THE ADDRESS
005024  000000                          .WORD   0               ;AND 1 FOR DATA
005026  010703                          MOV     PC,R3
005030  162703  000004                  SUB     #4,R3
005034  005013                          CLR     (R3)            ;PRESET DATA
005036  010300                          MOV     R3,R0           ;R0 POINTS TO DATA WORD
005040  005743                          TST     -(R3)
005042  010013                          MOV     R0,(R3)
005044  010304                          MOV     R3,R4

005046  000257                          CCC
005050  005733                          TST     @(R3)+          ;(R0)=000000,CC=0100
005052  001401                          BEQ     .+4
005054  104400                          HLT

005056  000261                          SEC
005060  006053                          ROR     @-(R3)          ;(R0)=100000,CC=1010
005062  103402                          BCS     ROR5
005064  102001                          BVC     ROR5
005066  100401                          BMI     .+4
005070  104400              ROR5:       HLT

005072  000257                          CCC
005074  006234                          ASR     @(R4)+          ;(R0)=140000,CC=1010
005076  102001                          BVC     ASR3
005100  100401                          BMI     .+4
005102  104400              ASR3:       HLT
```

```
005104  000250                  CLN
005106  006333                  ASL     @(R3)+          ;(R0)=100000,CC=1001
005110  103002                  BCC     ASL3
005112  102401                  BVS     ASL3
005114  100401                  BMI     .+4
005116  104400          ASL3:   HLT

005120  000277                  SCC
005122  005354                  DEC     @-(R4)          ;(R0)=077777, CC=0010
005124  103003                  BCC     DEC5
005126  102002                  BVC     DEC5
005130  001401                  BEQ     DEC5
005132  100001                  BPL     .+4
005134  104400          DEC5:   HLT

005136  005453                  NEG     @-(R3)          ;(R0)=100001, CC=1001
005140  103002                  BCC     NEG5
005142  102401                  BVS     NEG5
005144  100401                  BMI     .+4
005146  104400          NEG5:   HLT

005150  000262                  SEV
005152  005134                  COM     @(R4)+          ;(R0)=077776, CC=0001
005154  103001                  BCC     COM3
005156  102001                  BVC     .+4
005160  104400          COM3:   HLT

005162  005233                  INC     @(R3)+          ;(R0)=077777, CC=0001
005164  103001                  BCC     INC3
005166  100001                  BPL     .+4
005170  104400          INC3:   HLT

005172  005554                  ADC     @-(R4)          ;(R0)=100000, CC=1010
005174  103402                  BCS     ADC5
005176  102001                  BVC     ADC5
005200  100401                  BMI     .+4
005202  104400          ADC5:   HLT

005204  000257                  CCC
005206  006134                  ROL     @(R4)+          ;(R0)=000000,CC=0111
005210  103002                  BCC     ROL3
005212  102001                  BVC     ROL3
005214  001401                  BEQ     .+4
005216  104400          ROL3:   HLT

005220  005253                  INC     @-(R3)          ;(R0)=100001, CC=0001
005222  005654                  SBC     @-(R4)          ;(R0)=000000, CC=0100
005224  103401                  BCS     SBC5
005226  001401                  BEQ     .+4
005230  104400          SBC5:   HLT
005232  104000                  SCOPE

                        ;CHECK UNARY BYTE OPS USING ADDRESS MODES 3 AND 5
005234  000403                  BR      .+10            ;RESERVE 3 WORDS
005236  000000                  .WORD   0               ;1 FOR EVEN BYTE ADDRESS
```

```
005240  000000                  .WORD   0               ;1 FOR ODD BYTE ADDRESS
005242  000000                  .WORD   0               ;AND 1 FOR DATA
005244  010702                  MOV     PC,R2
005246  005742                  TST     -(R2)           ;BACK R2 UP TO
005250  005742                  TST     -(R2)           ;DATA WORD
005252  010200                  MOV     R2,R0           ;R0 POINTS TO THE DATA WORD
005254  005010                  CLR     (R0)            ;PRESET DATA
005256  005742                  TST     -(R2)           ;BACK R2 UP TO
005260  005742                  TST     -(R2)           ;EVEN BYTE ADDRESS WORD
005262  010022                  MOV     R0,(R2)+        ;LOAD ADDRESS
005264  005200                  INC     R0              ;ODD BYTE ADDRESS
005266  010022                  MOV     R0,(R2)+        ;LOAD ODD BYTE ADDRESS
005270  010200                  MOV     R2,R0           ;RESET R0
005272  010205                  MOV     R2,R5

005274  105152                  COMB    @-(R2)          ;(R0)=177400,CC=1001
005276  103001                  BCC     COMB5
005300  100401                  BMI     .+4
005302  104400          COMB5:  HLT

005304  105752                  TSTB    @-(R2)          ;(R0)=177400, CC=0100
005306  001401                  BEQ     .+4
005310  104400                  HLT

005312  000262                  SEV
005314  106255                  ASRB    @-(R5)          ;(R0)=177400, CC=1001
005316  103002                  BCC     ASRB5
005320  102401                  BVS     ASRB5
005322  100401                  BMI     .+4
005324  104400          ASRB5:  HLT

005326  105232                  INCB    @(R2)+          ;(R0)=177401, CC=000
005330  103001                  BCC     INCB3
005332  100001                  BPL     .+4
005334  104400          INCB3:  HLT

005336  000241                  CLC
005340  106055                  RORB    @-(R5)          ;(R0)=177400, CC=0111
005342  103003                  BCC     RORB5
005344  102002                  BVC     RORB5
005346  001001                  BNE     RORB5
005350  100001                  BPL     .+4
005352  104400          RORB5:  HLT

005354  106332                  ASLB    @(R2)+          ;(R0)=177000, CC=1001
005356  103002                  BCC     ASLB3
005360  102401                  BVS     ASLB3
005362  100401                  BMI     .+4
005364  104400          ASLB3:  HLT

005366  105552                  ADCB    @-(R2)          ;(R0)=177400, CC=1000
005370  103401                  BCS     ADCB5
005372  100401                  BMI     .+4
005374  104400          ADCB5:  HLT

005376  000277                  SCC
```

```
005400  106135                  ROLB    @(R5)+          ;(R0)=177401, CC=0000
005402  101402                  BLOS    ROLB3           ;BRANCH IF C OR Z IS SET
005404  102401                  BVS     ROLB3
005406  100001                  BPL     .+4
005410  104400          ROLB3:  HLT

005412  000352                  SWAB    @-(R2)          ;(R0)=000777, CC=100C
005414  100401                  BMI     .+4
005416  104400                  HLT

005420  000261                  SEC
005422  105635                  SBCB    @(R5)+          ;(R0)=000377, CC=0100
005424  103401                  BCS     SBCB3
005426  001401                  BEQ     .+4
005430  104400          SBCB3:  HLT

005432  105432                  NEGB    @(R2)+          ;(R0)=000001
005434  105352                  DECB    @-(R2)          ;(R0)=000000, CC=0101
005436  103001                  BCC     DECB5
005440  001401                  BEQ     .+4
005442  104400          DECB5:  HLT
005444  104000                  SCOPE

                        ;CHECK UNARY WORD OPS USING ADDRESS MODE 6 (PC)
005446  005027                  CLR     (PC)+           ;PRESET DATA = 0
005450  000000          UWM6:   .WORD   0               ;RESERVED FOR DATA
005452  010700                  MOV     PC,R0
005454  02404C                  CMP     -(R0),-(R0)     ;R0 POINTS TO DATA WORD
005456  000277                  SCC
005460  006167  177764          ROL     UWM6            ;(R0)=000001,CC=0000
005464  103403                  BCS     ROL6
005466  102402                  BVS     ROL6
005470  001401                  BEQ     ROL6
005472  100001                  BPL     .+4
005474  104400          ROL6:   HLT

005476  005167  177746          COM     UWM6            ;(R0)=177776, CC=1001
005502  103002                  BCC     COM6
005504  102401                  BVS     COM6
005506  100401                  BMI     .+4
005510  104400          COM6:   HLT
```

```
005512  006267  177732                  ASR     UWM6            ;(R0)=177777, CC=1010
005516  103402                          BCS     ASR6
005520  102001                          BVC     ASR6
005522  100401                          BMI     .+4
005524  104400              ASR6:       HLT

005526  000277                          SCC
005530  005467  177714                  NEG     UWM6            ;(R0)=000001, CC=0001
005534  103003                          BCC     NEG6
005536  102402                          BVS     NEG6
005540  001401                          BEQ     NEG6
005542  100001                          BPL     .+4
005544  104400              NEG6:       HLT

005546  000277                          SCC
005550  006067  177674                  ROR     UWM6            ;(R0)=100000, CC=1001
005554  103003                          BCC     ROR6
005556  102402                          BVS     ROR6
005560  001401                          BEQ     ROR6
005562  100001                          BMI     .+4
005564  104400              ROR6:       HLT

005566  005067  177656                  SBC     UWM6            ;(R0)=077777, CC=0010
005572  103402                          BCS     SBC6
005574  102001                          BVC     SBC6
005576  100001                          BPL     .+4
005600  104400              SBC6:       HLT

005602  000242                          CLV
005604  005267  177640                  INC     UWM6            ;(R0)=100000, CC=1011
005610  103403                          BCS     INC6
005612  102002                          BVC     INC6
005614  001401                          BEQ     INC6
005616  100401                          BMI     .+4
005620  104400              INC6:       HLT

005622  006267  177622                  ASR     UWM6            ;(R0)=140000, CC=1010
005626  000261                          SEC
005630  006367  177614                  ASL     UWM6            ;(R0)=100000, CC=1001
005634  103002                          BCC     ASL6
005636  102401                          BVS     ASL6
005640  100401                          BMI     .+4
005642  104400              ASL6:       HLT

005644  005367  177600                  DEC     UWM6            ;(R0)=077777, CC=0011
005650  103002                          BCC     DEC6
005652  102001                          BVC     DEC6
005654  100001                          BPL     .+4
005656  104400              DEC6:       HLT

005660  005567  177564                  ADC     UWM6            ;(R0)=100000, CC=1010
005664  103402                          BCS     ADC6
```

```
005666  102001                          BVC     ADC6
005670  100401                          BMI     .+4
005672  104400                  ADC6:   HLT
005674  000242                          CLV
005676  000367  177546                  SWAB    UWM6
005702  100401                          BMI     .+4
005704  104400                          HLT
005706  022710  000200                  CMP     #200,(R0)
005712  001401                          BEQ     .+4
005714  104400                          HLT
005716  104000                          SCOPE

                                ;CHECK UNARY BYTE OPS (EVEN/ODD) USING ADDRESS MODE 6 (PC)
005720  012700  006262                  MOV     #UBM6,R0
005724  063700  001004                  ADD     @#FACTOR,R0     ;R0 POINTS TO ADDRESS OF DATA
005730  005067  000326                  CLR     UBM6            ;CLEAR DATA
005734  000277                          SCC
005736  000244                          CLZ
005740  105767  000316                  TSTB    UBM6
005744  103403                          BCS     TSTB6
005746  102402                          BVS     TSTB6
005750  001001                          BNE     TSTB6
005752  100001                          BPL     .+4
005754  104400                  TSTB6:  HLT

005756  000257                          CCC
005760  105767  000277                  TSTB    UBM6+1          ;TEST ODD BYTE
005764  001401                          BEQ     .+4
005766  104400                          HLT

005770  105667  000266                  SBCB    UBM6            ;(R0)=000000, CC=0100
005774  103402                          BCS     SBCB6
005776  102401                          BVS     SBCB6
006000  001401                          BEQ     .+4
006002  104400                  SBCB6:  HLT

006004  000261                  1$:     SEC
006006  105267  000250                  INCB    UBM5            ;LOOP UNTIL (R0)=077600, CC=1011
006012  100403                          BMI     2$
006014  105567  000243                  ADCB    UBM6+1          ;INCB INST INCREMENTS EVEN BYTE
006020  000771                          BR      1$              ;ADCB INCREMENTS ODD BYTE
006022  103001                  2$:     BCC     INCB6
006024  102401                          BVS     .+4
006026  104400                  INCB6:  HLT

006030  106367  000226                  ASLB    UBM6            ;(R0)=077400, CC=0111
006034  103003                          BCC     ASLB6
006036  102002                          BVC     ASLB6
006040  001001                          BNE     ASLB6
006042  100001                          BPL     .+4
006044  104400                  ASLB6:  HLT

006046  000242                          CLV
006050  105567  000207                  ADCB    UBM6+1          ;(R0)=100000, CC=1010
006054  103402                          BCS     ADCB6
006056  102001                          BVC     ADCB6
```

```
006060  100401                          BMI     .+4
006062  104400              ADCB6:      HLT

006064  000261                          SEC
006066  106067  000171                  RORB    UBM6+1          ;(R0)=140000, CC=1010
006072  103402                          BCS     RORB6
006074  102001                          BVC     RORB6
006076  100401                          BMI     .+4
006100  104400              RORB6:      HLT

006102  105167  000154                  COMB    UBM6            ;(R0)=140377 CC=1001
006106  103002                          BCC     COMB6
006110  102401                          BVS     COMB6
006112  100401                          BMI     .+4
006114  104400              COMB6:      HLT

006116  000262                          SEV
006120  105467  000137                  NEGB    UBM6+1          ;(R0)=040377, CC=0001
006124  103002                          BCC     NEGB6
006126  102401                          BVS     NEGB6
006130  100001                          BPL     .+4
006132  104400              NEGB6:      HLT

006134  106167  000123                  ROLB    UBM6+1          ;(R0)=100777, CC=1010
006140  103402                          BCS     ROLB6
006142  102001                          BVC     ROLB6
006144  100401                          BMI     .+4
006146  104400              ROLB6:      HLT

006150  106267  000106                  ASRB    UBM6            ;(R0)=100777, CC=1001
006154  103002                          BCC     ASRB6
006156  102401                          BVS     ASRB6
006160  100401                          BMI     .+4
006162  104400              ASRB6:      HLT

006164  105267  000072                  INCB    UBM6            ;(R0)=100400, CC=0101
006170  103002                          BCC     INCB6A
006172  102401                          BVS     INCB6A
006174  001401                          BEQ     .+4
006176  104400              INCB6A:     HLT

006200  105367  000057                  DECB    UBM6+1          ;(R0)=100000, CC=1001
006204  103003                          BCC     DECB6A
006206  102402                          BVS     DECB6A
006210  001401                          BEQ     DECB6A
006212  100401                          BMI     .+4
006214  104400              DECB6A:     HLT

006216  000367  000040                  SWAB    UBM6            ;(R0)=000200, CC=1000
006222  103401                          BCS     SWAB6
006224  100401                          BMI     .+4
006226  104400              SWAB6:      HLT

006230  106167  000026                  ROLB    UBM6            ;(R0)=000000, CC=0111
006234  103002                          BCC     ROLB6A
006236  102001                          BVC     ROLB6A
```

```
006240  001401                  BEQ     .+4
006242  104400          ROLB6A: HLT

006244  005767  000012          TST     UBM6            ;(R0)=000000, CC=0100
006250  103402                  BCS     TST6
006252  102401                  BVS     TST6
006254  001401                  BEQ     .+4
006256  104400          TST6:   HLT

006260  000401                  BR      .+4             ;RESERVE A WORD
006262  000000          UBM6:   .WORD   0               ;WORD RESERVED FOR DATA
006264  104000                  SCOPE
006266  010702                  MOV     PC,R2
006270  062702  000012          ADD     #12,R2
006274  012707  001132          MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
006300  000240                  NOP                     ;PROGRAM RETURNS HERE+2
                        ;000000000000 LAST ADDRESS OF CODE TO BE RELOCATED 000000000000
```

```
                                        ;!!!!!!!!!!!!! FIRST ADDRESS TO BE RELOCATED !!!!!!!!!
006302  010700                  RELI:   MOV     PC,R0           ;GET PC
006304  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF RELI
006306  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
006312  010700                          MOV     PC,R0           ;GET CURRENT PC
006314  162700  006314                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
006320  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
006324  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR

                                        ;CHECK UNARY WORD OPS USING ADDRESS MODE 7
006326  000403                          BR      UW7             ;RESERVE 3 WORDS FOR ADDRESSES & DATA
006330  000000                          .WORD   0               ;CONTAINS ADDRESS OF UWM7
006332  000000                  UWM7:   .WORD   0               ;CONTAINS DATA
006334  000000                          .WORD   0               ;CONTAINS ADDRESS OF UWM7

006336  010700                  UW7:    MOV     PC,R0
006340  005740                          TST     -(R0)
006342  005740                          TST     -(R0)
006344  005040                          CLR     -(R0)           ;CLEAR TEST DATA
006346  010002                          MOV     R0,R2
006350  010240                          MOV     R2,-(R0)        ;SET UP ADDRESS
006352  005720                          TST     (R0)+           ;MOVE R0 TO NEXT ADDRESS
006354  005720                          TST     (R0)+
006356  010210                          MOV     R2,(R0)         ;SET NEXT ADDRESS
006360  010200                          MOV     R2,R0           ;SET R0 POINTING TO DATA
006362  000277                          SCC
006364  000244                          CLZ
006366  005772  000002                  TST     @2(2)           ;(R0)=000000, CC=0100
006372  001401                          BEQ     .+4
006374  104400                          HLT

006376  000277                          SCC
006400  005672  177776                  SBC     @-2(2)          ;(R0)=177777, CC=1001
006404  103002                          BCC     SBC7
006406  102401                          BVS     SBC7
006410  100401                          BMI     .+4
006412  104400                  SBC7:   HLT

006414  000277                          SCC
006416  000241                          CLC
006420  006372  000002                  ASL     @2(2)           ;(R0)=177776, CC=1001
006424  103002                          BCC     ASL7
006426  102401                          BVS     ASL7
006430  100401                          BMI     .+4
006432  104400                  ASL7:   HLT

006434  000257                          CCC
006436  005372  000002                  DEC     @2(2)           ;(R0)=177775, CC=.
006442  103402                          BCS     DEC7
006444  102401                          BVS     DEC7
006446  100401                          BMI     .+4
006450  104400                  DEC7:   HLT

006452  000262                          SEV
```

```
006454  006272  177776          ASR     @-2(2)          ;(R0)=177776, CC=1001
006460  103002                  BCC     ASR7
006462  102401                  BVS     ASR7
006464  100401                  BMI     .+4
006466  104400          ASR7:   HLT

006470  000241                  CLC
006472  000262                  SEV
006474  006072  177776          ROR     @-2(2)          ;(R0)=077777, CC=0000
006500  101402                  BLOS    ROR7            ;BRANCH IF C OR Z IS SET
006502  102401                  BVS     ROR7
006504  100001                  BPL     .+4
006506  104400          ROR7:   HLT

006510  000262                  SEV
006512  005472  000002          NEG     @2(2)           ;(R0)=100001, CC=1001
006516  103002                  BCC     NEG7
006520  102401                  BVS     NEG7
006522  100401                  BMI     .+4
006524  104400          NEG7:   HLT

006526  000250                  CLN
006530  000372  177776          SWAB    @-2(2)          ;(R0)=000600, CC=1000
006534  103401                  BCS     SWAB7
006536  100401                  BMI     .+4
006540  104400          SWAB7:  HLT

006542  000262                  SEV
006544  005172  000002          COM     @2(2)           ;(R0)=177177, CC=1001
006550  103002                  BCC     COM7
006552  102401                  BVS     COM7
006554  100401                  BMI     .+4
006556  104400          COM7:   HLT

006560  000372  000002          SWAB    @2(2)           ;(R0)=077776, CC=1000
006564  100401                  BMI     .+4
006566  104400                  HLT

006570  000277                  SCC
006572  005572  177776          ADC     @-2(2)          ;(R0)=077777, CC=0000
006576  103402                  BCS     ADC7
006600  102401                  BVS     ADC7
006602  100001                  BPL     .+4
006604  104400          ADC7:   HLT

006606  005272  000002          INC     @2(2)           ;(R0)=100000, CC=1010
006612  102001                  BVC     INC7
006614  100401                  BMI     .+4
006616  104400          INC7:   HLT

006620  000257                  CCC
006622  006172  177776          ROL     @-2(2)          ;(R0)=000000, CC=0111
006626  103002                  BCC     ROL7
006630  102001                  BVC     ROL7
006632  001400                  BEQ     .+4
006634  104400          ROL7:   HLT
```

```
006636  104000                          SCOPE

                                ;CHECK UNARY BYTE OPS USING ADDRESS MODE 7
006640  005720                          TST     (R0)+
006642  005210                          INC     (R0)            ;WORD FOLLOWING UWM7 CONTAINS ADDRESS
006644  005740                          TST     -(R0)           ;OF ODD BYTE, R0 POINTS TO DATA WORD
006646  005010                          CLR     (R0)            ;PRESET DATA
006650  010701                          MOV     PC,R1           ;SET SCOPE PTR
                                ;NOTE:  @2(2) REFERENCES THE ODD BYTE, AND @-2(2) REFERENCES THE EVEN BYTE.

006652  000263                          +SEC!SEV                ;SET C AND V
006654  105672  000002                  SBCB    @2(2)           ;(R0)=177400, CC=1001
006660  103003                          BCC     SBCB7
006662  102402                          BVS     SBCB7
006664  001401                          BEQ     SBCB7
006666  100401                          BMI     .+4
006670  104400                SBCB7:    HLT

006672  000277                          SCC                     ;SET CONDITION CODES
006674  105572  177776                  ADCB    @-2(2)          ;(R0)=177401, CC=0000
006700  103403                          BCS     ADCB7
006702  102402                          BVS     ADCB7
006704  001401                          BEQ     ADCB7
006706  100001                          BPL     .+4
006710  104400                ADCB7:    HLT

006712  105172  177776                  COMB    @-2(2)          ;(R0)=177776, CC=1001
006716  103002                          BCC     COMB7
006720  102401                          BVS     COMB7
006722  100401                          BMI     .+4
006724  104400                COMB7:    HLT

006726  000241                          CLC                     ;CLEAR CARRY
006730  106072  000002                  RORB    @2(2)           ;(R0)=077776, CC=0011
006734  103002                          BCC     RORB7
006736  102001                          BVC     RORB7
006740  100001                          BPL     .+4
006742  104400                RORB7:    HLT

006744  105272  000002                  INCB    @2(2)           ;(R0)=100376, CC=1011
006750  103002                          BCC     INCB7
006752  102001                          BVC     INCB7
006754  100401                          BMI     .+4
006756  104400                INCB7:    HLT

006760  105372  177776                  DECB    @-2(2)          ;(R0)=100375, CC=1001
006764  103002                          BCC     DECB7
006766  102401                          BVS     DECB7
006770  100401                          BMI     .+4
006772  104400                DECB7:    HLT

006774  106372  000002                  ASLB    @2(2)           ;(R0)=000375, CC=0111
007000  103002                          BCC     ASLB7
007002  102001                          BVC     ASLB7
007004  001401                          BEQ     .+4
007006  104400                ASLB7:    HLT
```

```
007010  000241                          CLC                     ;CLEAR CARRY
007012  106272  177776                  ASRB    @-2(2)          ;(R0)=000376, CC=1001
007016  103002                          BCC     ASRB7
007020  102401                          BVS     ASRB7
007022  100401                          BMI     .+4
007024  104400                  ASRB7:  HLT

007026  105472  000002                  NEGB    @2(2)           ;(R0)=000376, CC=0100
007032  103402                          BCS     NEGB7
007034  102401                          BVS     NEGB7
007036  001401                          BEQ     .+4
007040  104400                  NEGB7:  HLT

007042  000262                          SEV
007044  106172  177776                  ROLB    @-2(2)          ;(R0)=000374, CC=1001
007050  103002                          BCC     ROLB7
007052  102401                          BVS     ROLB7
007054  100401                          BMI     .+4
007056  104400                  ROLB7:  HLT

007060  105272  177776                  INCB    @-2(2)          ;(R0)=000375, CC=100
007064  105272  177776                  INCB    @-2(2)          ;(R0)=000376, CC=100
007070  105572  177776                  ADCB    @-2(2)          ;(R0)=000377, CC=1000
007074  105172  177776                  COMB    @-2(2)          ;(R0)=000000, CC=0100
007100  001401                          BEQ     .+4
007102  104400                          HLT
007104  104000                          SCOPE

                                :CHECK BINARY OPS  USING ADDRESS MODE 0
007106  000277                          SCC                     ;SET CONDITION CODES
007110  010700                          MOV     PC,R0           ;R0=PC, CC=X001
007112  103002                          BCC     MOV0
007114  102401                          BVS     MOV0
007116  001001                          BNE     .+4
007120  104400                  MOV0:   HLT

007122  010002                          MOV     R0,R2           ;R2=R0
007124  000262                          SEV                     ;SET V
007126  160002                          SUB     R0,R2           ;R2=000000, CC=0100
007130  103402                          BCS     SUB0
007132  102401                          BVS     SUB0
007134  001401                          BEQ     .+4
007136  104400                  SUB0:   HLT

007140  000244                          CLZ
007142  010203                          MOV     R2,R3           ;R2=R3=000000, CC=0100
007144  103401                          BCS     MOV0A
007146  001401                          BEQ     .+4
007150  104400                  MOV0A:  HLT

007152  000257                          CCC
007154  000272                          +SEV!SEN                ;SET V & N
007156  020203                          CMP     R2,R3           ;R2=R3=000000, CC=0100
007160  103403                          BCS     CMP0
007162  102402                          BVS     CMP0
```

```
007164  001001                          BNE     CMP0
007166  100001                          BPL     .+4
007170  104400                  CMP0:   HLT

007172  010002                          MOV     R0,R2           ;R0=R2
007174  010203                          MOV     R2,R3           ;R0=R2=R3
007176  060203                          ADD     R2,R3           ;R3=2*R0
007200  006302                          ASL     R2              ;R2=2*R0
007202  020203                          CMP     R2,R3           ;R2=R3=2*R0
007204  001401                          BEQ     .+4
007206  104400                          HLT                     ;ERROR! CHECK ADD INSTRUCTION

                                ;THE FOLLOWING SUBTEST SHIFTS A BIT THROUGH R2 AND R5 AND DOES A
                                ;BIT TEST (BIT) USING R2 AND R5.
007210  005002                          CLR     R2
007212  005202                          INC     R2
007214  000402                          BR      2$
007216  006302                  1$:     ASL     R2
007220  100407                          BMI     4$
007222  010205                  2$:     MOV     R2,R5
007224  000277                          SCC
007226  030205                          BIT     R2,R5           ;R2=R5
007230  103002                          BCC     3$
007232  102401                          BVS     3$
007234  001370                          BNE     1$
007236  104400                  3$:     HLT
007240  010205                  4$:     MOV     R2,R5
007242  000257                          CCC
007244  030205                          BIT     R2,R5
007246  100401                          BMI     .+4
007250  104400                          HLT

007252  005002                          CLR     R2
007254  000277                          SCC
007256  050002                          BIS     R0,R2
007260  103002                          BCC     BIS0
007262  102401                          BVS     BIS0
007264  001001                          BNE     .+4
007266  104400                  BIS0:   HLT

007270  010003                          MOV     R0,R3
007272  000277                          SCC
007274  000244                          CLZ
007276  040003                          BIC     R0,R3
007300  103003                          BCC     BIC0
007302  102402                          BVS     BIC0
007304  001001                          BNE     BIC0
007306  100001                          BPL     .+4
007310  104400                  BIC0:   HLT

007312  010004                          MOV     R0,R4
007314  005104                          COM     R4
007316  040004                          BIC     R0,R4
007320  005104                          COM     R4
007322  020004                          CMP     R0,R4
007324  001401                          BEQ     .+4
```

```
007326  104400                          HLT

007330  010004                          MOV     R0,R4
007332  005104                          COM     R4
007334  010403                          MOV     R4,R3
007336  050003                          BIS     R0,R3
007340  103001                          BCC     BISOA
007342  100401                          BMI     .+4
007344  104400                  BISOA:  HLT
007346  005203                          INC     R3
007350  001401                          BEQ     .+4
007352  104400                          HLT
007354  010304                          MOV     R3,R4           ;R3=R4=0
007356  005103                          COM     R3              ;R3=177777
007360  000261                          SEC                     ;SET C
007362  006004                          ROR     R4              ;R4=100000
007364  060304                          ADD     R3,R4           ;R3=177777,R4=077777, CC=0011
007366  103003                          BCC     ADDO
007370  102002                          BVC     ADDO
007372  001401                          BEQ     ADDO
007374  100001                          BPL     .+4
007376  104400                  ADDO:   HLT
007400  010700                          MOV     PC,R0
007402  022020                          CMP     (R0)+,(R0)+
007404  020007                          CMP     R0,PC
007406  001401                          BEQ     .+4
007410  104400                          HLT

007412  010700                          MOV     PC,R0
007414  062700  000010                  ADD     #10,R0
007420  010002                          MOV     R0,R2
007422  020700                          CMP     PC,R0
007424  001002                          BNE     CMPOA
007426  020200                          CMP     R2,R0
007430  001401                          BEQ     .+4
007432  104400                  CMPOA:  HLT
007434  104000                          SCOPE

                                ;CHECK BINARY BYTE OPS USING ADDRESS MODE 0.
007436  012703  125252                  MOV     #125252,R3
007442  010304                          MOV     R3,R4           ;R3=R4=125252
007444  140304                          BICB    R3,R4           ;R3=125252, R4=125000
007446  022704  125000                  CMP     #125000,R4
007452  001401                          BEQ     .+4
007454  104400                          HLT                     ;ERROR! BICB FAILED

007456  005004                          CLR     R4              ;R3=125252, R4=0
007460  150304                          BISB    R3,R4           ;R3=125252, R4=000252
007462  022704  000252                  CMP     #252,R4
007466  001401                          BEQ     .+4
007470  104400                          HLT                     ;ERROR! BISB FAILED

007472  110404                          MOVB    R4,R4           ;R4=177652
007474  022704  177652                  CMP     #177652,R4      ;MOVB EXTENDS THE SIGN
007500  001401                          BEQ     .+4
007502  104400                          HLT                     ;ERROR! MOVB FAILED
```

```
007504  132704  177525                  BITB    #177525,R4
007510  001401                          BEQ     .+4
007512  104400                          HLT                     ;ERROR! BITB FAILED

007514  105104                          COMB    R4              ;R4=177525
007516  110404                          MOVB    R4,R4           ;R4=000125
007520  022704  000125                  CMP     #125,R4
007524  001401                          BEQ     .+4
007526  104400                          HLT

007530  150304                          BISB    R3,R4           ;R3=125252, R4=000377
007532  105204                          INCB    R4
007534  005704                          TST     R4
007536  001401                          BEQ     .+4
007540  104400                          HLT
007542  104000                          SCOPE


                              ;CHECK BINARY OPS USING ADDRESS MODE 1
007544  000402                          BR      .+6             ;RESERVE TWO WORDS
007546  000000                          .WORD   0               ;RESERVED FOR SOURCE DATA
007550  000000                          .WORD   0               ;RESERVED FOR DESTINATION DATA
007552  010704                          MOV     PC,R4
007554  005744                          TST     -(R4)
007556  005044                          CLR     -(R4)           ;R4 POINTS TO DESTINATION DATA
007560  010403                          MOV     R4,R3
007562  005043                          CLR     -(R3)           ;R3 POINTS TO SOUCE DATA

007564  005113                          COM     (R3)            ;(R3)=177777
007566  005214                          INC     (R4)            ;(R4)=000001
007570  000262                          SEV                     ;SET V
007572  061314                          ADD     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=0101
007574  103002                          BCC     ADD1
007576  102401                          BVS     ADD1
007600  001401                          BEQ     .+4
007602  104400                  ADD1:   HLT

007604  000277                          SCC
007606  000250                          CLN
007610  021314                          CMP     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=1000
007612  103403                          BCS     CMP1
007614  102402                          BVS     CMP1
007616  001401                          BEQ     CMP1
007620  100401                          BMI     .+4
007622  104400                  CMP1:   HLT

007624  000277                          SCC
007626  000244                          CLZ
007630  031314                          BIT     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=0101
007632  103002                          BCC     BIT1
007634  102401                          BVS     BIT1
007636  001401                          BEQ     .+4
007640  104400                  BIT1:   HLT

007642  000277                          SCC
```

```
007644  000245                          +CLC!CLZ
007646  005114                          COM     (R4)            ;(R4)=177777
007650  161314                          SUB     (R3),(R4)       ;(R3)=177777,(R4)=000000, CC=0100
007652  103402                          BCS     SUB1
007654  102401                          BVS     SUB1
007656  001401                          BEQ     .+4
007660  104400              SUB1:       HLT

007662  105013                          CLRB    (R3)            ;(R3)=177400
007664  000313                          SWAB    (R3)            ;(R3)=000377
007666  000270                          SEN
007670  011314                          MOV     (R3),(R4)       ;(R3)=(R4)=000377
007672  100001                          BPL     .+4
007674  104400                          HLT
007676  000314                          SWAB    (R4)            ;(R3)=000377,(R4)=177400
007700  000263                          +SEC!SEV                ;SET C & V
007702  051314                          BIS     (R3),(R4)       ;(R3)=000377,(R4)=177777, CC=1001
007704  103002                          BCC     BIS1
007706  102401                          BVS     BIS1
007710  100401                          BMI     .+4
007712  104400              BIS1:       HLT

007714  041314                          BIC     (R3),(R4)       ;(R3)=000377,(R4)=177400, CC=1001
007716  103002                          BCC     BIC1
007720  102401                          BVS     BIC1
007722  100401                          BMI     .+4
007724  104400              BIC1:       HLT

007726  000262                          SEV                     ;SET V
007730  021314                          CMP     (R3),(R4)       ;(R3)=000377,(R4)=177400, CC=0001
007732  103003                          BCC     CMP1A
007734  102402                          BVS     CMP1A
007736  001401                          BEQ     CMP1A
007740  100001                          BPL     .+4
007742  104400              CMP1A:      HLT

007744  005013                          CLR     (R3)            ;(R3)=000000
007746  000261                          SEC
007750  006013                          ROR     (R3)            ;(R3)=100000
007752  011314                          MOV     (R3),(R4)       ;(R3)=(R4)=100000
007754  005114                          COM     (R4)            ;(R4)=077777
007756  161314                          SUB     (R3),(R4)       ;(R3)=100000,(R4)=177777, CC=1011
007760  103002                          BCC     SUB1A
007762  102001                          BVC     SUB1A
007764  100401                          BMI     .+4
007766  104400              SUB1A:      HLT

007770  000277                          SCC
007772  161314                          SUB     (R3),(R4)       ;(R3)=100000,(R4)=077777, CC=0000
007774  101402                          BLOS    SUB1B           ;BRANCH IF C OR Z IS SET
007776  102401                          BVS     SUB1B
010000  100001                          BPL     .+4
010002  104400              SUB1B:      HLT

010004  011314                          MOV     (R3),(R4)       ;(R3)=100000,(R4)=100000, CC=1000
010006  001401                          BEQ     MOV1
```

```
010010  100401                  BMI     .+4
010012  104400          MOV1:   HLT

010014  061314                  ADD     (R3),(R4)       ;(R3)=100000,(R4)=000000, CC=0111
010016  103003                  BCC     ADD1A
010020  102002                  BVC     ADD1A
010022  001001                  BNE     ADD1A
010024  100001                  BPL     .+4
010026  104400          ADD1A:  HLT

010030  005113                  COM     (R3)            ;(R3)=077777
010032  011314                  MOV     (R3),(R4)               ;(R4)=077777
010034  061314                  ADD     (R3),(R4)       ;(R3)=077777,(R4)=177776, CC=1010
010036  103402                  BCS     ADD1B
010040  102001                  BVC     ADD1B
010042  100401                  BMI     .+4
010044  104400          ADD1B:  HLT

010046  062714  000002          ADD     #2,(R4)
010052  005714                  TST     (R4)            ;CHECK FINAL RESULT
010054  001401                  BEQ     .+4
010056  104400                  HLT
010060  104000                  SCOPE

                        ;CHECK BINARY BYTE OPS USING ADDRESS MODE 1
010062  000402                  BR      .+6
010064  000000                  .WORD   0
010066  000000                  .WORD   0
010070  010705                  MOV     PC,R5
010072  005745                  TST     -(R5)
010074  005045                  CLR     -(R5)           ;(R5)=000000
010076  010502                  MOV     R5,R2
010100  005042                  CLR     -(R2)           ;(R2)=000000
010102  005202                  INC     R2              ;R2 POINTS TO ODD BYTE
010104  105112                  COMB    (R2)            ;(R2)=177400

010106  000277                  SCC
010110  111215                  MOVB    (R2),(R5)       ;(R2)=177400,(R5)=000377,CC=1001
010112  103005                  BCC     MOVB1
010114  102404                  BVS     MOVB1
010116  001403                  BEQ     MOVB1
010120  100002                  BPL     MOVB1
010122  105215                  INCB    (R5)            ;CHECK RESULT
010124  001401                  BEQ     .+4
010126  104400          MOVB1:  HLT

010130  106312                  ASLB    (R2)            ;SHIFT (R2) UNTIL
010132  102376                  BVC     .-2             ;(R2)=000000
010134  106012                  RORB    (R2)            ;(R2)=100000
010136  105315                  DECB    (R5)            ;(R5)=00377
010140  106015                  RORB    (R5)            ;(R5)=000177
010142  000257                  CCC
010144  121512                  CMPB    (R5),(R2)       ;(R5)=000177,(R2)=100000, CC=1010
010146  102001                  BVC     CMPB1
010150  100401                  BMI     .+4
010152  104400          CMPB1:  HLT
```

```
010154  005003                          CLR     R3
010156  000261                          SEC
010160  006003                          ROR     R3              ;R3=100000
010162  050315                          BIS     R3,(R5)         ;(R5)=100177
010164  000273                          +SEC!SEV!SEN            ;SET C,V, & N
010166  131215                          BITB    (R2),(R5)       ;(R2)=100000,(R5)=100177, CC=0101
010170  103002                          BCC     BITB1
010172  102401                          BVS     BITB1
010174  001401                          BEQ     .+4
010176  104400                  BITB1:  HLT

010200  151215                          BISB    (R2),(R5)       ;(R2)=100000,(R5)=100377, CC=1001
010202  103001                          BCC     BISB1
010204  100401                          BMI     .+4
010206  104400                  BISB1:  HLT

010210  141215                          BICB    (R2),(R5)       ;(R2)=100000,(R5)=100177, CC=0001
010212  103002                          BCC     BICB1
010214  001401                          BEQ     BICB1
010216  100001                          BPL     .+4
010220  104400                  BICB1:  HLT

010222  105112                          COMB    (R2)            ;(R2)=077400,(R5)=100177
010224  121215                          CMPB    (R2),(R5)
010226  001401                          BEQ     .+4
010230  104400                          HLT

010232  141512                          BICB    (R5),(R2)       ;(R5)=100177,(R2)=000000, CC=0100
010234  001002                          BNE     BICB1A
010236  105712                          TSTB    (R2)
010240  001401                          BEQ     .+4
010242  104400                  BICB1A: HLT

010244  000402                          BR      .+6             ;RESERVE TWO WORDS FOR DATA
010246  000000                          .WORD   0               ;SOURCE DATA
010250  000000                          .WORD   0               ;DEST DATA
010252  010705                          MOV     PC,R5
010254  005745                          TST     -(R5)
010256  105045                          CLRB    -(R5)           ;R5 POINTS TO DEST ODD BYTE
010260  010504                          MOV     R5,R4
010262  105044                          CLRB    -(R4)           ;R4 POINTS TO DEST EVEN BYTE
010264  010403                          MOV     R4,R3
010266  105043                          CLRB    -(R3)           ;R3 POINTS TO SOURCE ODD BYTE
010270  010302                          MOV     R3,R2
010272  105042                          CLRB    -(R2)           ;R2 POINTS TO SOURCE EVEN BYTE

                                ;COMMENTS ARE LEAST SIGNIFICANT 4 BITS OF BYTES POINTED TO BY R2,R3
                                ;R4, AND R5 RESPECTIVELY AND THE REMAINING BITS ARE 0'S.
010274  000261                          SEC                     ;SET CARRY
                                                                ;(R2),(R3),(R4),(R5)
010276  106112                          ROLB    (R2)            ;0001,0000,0000,0000
010300  111214                          MOVB    (R2),(R4)       ;0001,0000,0001,0000
010302  106112                          ROLB    (R2)            ;0010,0000,0001,0000
010304  111213                          MOVB    (R2),(R3)       ;0010,0010,0001,0000
010306  106112                          ROLB    (R2)            ;0100,0010,0001,0000
```

```
010310  111315                  MOVB    (R3),(R5)       ;0100,0010,0001,0010
010312  106112                  ROLB    (R2)            ;1000,0010,0001,0010
010314  106113                  ROLB    (R3)            ;1000,0100,0001,0010
010316  151215                  BISB    (R2),(R5)       ;1000,0100,0001,1010
010320  131512                  BITB    (R5),(R2)       ;1000,0100,0001,1010
010322  001426                  BEQ     BIN1
010324  151314                  BISB    (R3),(R4)       ;1000,0100,0101,1010
010326  131413                  BITB    (R4),(R3)       ;1000,0100,0101,1010
010330  001423                  BEQ     BIN1
010332  105213                  INCB    (R3)            ;1000,0101,0101,1010
010334  121314                  CMPB    (R3),(R4)       ;1000,0101,0101,1010
010336  001020                  BNE     BIN1
010340  106113                  ROLB    (R3)            ;1000,1010,0101,1010
010342  121315                  CMPB    (R3),(R5)       ;1000,1010,0101,1010
010344  001015                  BNE     BIN1
010346  106212                  ASRB    (R2)            ;0100,1010,0101,1010
010350  131214                  BITB    (R2),(R4)       ;0100,1010,0101,1010
010352  001412                  BEQ     BIN1
010354  106015                  RORB    (R5)            ;0100,1010,0101,0101
010356  121415                  CMPB    (R4),(R5)       ;0100,1010,0101,0101
010360  001007                  BNE     BIN1
010362  105314                  DECB    (R4)            ;0100,1010,0100,0101
010364  141214                  BICB    (R2),(R4)       ;0100,1010,0000,0101
010366  001004                  BNE     BIN1
010370  111314                  MOVB    (R3),(R4)       ;0100,1010,1010,0101
010372  106213                  ASRB    (R3)            ;0100,0101,1010,0101
010374  141315                  BICB    (R3),(R5)       ;0100,0101,1010,0101
010376  001401                  BEQ     .+4
010400  104400          BIN1:   HLT
010402  104000                  SCOPE

                        ;CHECK BINARY WORD OPS USING ADDRESS MODE 2 & 4
010404  010405                  MOV     R4,R5           ;SET DESTINATION REGISTER
010406  012715  000001          MOV     #1,(R5)
010412  012712  177777          MOV     #-1,(R2)
010416  000257                  CCC
010420  000262                  SEV
010422  062225                  ADD     (R2)+,(R5)+     ;(R2)=177777,(R5)=000000, CC=0101
010424  103002                  BCC     ADD2
010426  102401                  BVS     ADD2
010430  001401                  BEQ     .+4
010432  104400          ADD2:   HLT

010434  000262                  SEV                     ;SET V
010436  024527  000001          CMP     -(R5),#1        ;(R5)=000000, CC=1001
010442  103002                  BCC     CMP2
010444  102401                  BVS     CMP2
010446  100401                  BMI     .+4
010450  104400          CMP2:   HLT

010452  054225                  BIS     -(R2),(R5)+     ;(R2)=177777,(R5)=177777,CC=1001
010454  103001                  BCC     BIS2
010456  100401                  BMI     .+4
010460  104400          BIS2:   HLT
010462  000277                  SCC
010464  000244                  CLZ
```

```
010466  162245                          SUB     (R2)+,-(R5)     ;(R2)=177777,(R5)=000000, CC=0100
010470  103402                          BCS     SUB2
010472  102401                          BVS     SUB2
010474  001401                          BEQ     .+4
010476  104400                  SUB2:   HLT

010500  005442                          NEG     -(R2)           ;(R2)=000001
010502  005115                          COM     (R5)            ;(R5)=177777
010504  000277                          SCC
010506  000250                          CLN
010510  042225                          BIC     (R2)+,(R5)+     ;(R2)=000001,(R5)=177776, CC=1001
010512  103003                          BCC     BIC2
010514  102402                          BVS     BIC2
010516  001401                          BEQ     BIC2
010520  100401                          BMI     .+4
010522  104400                  BIC2:   HLT

010524  012742  125252                  MOV     #125252,-(R2)
010530  012245                          MOV     (R2)+,-(R5)
010532  005125                          COM     (R5)+           ;(R5)=052525
010534  000262                          SEV
010536  034245                          BIT     -(R2),-(R5)     ;(R2)=125252,(R5)=052525, CC=0101
010540  103002                          BCC     BIT2
010542  102401                          BVS     BIT2
010544  001401                          BEQ     .+4
010546  104400                  BIT2:   HLT

010550  000262                          SEV
010552  052225                          BIS     (R2)+,(R5)+     ;(R2)=125252,(R5)=177777, CC=1001
010554  103002                          BCC     BIS2A
010556  102401                          BVS     BIS2A
010560  100401                          BMI     .+4
010562  104400                  BIS2A:  HLT

010564  042745  125252                  BIC     #125252,-(R5)   ;(R5)=052525
010570  005125                          COM     (R5)+           ;(R5)=125252
010572  024245                          CMP     -(R2),-(R5)
010574  001401                          BEQ     .+4
010576  104400                          HLT

010600  005012                          CLR     (R2)
010602  005122                          COM     (R2)+           ;(R2)=177777
010604  162742  000001                  SUB     #1,-(R2)        ;(R2)=177775, CC=1000
010610  103402                          BCS     SUB2A
010612  102401                          BVS     SUB2A
010614  100401                          BMI     .+4
010616  104400                  SUB2A:  HLT
010620  104000                          SCOPE

010622  010702                          MOV     PC,R2           ;GET CURRENT PC
010624  010205                          MOV     R2,R5           ;MOVE TO R5
010626  124245                  1$:     CMPB    -(R2),-(R5)     ;COMPARE ALL PREVIOUS MEMORY ADDRESSES
010630  001401                          BEQ     .+4
010632  104400                          HLT                     ;ERROR!
010634  020237  001010                  CMP     R2,@#FRSTAD     ;CHECK FOR LOW LIMIT
010640  001372                          BNE     1$
```

```
010642  104000                          SCOPE

                                ;CHECK BINARY BYTE OPS USING ADDRESS MODES 2 & 4.
010644  000402                  BR      .+6             ;RESERVE TWO WORDS
010646  000000                  .WORD   0               ;SOURCE DATA
010650  000000                  .WORD   0               ;DESTINATION DATA
010652  010703                  MOV     PC,R3
010654  005743                  TST     -(R3)
010656  112743  000200          MOVB    #200,-(R3)
010662  112743  000377          MOVB    #377,-(R3)      ;(R3)=100377
010666  010304                  MOV     R3,R4
010670  112744  000177          MOVB    #177,-(R4)
010674  112744  000000          MOVB    #0,-(R4)        ;(R4)=077400
010700  001401                  BEQ     .+4
010702  104400                  HLT

010704  152324                  BISB    (R3)+,(R4)+     ;(R3)=100377,(R4)=077777
010706  100401                  BMI     .+4
010710  104400                  HLT

010712  122324                  CMPB    (R3)+,(R4)+
010714  103402                  BCS     CMPB2
010716  102001                  BVC     CMPB2
010720  100001                  BPL     .+4
010722  104400          CMPB2:  HLT

010724  000261                  SEC
010726  134344                  BITB    -(R3),-(R4)
010730  103002                  BCC     BITB2
010732  102401                  BVS     BITB2
010734  001401                  BEQ     .+4
010736  104400          BITB2:  HLT

010740  000244                  CLZ
010742  144344                  BICB    -(R3),-(R4)     ;(R3)=100377,(R4)=077400
010744  001401                  BEQ     .+4
010746  104400                  HLT
010750  104000                  SCOPE

                                ;CHECK BINARY WORD OPS USING ADDRESS MODES 3 & 5.
010752  000404                  BR      2$              ;RESERVE SPACE FOR DATA AND ADDRESSES
010754  000000                  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA
010756  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA
010760  000000                  .WORD   0               ;CONTAINS SOURCE DATA
010762  000000                  .WORD   0               ;CONTAINS DEST DATA
010764  010701          2$:     MOV     PC,R1
010766  010100                  MOV     R1,R0           ;SET SCOPE PTR
010770  024040                  CMP     -(R0),-(R0)     ;ADJUST R0
010772  010005                  MOV     R0,R5           ;R5 POINTS TO DEST DATA
010774  024545                  CMP     -(R5),-(R5)     ;SUB 4 FROM R5
010776  010015                  MOV     R0,(R5)         ;R5 POINTS TO ADDRESS OF DEST DATA
011000  010502                  MOV     R5,R2
011002  010004                  MOV     R0,R4           ;R4 POINTS TO DEST DATA
011004  005740                  TST     -(R0)
011006  010003                  MOV     R0,R3           ;R3 POINTS TO SOURCE DATA
011010  010042                  MOV     R0,-(R2)        ;R2 POINTS TO ADDRESS OF SOURCE DATA
```

```
011012  005013                          CLR     (R3)            ;PRESET SOURCE DATA
011014  005014                          CLR     (R4)            ;PRESET DEST DATA

011016  000277                          SCC
011020  000244                          CLZ
011022  163235                          SUB     @(R2)+,@(R5)+   ;(R3)=000000,(R4)=000000, CC=0100
011024  103402                          BCS     SUB3
011026  102401                          BVS     SUB3
011030  001401                          BEQ     .+4
011032  104400                  SUB3:   HLT

011034  052752  100000                  BIS     #100000,@-(R2)  ;(R3)=100000
011040  062755  000001                  ADD     #1,@-(R5)       ;(R4)=000001
011044  163235                          SUB     @(R2)+,@(R5)+   ;(R3)=100000,(R4)=100001, CC=1011
011046  103002                          BCC     SUB3A
011050  102001                          BVC     SUB3A
011052  100401                          BMI     .+4
011054  104400                  SUB3A:  HLT

011056  005414                          NEG     (R4)            ;(R4)=077777
011060  035255                          BIT     @-(R2),@-(R5)   ;(R3)=100000,(R4)=077777
011062  001401                          BEQ     .+4
011064  104400                          HLT
011066  023235                          CMP     @(R2)+,@(R5)+
011070  102401                          BVS     .+4
011072  104400                          HLT
011074  005152                          COM     @-(R2)
011076  000257                          CCC
011100  063255                          ADD     @(R2)+,@-(R5)
011102  102001                          BVC     ADD3
011104  100401                          BMI     .+4
011106  104400                  ADD3:   HLT
011110  000261                          SEC
011112  045235                          BIC     @-(R2),@(R5)+   ;(R3)=077777,(R4)=100000
011114  103001                          BCC     BIC3
011116  100401                          BMI     .+4
011120  104400                  BIC3:   HLT

011122  005                             COM     @-(R5)          ;(R4)=077777
011124  02                              CMP     @(R2)+,@(R5)+   ;(R3)=077777,(R4)=077777
011126  00                              BEQ     .+4
011130  104400                          HLT
011132  104000                          SCOPE

                                ;CHECK BINARY BYTE OPS USING ADDRESS MODES 3 & 5.
011134  000406                          BR      1$              ;RESERVE SPACE FOR ADDRESSES & DATA
011136  000000                  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA (EVEN BYTE)
011140  000000                  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA (ODD BYTE)
011142  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA (EVEN BYTE)
011144  000000                  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA (ODD BYTE)
011146  000000                  .WORD   0               ;CONTAINS SOURCE DATA
011150  000000                  .WORD   0               ;CONTAINS DEST DATA

011152  010700                  1$:     MOV     PC,R0
011154  024040                          CMP     -(R0),-(R0)     ;R0=ADDRESS OF DEST DATA .
011156  010003                          MOV     R0,R3           ;R3      "       "
```

```
011160  010305                  MOV     R3,R5           ;R5     "       "
011162  005743                  TST     -(R3)           ;SUB 2 FROM R3
011164  010043                  MOV     R0,-(R3)        ;R3 POINTS TO ADDRESS OF DEST DATA
011166  005213                  INC     (R3)            ;ODD BYTE
011170  010043                  MOV     R0,-(R3)        ;EVEN BYTE
011172  010304                  MOV     R3,R4
011174  005740                  TST     -(R0)           ;R0=ADDRESS OF SOURCE DATA
011176  010044                  MOV     R0,-(R4)        ;R4 POINTS TO ADDRESS OF SOURCE DATA
011200  005214                  INC     (R4)            ;ODD BYTE
011202  010044                  MOV     R0,-(R4)        ;EVEN BYTE

011204  000261                  SEC                     ;SET CARRY
011206  012734  177001          MOV     #177001,@(R4)+
011212  112734  000200          MOVB    #200,@(R4)+     ;SOURCE DATA=100001
011216  115433                  MOVB    @-(R4),@(R3)+
011220  115433                  MOVB    @-(R4),@(R3)+   ;DEST DATA=000600
011222  103401                  BCS     .+4
011224  104000                  HLT                     ;ERROR! MOV DOES AFFECT C BIT IN PSW
011226  022715  000600          CMP     #600,(R5)       ;CHECK DEST DATA
011232  001401                  BEQ     .+4
011234  104000                  HLT                     ;ERROR! INCORRECT RESULT
011236  024343                  CMP     -(R3),-(R3)     ;POINT R4 BACK TO EVEN BYTE
011240  153433                  BISB    @(R4)+,@(R3)+
011242  153433                  BISB    @(R4)+,@(R3)+   ;DEST DATA=100601
011244  022715  100601          CMP     #100601,(R5)    ;CHECK RESULT
011250  001401                  BEQ     .+4
011252  104000                  HLT                     ;ERROR! INCORRECT DEST DATA AFTER BISB
011254  145453                  BICB    @-(R4),@-(R3)
011256  145453                  BICB    @-(R4),@-(R3)
011260  133433                  BITB    @(R4)+,@(R3)+
011262  001002                  BNE     BITB3
011264  135433                  BITB    @-(R4),@(R3)+
011266  001001                  BNE     .+4
011270  104000          BITB3:  HLT

011272  123453                  CMPB    @(R4)+,@-(R3)
011274  001002                  BNE     CMPB3
011276  123453                  CMPB    @(R4)+,@-(R3)
011300  001401                  BEQ     .+4
011302  104000          CMPB3:  HLT
011304  104000                  SCOPE

                        ;CHECK BINARY OPS USING ADDRESS MODE 6
011306  000402                  BR      .+6             ;RESERVE TWO LOCATIONS
011310  000000          SDATA:  .WORD   0               ;RESERVED FOR SOURCE DATA
011312  000000          DDATA:  .WORD   0               ;RESERVED FOR DESTINATION DATA

011314  013702  001004          MOV     @#FACTOR,R2     ;GET RELOCATION FACTOR AND USE AS AN
011320  010205                  MOV     R2,R5           ;INDEX VALUE TO POINT TO DATA
011322  005065  011312          CLR     DDATA(5)        ;PRESET DESTINATION DATA
011326  012762  000001  011310  MOV     #1,SDATA(2)     ;THIS ROUTSINE PUT A 1 BIT INTO EVERY
011334  056265  011310  011312  1$:     BIS     SDATA(2),DDATA(5)       ;OTHER BIT POSITION IN THE DEST-
011342  006362  011310          ASL     SDATA(2)                ;INATION ADDRESS (52525)
011346  006362  011310          ASL     SDATA(2)
011352  103370                  BCC     1$
011354  022765  052525  011312  CMP     #52525,DDATA(5) ;CHECK RESULT
```

```
011362  001401                          BEQ     .+4
011364  104400                          HLT                     ;ERROR! INCORRECT RESULT
011366  012762  177777  011310          MOV     #-1,SDATA(2)
011374  046562  011312  011310          BIC     DDATA(5),SDATA(2)       ;SOURCE DATA=125252
011402  036265  011310  011312          BIT     SDATA(2),DDATA(5)
011410  001401                          BEQ     .+4
011412  104400                          HLT                     ;ERROR! BIT INST FAILED
011414  006365  011312                  ASL     DDATA(5)                ;DDATA=125252
011420  026265  011310  011312          CMP     SDATA(2),DDATA(5)
011426  001401                          BEQ     .+4

011430  104400                          HLT                     ;ERROR! CMP INST FAILED
011432  000257                          CCC
011434  066265  011310  011312          ADD     SDATA(2),DDATA(5)
011442  103002                          BCC     ADD6
011444  102001                          BVC     ADD6
011446  100001                          BPL     .+4
011450  104400                  ADD6:   HLT

011452  006362  011310                  ASL     SDATA(2)                ;SDATA=52524
011456  166265  011310  011312          SUB     SDATA(2),DDATA(5)
011464  103401                          BCS     SUB6
011466  001401                          BEQ     .+4
011470  104400                  SUB6:   HLT

011472  112700  000377                  MOVB    #377,R0         ;R0=177777 (MOVB %R EXTENDS SIGN)
011476  010062  011310                  MOV     R0,SDATA(2)
011502  012765  177777  011312          MOV     #-1,DDATA(5)
011510  166500  011312                  SUB     DDATA(5),R0
011514  001401                          BEQ     .+4
011516  104400                          HLT
011520  066265  011310  011312  1$:     ADD     SDATA(2),DDATA(5)
011526  006362  011310                  ASL     SDATA(2)
011532  005162  011310                  COM     SDATA(2)
011536  036265  011310  011312          BIT     SDATA(2),DDATA(5)
011544  001401                          BEQ     .+4
011546  104400                          HLT
011550  005162  011310                  COM     SDATA(2)
011554  026265  011310  011312          CMP     SDATA(2),DDATA(5)
011562  001401                          BEQ     .+4
011564  104400                          HLT
011566  026200  011310                  CMP     SDATA(2),R0
011572  001352                          BNE     1$
011574  104000                          SCOPE

                                ;CHECK BIT , BYTE OPS USING ADDRESS MODE 6
                                ;NOTE: SDATAB(2), AND DDATAB(4) REFERENCE EVEN BYTE OF SOURCE & DEST DATA
                                ;AND SDATAB(3), AND DDATAB(5) REFERENCE ODD BYTE OF SOURCE & DEST  ATA

011576  013702  001004                  MOV     @#FACTOR,R2     ;GET INDEX VALUE
011602  010204                          MOV     R2,R4           ;R2 FOR SOURCE EVEN BYTE INDEX, R4 FOR
011604  010403                          MOV     R4,R3           ;DEST ODD BYTE, R3 FOR SOURCE EVEN
011606  005203                          INC     R3              ;AND R5 FOR DEST ODD BYTE
011610  010305                          MOV     R3,R5
011612  000261                          SEC                     ;SET CARRY
011614  012762  125252  011740          MOV     #125252,SDATAB(2)
```

```
011622  112763  177125  011740          MOVB    #^O125,SDATAB(3)        ;SOURCE DATA = 052652
011630  016264  011740  011742          MOV     SDATAB(2),DDATAB(4)
011636  052764  125125  011742          BIS     #125125,DDATAB(4)       ;DEST DATA = 177777
011644  136263  011740  011740          BITB    SDATAB(2),SDATA9(3)
011652  001401                          BEQ     .+4
011654  104400                  BIT86:  HLT

011656  146264  011740  011742          BICB    SDATAB(2),DDATAB(4)
011664  103401                          BCS     .+4
011666  104400                          HLT                     ;ERROR MOV,BIS,BIT;BIC DO NOT AFFECT 'C'
011670  126364  011740  011742          CMPB    SDATAB(3),DDATAB(4)
011676  001401                          BEQ     .+4
011700  104400                          HLT

011702  146365  011740  011742          BICB    SDATAB(3),DDATAB(5)
011710  126265  011740  011742          CMPB    SDATAB(2),DDATAB(5)
011716  001401                          BEQ     .+4
011720  104400                          HLT

011722  136564  011742  011742          BITB    DDATAB(5),DDATAB(4)
011730  001401                          BEQ     .+4
011732  104400                          HLT
011734  104000                          SCOPE

011736  000406                          BR      UB7             ;RESERVE TWO WORDS
011740  000000                  SDATAB: .WORD   0               ;RESERVED FOR SOURCE DATA
011742  000000                  DDATAB: .WORD   0               ;RESERVED FOR DEST DATA

                                        ;CHECK BINARY WORD OPS USING ADDRESS MODE 7
                                        ;R2=ADDRESS OF SOURCE DATA, AND R3= ADDRESS OF DEST DATA
011744  000000                  SBIN7:  .WORD   0               ;CONTAINS ADDRESS OF SOURCE DATA
011746  000000                  DBIN7:  .WORD   0               ;CONTAINS ADDRESS OF DEST DATA
011750  000000                          .WORD   0               ;CONTAINS SOURCE DATA
011752  000000                          .WORD   0               ;CONTAINS DEST DATA

011754  010700                  UB7:    MOV     PC,R0
011756  024040                          CMP     -(R0),-(R0)
011760  010002                          MOV     R0,R2
011762  024242                          CMP     -(R2),-(R2)
011764  010012                          MOV     R0,(R2)
011766  010203                          MOV     R2,R3
011770  024043                          CMP     -(R0),-(R3)
011772  010013                          MOV     R0,(R3)

011774  000261                          SEC
011776  012777  100000  177740          MOV     #100000,@SBIN7  ;SOURCE DATA = 100000
012004  017777  177734  177734          MOV     @SBIN7,@DBIN7   ;DEST DATA = 100000
012012  103001                          BCC     MOV7
012014  100401                          BMI     .+4
012016  104400                  MOV7:   HLT
012020  006377  177722                  ASL     @DBIN7          ;DEST DATA = 000000
012024  102001                          BVC     .+4
012026  001401                          BEQ     .+4
012030  104400                          HLT

012032  027777  177706  177706          CMP     @SBIN7,@DBIN7   ;(R2)=100000,(R3)=000000
```

```
012040  103402                          BCS     CMP7
012042  102401                          BVS     CMP7
012044  100401                          BMI     .+4
012046  104400                  CMP7:   HLT

012050  167777  177670  177670          SUB     @SBIN7,@DBIN7   ;(R2)=100000,(R3)=100000
012056  103003                          BCC     SUB7
012060  102002                          BVC     SUB7
012062  001401                          BEQ     SUB7
012064  100401                          BPL     .+4
012066  104400                  SUB7:   HLT

012070  006277  177650                  ASR     @SBIN7          ;(R2)=140000
012074  067777  177644  177644          ADD     @SBIN7,@DBIN7   ;(R2)=140000,(R3)=040000
012102  103003                          BCC     ADD7
012104  102002                          BVC     ADD7
012106  001401                          BEQ     ADD7
012110  100001                          BPL     .+4
012112  104400                  ADD7:   HLT

012114  047777  177624  177624          BIC     @SBIN7,@DBIN7   ;(R2)=140000,(R3)=000000
012122  001401                          BEQ     .+4
012124  104400                          HLT

012126  057777  177612  177612          BIS     @SBIN7,@DBIN7   ;(R2)=140000,(R3)=140000
012134  100401                          BMI     .+4
012136  104400                          HLT

012140  027777  177600  177600          CMP     @SBIN7,@DBIN7
012146  001401                          BEQ     .+4
012150  104400                          HLT
012152  104000                          SCOPE

                        ;SOME MISCELLANEOUS OPERATION INVOLVING THE PC
                        ;NOTE: NONE OF THESE OPERATIONS SHOULD AFFECT THE PC
012154  005000                          CLR     R0
012156  005067  000072                  CLR     1$
012162  010707                          MOV     PC,PC
012164  120707                          CMPB    PC,PC
012166  030707                          BIT     PC,PC
012170  060007                          ADD     R0,PC
012172  105707                          TSTB    PC
012174  005507                          ADC     PC
012176  021007                          CMP     (R0),PC
012200  131007                          BITB    (R0),PC
012202  062707  000000                  ADD     #0,PC
012206  023707  001004                  CMP     @#FACTOR,PC
012212  133707  001004                  BITB    @#FACTOR,PC
012216  000240                          NOP
                        ;THE NEXT TWO INSTRUCTION CAUSE THE PROGRAM TO JUMP TO THE UNRELOCATED
                        ;CODE AND TO RETURN ON THE FOLLOWING INST (IF THE CODE IS RELOCATED)
012220  163707  001004                  SUB     @#FACTOR,PC     ;JUMPS TO UNRELOCATED CODE
012224  063707  001004                  ADD     @#FACTOR,PC     ;RETURNS
012230  000240                          NOP
012232  024607                          CMP     -(SP),PC
012234  132607                          BITB    (SP)+,PC
```

```
012236  026707  000012                  CMP     1$,PC
012242  166707  000006                  SUB     1$,PC
012246  046707  000002                  BIC     1$,PC
012252  000401                          BR      .+4             ;BRANCH OVER 1$
012254  000000                  1$:     0
012256  104000                          SCOPE

012260  010702                          MOV     PC,R2
012262  062702  000012                  ADD     #12,R2
012266  012707  001132                  MOV     #RELCC,PC       ;GO RELOCATE PROGRAM CODE
012272  000240                          NOP                     ;PROGRAM RETURNS HERE+2
                                ;111111111111 LAST ADDRESS OF CODE TO BE RELOCATED 1111111111


                                ;22222222222? FIRST ADDRESS TO BE RELOCATED 222222222
012274  010700                  REL2:   MOV     PC,R0           ;GET PC
012276  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL2
012300  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
012304  010700                          MOV     PC,R0           ;GET CURRENT PC
012306  162700  012306                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
012312  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
012316  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR
                                ;CHECK BINARY BYTE OPS USING ADDRESS MODE 7
012320  000406                          BR      BINB7           ;RESERVE SPACE FOR ADDRESSES & DATA
012322  000000                  SBINB7: .WORD   0               ;CONTAINS ADDRESS OF SOURCE EVEN BYTE
012324  000000                          .WORD   0               ;CONTAINS ADDRESS OF SOURCE ODD BYTE
012326  000000                          .WORD   0               ;CONTAINS ADDRESS OF DEST EVEN BYTE
012330  000000                          .WORD   0               ;CONTAINS ADDRESS OF DEST ODD BYTE
012332  000000                  DBINB7: .WORD   0               ;CONTAINS SOURCE DATA
012334  000000                          .WORD   0               ;CONTAINS DEST DATA

012336  010700                  BINB7:  MOV     PC,R0
012340  024040                          CMP     -(R0),-(R0)     ;R0 = ADDRESS OF DEST DATA
012342  010060  177772                  MOV     R0,-6(R0)       ;LOAD ADDRES OF DEST EVEN BYTE DATA
012346  010060  177774                  MOV     R0,-4(R0)
012352  005260  177774                  INC     -4(R0)          ;LOAD ADDRESS OF DEST ODD BYTE DATA
```

```
012356  005740                          TST     -(R0)               ;R0=ADDRESS OF SOURCE DATA
012360  010060  177770                  MOV     R0,-10(R0)          ;LOAD ADDRESS OF SOURCE EVEN BYTE DATA
012364  010060  177772                  MOV     R0,-6(R0)
012370  005260  177772                  INC     -6(R0)              ;LOAD ADDRESS OF SOURCE ODD BYTE DATA

012374  005002                          CLR     R2                  ;SET INDEX REGISTERS
012376  012703  000002                  MOV     #2,R3               ;@SBINB7(2);@SBINB7(3) REFERENCE EVEN &
012402  012704  177774                  MOV     #-4,R4              ;ODD BYTE SOURCE DATA; @DBINB7(4);@DBINB7(5)
012406  012705  177776                  MOV     #-2,R5              ;REFERENCE DEST EVEN& ODD BYTE DATA
                .
                .
012412  005020                          CLR     (R0)+               ;PRESET SOURCE DATA
012414  005010                          CLR     (R0)                ;PRESET DEST DATA
012416  013746  001004                  MOV     @#FACTOR,-(SP)      ;GET RELOCATION FACTOR
012422  061602                          ADD     (SP),R2             ;AND ADD TO INDEX VALUES
012424  061603                          ADD     (SP),R3
012426  061604                          ADD     (SP),R4
012430  062605                          ADD     (SP)+,R5

012432  112773  177777  012322          MC 3    #-1,@SBINB7(3)      ;SRC DATA = 177400
012440  132772  000377  012322          BITB    #377,@SBINB7(2)     ;CHECK THAT EVEN BYTE WAS NOT AFFECTED
012446  001401                          BEQ     .+4                 ;BY MOVB INSTRUCTION
012450  104400                          HLT

012452  157374  012322  012332          BISB    @SBINB7(3),@DBINB7(4)
012460  105274  012332                  INCB    @DBINB7(4)          ;CHECK THAT BIS SET ALL BITS
012464  001401                          BEQ     .+4
012466  104400                          HLT

012470  105375  012332                  DECB    @DBINB7(5)          ;DEST DATA = 177400
012474  005274  012332                  INC     @DBINB7(4)          ;DEST DATA = 177401
012500  127375  012322  012332          CMPB    @SBINB7(3),@DBINB7(5)
012506  001401                          BEQ     .+4
012510  104400                          HLT

012512  147375  012322  012332          BICB    @SBINB7(3),@DBINB7(5)
012520  001401                          BEQ     .+4
012522  104400                          HLT

012524  105073  0123??                  CLRB    @SBINB7(3)          ;SRC DATA = 000000
                .                       ;THIS ROUTINE SETS ALL BITS IN THE SOURCE ODD BYTE BY BISING A BIT FROM
                                        ;THE DEST EVEN BYTE INTO THE SOURCE ODD BYTE
012530  157473  012332  012322  BIS7:   BISB    @DBINB7(4),@SBINB7(3)
012536  106174  012332                  ROLB    @DBINB7(4)
012542  103372                          BCC     BIS7
012544  022772  177400  012322          CMP     #177400,@SBINB7(2)      ;CHECK RESULT
012552  001401                          BEQ     .+4
012554  104400                          HLT

012556  000372  012322                  SWAB    @SBINB7(2)          ;SRC DATA = 000377
012562  112775  000200  012332          MOVB    #200,@DBINB7(5)     ;DEST DATA = 100000

012570  147572  012332  012322  BIC7:   BICB    @DBINB7(5),@SBI  7(2)
012576  106075  012332                  RORB    @DBINB7(5)
012602  103372                          BCC     BIC7
012604  005772  012322                  TST     @SBINB7(2)
```

```
012610  001401                          BEQ     .+4
012612  104400                          HLT
012614  104000                          SCOPE

012616  012702  000001          0AERR:  MOV     #1,R2           ;LOAD R2 WITH ODD #
012622  010703                          MOV     PC,R3
012624  000401                          BR      .+4             ;RESERVE SPACE FOR A WORD
012626  000000                          .WORD   0               ;WILL CONTAIN AN ODD ADDRESS
012630  005723                          TST     (R3)+           ;STEP R3 TO POINT TO WORD ABOVE
012632  010313                          MOV     R3,(R3)
012634  005213                          INC     (R3)            ;AND MAKE ODD
012636  012737  012764  000004          MOV     #1$,@#ERRVEC    ;SET ODD ADDRESS & RESERVED INSTRUCTION
012644  063737  001004  000004          ADD     @#FACTOR,@#ERRVEC
012652  013737  000004  000010          MOV     @#ERRVEC,@#RESVEC       ;TO TRAP TO 1$ BELOW

012660  000277                          SCC                     ;SET ALL CC'S
012662  160212                          SUB     R2,(R2)
012664  104400                          HLT
012666  060222                          ADD     R2,(R2)+
012670  104400                          HLT
012672  006342                          ASL     -(R2)
012674  104400                          HLT
012676  106512                          MFPD    (R2)            ;NOTE: MAY BE RESERVED
012700  104400                          HLT
012702  170412                          CLRF    (R2)
012704  104400                          HLT
012706  042202                          BIC     (R2)+,R2
012710  104400                          HLT
012712  164202                          SUB     -(R2),R2
012714  104400                          HLT
012716  155202                          BISB    @-(R2),R2
012720  104400                          HLT
012722  105532                          ADCB    @(R2)+
012724  104400                          HLT
012726  163302                          SUB     @(R3)+,R2
012730  104400                          HLT
012732  005733                          TST     @(R3)+
012734  104400                          HLT
012736  106533                          MFPD    @(R3)+
012740  104400                          HLT
012742  170453                          CLRD    @-(R3)
012744  104400                          HLT
012746  137702  177775                  BITB    @.+1,R2
012752  104400                          HLT
012754  105477  177773                  NEGB    @.-1
012760  104400                          HLT
012762  000406                          BR      2$

012764  062716  000002          1$:     ADD     #2,(SP)         ;ADJUST RETURN PC
012770  052766  000017  000002          BIS     #17,2(SP)       ;SET CONDITION CODES ON RETURN
012776  000002                          RTI

013000  012706  000500          2$:     MOV     #STKPTR,SP      ;RESET STACK PTR
013004  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC
013012  012737  000012  000010          MOV     #RESVEC+2,@#RESVEC
013020  104000                          SCOPE
```

```
                                        ;CHECK JMP INSTRUCTIONS

013022  010700                          MOV     PC,R0
013024  062700  000012                  ADD     #12.R0          ;SET ADDRESS FOR JMP INST
013030  000277                          SCC                     ;SET CC'S
013032  000110                          JMP     (R0)
013034  000402                          BR      .+6
013036  000250                          CLN                     ;JMP INST JUMPS HERE
013040  000775                          BR      .-4

013042  103003                          BCC     JMP1
013044  102002                          BVC     JMP1
013046  001001                          BNE     JMP1
013050  100001                          BPL     .+4
013052  104400                  JMP1:   HLT                     ;ERROR! INCORRECT CC'S AFTER JMP

013054  005002                          CLR     R2              ;SET INDICATOR
013056  010703                          MOV     PC,R3
013060  000401                          BR      .+4             ;RESERVE WORD FOR JMP ADDRESS
013062  000000                          .WORD   0               ;CONTAINS ADDRESS FOR JMP INST
013064  005723                          TST     (R3)+
013066  010313                          MOV     R3,(R3)
013070  010300                          MOV     R3,R0
013072  062713  000022                  ADD     #22,(R3)        ;(R3) IS JMP ADDRESS
013076  010300                          MOV     R3,R0
013100  000133                          JMP     @(R3)+          ;JUMP TO ADDRESS CONTAINED IN R3
013102  000402                          BR      .+6
013104  005102                          COM     R2              ;COMPLEMENT INDICATOR
013106  000775                          BR      .-4
013110  005202                          INC     R2              ;CHECK INDICATOR
013112  001003                          BNE     JMP3
013114  005720                          TST     (R0)+
013116  020003                          CMP     R0,R3           ;CHECK AUTO-INC R3
013120  001401                          BEQ     .+4
013122  104400                  JMP3:   HLT

013124  005002                          CLR     R2              ;SET INDICATOR
013126  010704                          MOV     PC,R4           ;SET UP JMP REGISTER
013130  010400                          MOV     R4,R0           ;SET UP CHECK REGISTER
013132  000402                          BR      1$
013134  005102                          COM     R2              ;COMPLEMENT INDICATOR
013136  000403                          BR      2$
013140  022424                  1$:     CMP     (R4)+,(R4)+
013142  005724                          TST     (R4)+           ;R4=JMP ADDRESS
013144  000144                          JMP     -(R4)           ;USE R4 AS ADDRESS
013146  005202                  2$:     INC     R2              ;CHECK INDICATOR
013150  001003                          BNE     JMP4
013152  022020                          CMP     (R0)+,(R0)+
013154  020004                          CMP     R0,R4           ;CHECK AUTO-DEC R4
013156  001401                          BEQ     .+4
013160  104400                  JMP4:   HLT

013162  010703                          MOV     PC,R3
013164  000401                          BR      .+4             ;RESERVE WORD FOR JMP ADDRESS
013166  000000                  1$:     .WORD   0               ;CONTAINS JUMP ADDRESS
```

```
013170  006723                          TST     (R3)+
013172  010313                          MOV     R3,(R3)
013174  062723  000016                  ADD     #16.,(R3)+
013200  010300                          MOV     R3,R0           ;LOAD CHECK REGISTER
013202  000402                          BR      3$
013204  005102                  2$:     COM     R2
013206  000401                          BR      4$
013210  000153                  3$:     JMP     @-(R3)          ;JUMP TO 2$ VIA 1$ ABOVE
013212  005202                  4$:     INC     R2              ;CHECK INDICATOR
013214  001003                          BNE     JMP5
013216  005740                          TST     -(R0)
013220  020003                          CMP     R0,R3           ;CHECK AUTO-DEC R3
013222  001401                          BEQ     .+4
013224  104400                  JMP5:   HLT

013226  000402                          BR      2$
013230  005102                  1$:     COM     R2              ;COMPLEMENT INDICATOR
013232  000402                          BR      3$
013234  000167  177770          2$:     JMP     1$
013240  005202                  3$:     INC     R2
013242  001401                          BEQ     .+4
013244  104400                  JMP6:   HLT

013246  012767  013264  000020          MOV     #1$,7$          ;SET UP JMP ADDRESS
013254  063767  001004  000012          ADD     @#FACTOR,7$     ;ADD RELOCATION FACTOR
013262  000402                          BR      2$              ;GO TO JMP  @7$ INST
013264  005102                  1$:     COM     R2              ;COMPLEMENT INDICATOR
013266  000403                          BR      3$              ;GO TO CHECK ROUTINE
013270  000177  000000          2$:     JMP     @7$             ;JMP TO 1$ ABOVE VIA 7$
013274  000000                  7$:     .WORD   0               ;CONTAINS JMP ADDRESS
013276  005202                  3$:     INC     R2              ;CHECK INDICATOR
013300  001401                          BEQ     .+4
013302  104400                  JMP7:   HLT
013304  104000                          SCOPE

                                ;CHECK JSR INSTRUCTIONS
013306  013705  001004          JSRTST: MOV     @#FACTOR,R5     ;GET RELOCATION FACTOR
013312  012702  013344                  MOV     #3$,R2          ;FORM DEST ADRS
013316  060502                          ADD     R5,R2           ;ADD RELOCATION FACTOR
013320  000277                          SCC                     ;PRESET CC'S
013322  000242                          CLV
013324  004512                          JSR     R5,(R2)         ;GO TO 3$ VIA R2
013326  005702                  1$:     TST     R2              ;CHECK INDICATOR
013330  001017                          BNE     JSR1            ;R2 SHOULD=0
013332  023705  001004                  CMP     @#FACTOR,R5     ;CHECK THAT RTS  R5 RESTORED R5
013336  001014                          BNE     JSR1
013340  000414                          BR      JSR1A           ;EXIT TO SCOPE
013342  000205                  2$:     RTS     R5              ;RETURN FROM SUBROUTINE
013344  103011                  3$:     BCC     JSR1            ;CHECK THAT JSR DID NOT
013346  102410                          BVS     JSR1
013350  001007                          BNE     JSR1            ;AFFECT CC'S
013352  100006                          BPL     JSR1
013354  005002                          CLR     R2              ;CLEAR INDICATOR
013356  012704  013326                  MOV     #1$,R4          ;GET UNRELOCATED RETURN ADDRESS
013362  061604                          ADD     (SP),R4         ;ADD RELOCATION FACTOR (OLD R5)
013364  020405                          CMP     R4,R5           ;CHECK THAT OLD R5 WAS PLACED ON THE
```

```
013366  001765                          BEQ     2$              ;STACK,& THAT NEW R5 CONTAINS RETURN PC
013370  104400          JSR1:           HLT                     ;ERROR! ABOVE

013372  013704  001004  JSR1A:  MOV     @#FACTOR,R4     ;GET RELOCATION FACTOR
013376  005000                  CLR     R0              ;SET INDICATOR
013400  012705  013420          MOV     #1$,R5
013404  060405                  ADD     R4,R5           ;SET UP JSR DEFERRED ADRS
013406  010502                  MOV     R5,R2
013410  012715  013436          MOV     #5$,(R5)
013414  060415                  ADD     R4,(R5)         ;(R5)=DEST ADRS
013416  000401                  BR      2$              ;RESERVE WORD FOR ADDRESS
013420  000000          1$:     .WORD   0               ;CONTAINS DEST ADRS FOR JSR
013422  004435          2$:     JSR     R4,@(R5)+       ;JSR TO 5$ VIA 1$ ABOVE
013424  005200          3$:     INC     R0              ;CHECK INDICATOR
013426  001013                  BNE     JSR3
013430  000413                  BR      JSR3A
013432  005100          4$:     COM     R0              ;COMPLEMENT INDICATOR
013434  000204                  RTS     4               ;RETURN FROM SUBROUTINE
013436  012703  013424  5$:     MOV     #3$,R3          ;GET UNRELOCATED RETURN ADDRESS
013442  061603                  ADD     (SP),R3         ;ADD RELOCATION FACTOR (OLD R4)
013444  020403                  CMP     R4,R3
013446  001003                  BNE     JSR3
013450  005722                  TST     (R2)+
013452  020205                  CMP     R2,R5           ;CHECK AUTO-INC R5
013454  001766                  BEQ     4$              ;GO TO RTS
013456  104400          JSR3:   HLT                     ;ERROR ABOVE

013460  013704  001004  JSR3A:  MOV     @#FACTOR,R4
013464  010405                  MOV     R4,R5
013466  010703                  MOV     PC,R3
013470  000401                  BR      2$
013472  000405          1$:     BR      4$
013474  022323          2$:     CMP     (R3)+,(R3)+
013476  000277                  SCC
013500  004443                  JSR     R4,-(R3)                ;GO TO 2$
013502  104400          3$:     HLT
013504  000414                  BR      JSR4A
013506  103012          4$:     BCC     JSR4
013510  102011                  BVC     JSR4
013512  001010                  BNE     JSR4
013514  100007                  BPL     JSR4
013516  012702  013502          MOV     #3$,R2          ;GET UNRELOCATED RETURN ADDRESS
013522  061602                  ADD     (SP),R2         ;ADD RELOCATION FACTOR (OLD R4)
013524  020204                  CMP     R2,R4           ;CHECK THAT CALCULATED RETURN
013526  001002                  BNE     JSR4            ;PC = NEW R4
013530  005724                  TST     (R4)+
013532  000204                  RTS     R4
013534  104400          JSR4:   HLT


013536  000401          JSR4A:  BR      2$
013540  000405          1$:     BR      3$
013542  010700          2$:     MOV     PC,R0
013544  004767  177770          JSR     PC,1$
013550  100407                  BMI     JSR6A
013552  104400                  HLT
```

```
013554  022020                  3$:     CMP     (R0)+,(R0)+
013556  020016                          CMP     R0,(SP)         ;CHECK THAT RETURN ADDRESS IS ON THE
013560  001401                          BEQ     .+4             ;STACK
013562  104400                          HLT
013564  000270                          SEN                     ;SET N
013566  000207                          RTS     PC
013570  104000                  JSR6A:  SCOPE

                                ;CHECK IOT TRAP (AND ROLB/ASLB)
013572  012737  013624  000020          MOV     #IOT1,@#IOTVEC
013600  063737  001004  000020          ADD     @#FACTOR,@#IOTVEC       ;ADD RELOCATION FACTOR
013606  000261                          SEC                             ;SET CARRY
013610  013737  177776  000022          MOV     @#PSW,@#IOTVEC+2        ;RETAIN CURRENT PSW ON TRAP
013616  005000                          CLR     R0                      ;PRESET R0
013620  000004                          IOT
013622  000403                          BR      IOT1A
013624  106100                  IOT1:   ROLB    R0                      ;ROTATE R0
013626  102376                          BVC     .-2             ;UNTIL V SETS (R0=200)
013630  000002                          RTI
013632  106300                  IOT1A:  ASLB    R0              ;SHIFT SHOULD SET CARRY
013634  103004                          BCC     IOT1B
013636  102003                          BVC     IOT1B
013640  001002                          BNE     IOT1B
013642  005700                          TST     R0              ;R0 SHOULD =0
013644  001401                          BEQ     .+4
013646  104400                  IOT1B:  HLT                     ;ERROR! ROL/ASL FAILED TO SET CC'S PROPERLY
013650  012737  000022  000020          MOV     #IOTVEC+2,@#IOTVEC      ;RESTORE IOT TRAP
013656  005037  000022                  CLR     @#IOTVEC+2              ;VECTOR
013662  104000                          SCOPE

                                ;CHECK EMT TRAP SEQUENCE
013664  013746  000030                  MOV     @#EMTVEC,-(SP)  ;SAVE SCOPE PTR
013670  012737  013724  000030          MOV     #EMT1,@#EMTVEC  ;SET EMT TRAP VECTOR
013676  063737  001004  000030          ADD     @#FACTOR,@#EMTVEC       ;ADD RELOCATION FACTOR
013704  000262                          SEV                     ;SET V
013706  013737  177776  000032          MOV     @#PSW,@#EMTVEC+2        ;RETAIN CURRENT PSW ON TRAP
013714  000265                          +SEZ!SEC
013716  104000                          EMT                     ;TRAP TO EMT1
013720  001433                          BEQ     EMT1C           ;GO TO EMT1C
013722  104400                          HLT                     ;ERROR! INCORRECT CC'S WERE SET ON RETURN
013724  102027                  EMT1:   BVC     EMT1B           ;'V' SHOULD'VE SET ON EMT TRAP
013726  105100                          COMB    R0              ;R0=000377,CC'S=1001
013730  105500                          ADCB    R0              ;R0=000000,CC'S=0101
013732  106000                          RORB    R0              ;R0=000200,CC'S=1010
013734  102023                          BVC     EMT1B
013736  100022                          BPL     EMT1B
013740  000257                          CCC
013742  105400                          NEGB    R0              ;R0=000200,CC'S=1010
013744  102017                          BVC     EMT1B
013746  100016                          BPL     EMT1B
013750  000242                          CLV                     ;CLEAR 'V'
013752  000261                          SEC                     ;AND SET 'C'
013754  105300                          DECB    R0              ;R0=000177,CC'S=0011
013756  102012                          BVC     EMT1B
013760  100411                          BMI     EMT1B
013762  000242                          CLV                     ;CLEAR 'V'
```

```
013764  105200                          INCB    R0              ;R0=000200,CC'S=1011
013766  103006                          BCC     EMT1B
013770  102005                          BVC     EMT1B
013772  100004                          BPL     EMT1B
013774  000242                          CLV                     ;CLEAR 'V'
013776  106200                          ASRB    R0              ;SHIFT R0 UNTIL 'V' CLEARS
014000  102776                          BVS     .-2
014002  000401                          BR      .+4
014004  104400                  EMT1B:  HLT                     ;ERROR!
014006  000002                          RTI                     ;EXIT WITH R0=000377
014010  105500                  EMT1C:  ADCB    R0              ;R0=000000
014012  103003                          BCC     EMT1D
014014  001002                          BNE     EMT1D
014016  005700                          TST     R0
014020  001401                          BEQ     .+4
014022  104400                  EMT1D:  HLT
014024  012637  000030                  MOV     (SP)+,@#EMTVEC  ;RESTORE SCOPE PTR
014030  005037  000032                  CLR     @#EMTVEC+2
014034  104000                          SCOPE

                                ;CHECK TRAP INSTRUCTION TRAP SEQUENCE
        000004                          HLT=IOT                 ;REDEFINE HLT
014036  013737  000034  000020          MOV     @#TRAPVEC,@#IOTVEC      ;SET IOT (HLT) TRAP VECTOR
014044  012737  014112  000034          MOV     #TRAP1,@#TRAPVEC        ;SET TRAP VECTOR
014052  063737  001004  000034          ADD     @#FACTOR,@#TRAPVEC      ;ADD RELOCATION FACTOR
014060  000270                          SEN                     ;SET N
014062  013737  177776  000036          MOV     @#PSW,@#TRAPVEC+2       ;RETAIN CURRENT PSW ON TRAP
014070  000261                          SEC                             ;SET CARRY
014072  110700                          MOVB    PC,R0
014074  000264                          SEZ                     ;SET Z BIT
014076  104400                          TRAP                    ;TRAP TO TRAP1
014100  103401                  1$:     BCS     .+4
014102  000004                          HLT
014104  001401                          BEQ     .+4
014106  000004                          HLT
014110  000412                          BR      TRAP1C
014112  100401                  TRAP1:  BMI     .+4             ;N BIT GOT SET ON TRAP
014114  000004                          HLT
014116  062700  000004                  ADD     #4,R0
014122  120016                          CMPB    R0,(SP)         ;CHECK LOW BYTE OF RETURN PC ON
014124  001401                          BEQ     .+4             ;STACK
014126  000004                          HLT
014130  124646                          CMPB    -(SP),-(SP)
014132  032626                          BIT     (SP)+,(SP)+
014134  000002                          RTI                     ;RETURN TO INST FOLLOWING TRAP (1$)

014136  013737  000020  000034  TRAP1C: MOV     @#IOTVEC,@#TRAPVEC      ;RESTORE TRAP (HLT) TRAP VECTOR
014144  012737  000340  000036          MOV     #PRTY7,@#TRAPVEC+2
014152  012737  000022  000020          MOV     #IOTVEC+2,@#IOTVEC
014160  005037  000022                  CLR     @#IOTVEC+2
014164  104000                          SCOPE
        104400                          HLT=TRAP                ;RESTORE HLT TO A TRAP INST

014166  010702                          MOV     PC,R2
014170  062702  000012                  ADD     #12,R2
014174  012707  001132                  MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
```

```
014200  000240                          NOP                     ;PROGRAM RETURNS HERE+2
                                ;2222222222222 LAST ADDRESS OF CODE TO BE RELOCATED 22222222222

014202  010701                          MOV     PC,R1           ;SET SCOPE PTR

                                ;THE BELOW ROUTINE ASCERTAINS WHICH CP & CP OPTIONS THE PROGRAM IS RUN-
                                ;NING ON AND SETS AN INDICATOR IN OPT.CP ACCORDINGLY.
014204  005767  164570          CPCHK:  TST     ICNT            ;CHECK IF PASS 0
014210  001036                          BNE     REL3            ;DO NOT EXECUTE ROUTINE IF NOT PASS 0
014212  012737  000002  000006          MOV     #RTI,@#ERRVEC+2 ;SET UP ERROR TRAP TO RETURN
014220  012700  000003                  MOV     #3,R0
014224  000261                          SEC
014226  005737  177772                  TST     @#PIRQ          ;R0=3 IF 11/45
014232  005600                          SBC     R0              ;R0=2 IF 11/40
014234  000261                          SEC
014236  105737  177777                  TSTB    @#PSW+1         ;R0=1 IF 11/2
014242  005600                          SBC     R0
014244  005037  177700                  CLR     @#177700        ;R0=0 IF 11/05
014250  006300                          ASL     R0              ;SHIFT INDICATOR
014252  010027                          MOV     R0,(PC)+        ;SET CP INDICATOR
014254  000000                  OPT.CP: .WORD   0               ;CONTAINS OP .ON & CP INDICATORS
                                        ;EVEN BYTE:     0=11/05,2=11/20,4=11/40,6=11/45
                                        ;ODD BYTE:      200=MEM MGMT,100=EIS, 40=11/45 FLOATING POINT
014256  005037  000006          3$:     CLR     @#ERRVEC+2      ;RESTORE ERROR TRAP TO HALT ON TRAP
014262  005037  000012                  CLR     @#RESVEC+2

014266  126727  177762  000004          CMPB    OPT..CP,#4      ;BRANCH IF 11/05 OR 11/20
014274  002404                          BLT     REL3
014276  004767  164730                  JSR     PC,..PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
014302  016631                          ILLTEST
014304  000000                          HALT



                                ;3333333333333 FIRST ADDRESS TO BE RELOCATED 333333333
014306  010700                  REL3:   MOV     PC,R0           ;GET PC
014310  005740                          TST     -(R0)           ;R0 CONTAINS THE ADDRESS OF REL3
014312  010037  001010                  MOV     R0,@#FRSTAD     ;SAVE
014316  010700                          MOV     PC,R0           ;GET CURRENT PC
014320  162700  014320                  SUB     #.,R0           ;SUBTRACT RELOCATION FACTOR
014324  010037  001004                  MOV     R0,@#FACTOR     ;SAVE RELOCATION FACTOR
014330  010701                          MOV     PC,R1           ;SET NEW SCOPE PTR

                                ;CHECK STACK OVERFLOW
014332  013767  177776  000306  OVFLW:  MOV     @#PSW,7$        ;SAVE STATUS IN 7$ BELOW
014340  005037  177776                  CLR     @#PSW           ;SET KERNEL MODE
014344  010746                          MOV     PC,-(SP)        ;PUSH CURRENT PC ONTO STACK
014346  062716  000136                  ADD     #2$-.,(SP)      ;FORM ADDRESS OF 2$ BELOW
014352  011637  000004                  MOV     (SP),@#ERRVEC   ;SET ERROR VECTOR
014356  012737  000340  000006          MOV     #340,@#ERRVEC+2 ;SET PRIORITY LEVEL 7 ON TRAP
014364  062716  000074                  ADD     #4$-2$,(SP)     ;FORM ADDRESS OF 4$ BELOW
014370  012637  000020                  MOV     (SP)+,@#IOTVEC  ;SET IOT TRAP VECTOR TO 4$
014374  012746  000340                  MOV     #340,-(SP)
014400  011637  000022                  MOV     (SP),@#IOTVEC+2 ;SET PRIORITY LEVEL 7 ON IOT TRAP
014404  010746                          MOV     PC,-(SP)        ;PUSH CURRENT PC ONTO THE STAK
014406  062716  000006                  ADD     #6,(SP)         ;ADD OFFSET TO INST FOLLOWING RTI
```

```
014412  000002                          RTI                     ;SET PRIORITY LEVEL 7,CLEAR 'T' BIT
                                                                ;AND EXECUTE FOLLOWING INST NEXT
014414  012703  000376                  MOV     #376,R3
014420  010313                          MOV     R3,(R3)         ;LOAD 376 INTO ADDRESS 376
014422  010306                          MOV     R3,SP           ;SET STACK PTR AT BOUNDARY

                                ;THE BELOW INSTRUCTIONS SHOULD NOT CAUSE AN OVERFLOW TRAP
014424  005716                          TST     (SP)            ;BECAUSE TST IS A NON MODIFYING INST
014426  021666  177776                  CMP     (SP),-2(SP)     ;SO IS COMPARE
014432  122737  000002  014254          CMPB    #2,@#OPT.CP     ;CHECK IF 11/20 OR 11/05
014440  002411                          BLT     12$             ;BRANCH IF 11/40 OR 11/45
014442  001404                          BEQ     11$             ;BRANCH IF 11/20
014444  012767  000014  000144          MOV     #14,51$         ;CHANGE CHECK WORD IN 51$ IF 11/05
014452  000407                          BR      10$
014454  012767  000034  000134  11$:    MOV     #34,51$         ;CHANGE CHECK WORD IN 51$ IF 11/20
014462  000403                          BR      10$
014464  012656                  12$:    MOV     (SP)+,@-(SP)    ;BECAUSE OF ADDRESS MODE 5
014466  054676  000000                  BIS     -(SP),@(SP)     ;BECAUSE OF ADDRESS MODE 7
014472  005066  000004          10$:    CLR     4(SP)           ;BECAUSE DEST ADDRESS IS > 376
014476  057636  000000                  BIS     @(SP),@(SP)+    ;BECAUSE OF ADDRESS MODE 3
014502  000406                          BR      3$              ;BRANCH OVER NON KERNEL MODE TESTS


                                ;ERROR SERVICE ROUTINE
014504  012600                  2$:     MOV     (SP)+,R0        ;SAVE PC OF INSTRUCTION THAT TRAPPED
014506  012602                          MOV     (SP)+,R2        ;SAVE PSW
014510  012706  000500                  MOV     #STKPTR,SP      ;SET STACK PTR
014514  104400                          HLT                     ;ERROR! AN INSTRUCTION THAT WAS NOT
                                                                ;SUPPESED TO TRAP TRAPPED
                                                                ;R0 CONTAINS PC, R2 CONTAINS PSW
014516  000450                          BR      6$              ;EXIT TEST
                                ;THE BELOW INSTRUCTIONS WILL CAUSE A STACK OVERFLOW
                                ;STACK PTR IS AT 376
014520  062737  000066  000004  3$:     ADD     #4$-2$,@#ERRVEC ;SET ERROR VECTOR TO 4$
014526  010306                          MOV     R3,SP           ;SET STACK PTR AT 376
014530  012702  000001                  MOV     #1,R2
014534  005000                          CLR     R0
014536  005016                          CLR     (SP)            ;SETS BIT 0 IN R0
014540  006302                          ASL     R2              ;SHIFT INDICATOR BIT
014542  105226                          INCB    (SP)+           ;SETS BIT 1 IN R0
014544  006302                          ASL     R2
014546  060746                          ADD     PC,-(SP)        ;SETS BIT 2 IN R0
014550  006302                          ASL     R2
014552  000004                          IOT                     ;SETS BIT 3 IN R0
014554  006302                          ASL     R2
014556  004767  000014                  JSR     PC,40$          ;SETS BIT 4 IN R0
014562  006302                          ASL     R2              ;NOTE: 11/05 WITHOUT ECO # KD11A-00005
                                                                ;DOES NOT SET BIT 4.
014564  050666  177776                  BIS     SP,-2(SP)       ;SETS BIT 5 IN R0
014570  000407                          BR      5$

                                ;PROGRAM WILL TRAP HERE ON OVERFLOW TRAP
014572  050200                  4$:     BIS     R2,R0           ;SET APPROPRIATE BIT IN R0
014574  000002                          RTI                     ;RETURN FROM TRAP

014576  000207                  40$:    RTS     PC
```

```
014600  012737  000022  000020  41$:    MOV     #IOTVEC+2,@#IOTVEC
014606  000002                          RTI

                                ;CHECK THAT ABOVE INSRUCTIONS DID TRAP
014610  012706  000500          5$:     MOV     #STKPTR,SP          ;SET STACK PTR
014614  022700                  50$:    CMP     (PC)+,R0            ;EACH INSTRUCTION SET A BIT IN R0
014620  000000                  51$:    .WORD   0                   ;CONTAINS CHECK WORD
014622  001407                          BEQ     6$                  ;R0= 77 IF 40 OR 45,14 IF 05,34 IF 20
014624  105737  014254                  TSTB    @#OPT.CP            ;CHECK IF 11/05
014630  001003                          BNE     52$                 ;BRANCH IF NOT AN 11/05
014632  022700  000034                  CMP     #34,R0              ;USE ECO KD11A-00005 CHECK WORD
014636  001401                          BEQ     6$
014640  104400                  52$:    HLT

                                ;EXIT ROUTINE
014642  012706  000600          6$:     MOV     #KPTR,SP            ;SET KERNEL STACK PTR
014646  012746                          MOV     (PC)+,-(SP)         ;PUSH OLD PSW ONTO STACK
014650  000000                  7$:     .WORD   0                   ;CONTAINS SAVED PSW
014652  010746                          MOV     PC,-(SP)            ;PUSH CURRENT PC ONTO STACK
014654  062716  000006                  ADD     #6,(SP)             ;ADD OFFSET
014660  000002                          RTI
014662  012706  000500                  MOV     #STKPTR,SP          ;SET STACK PTR
014666  012737  000006  000004          MOV     #ERRVEC+2,@#ERRVEC
014674  104000                          SCOPE

                                ;CHECK THAT ALL RESERVED INSTRUCTIONS TRAP (TO LOCATION 10)
014676  012737  000002  001114  RESTRP: MOV     #2,@#SCOPED         ;LIMIT TO TWO ITERATIONS
014704  010701                          MOV     PC,R1               ;SET SCOPE POINTER
014706  012702  015024                  MOV     #5$,R2              ;GET ADDRESS OR RESERVED INSTRUCTION TABLE
014712  063702  001004                  ADD     @#FACTOR,R2
014716  122737  000004  014254          CMPB    #4,@#OPT.CP         ;ADJUST TABLE ADDRESS IF 11/20, 11/05
014724  003402                          BLE     11$                 ;5$=11/45, 11/40 TABLE, 6$=11/05
014726  062702  000036                  ADD     #6$-5$,R2           ;11/20 TABLE
014732  132737  000040  014255  11$:    BITB    #40,@#OPT.CP+1      ;CHECK IF 11/45 FLOATING POINT IS AVAIL.
014740  001402                          BEQ     .+6                 ;BRANCH IF NOT AVAILABLE
014742  005067  000110                  CLR     50$                 ;SET TABLE TERMINATOR AT GROUP 7
014746  012737  015002  000010          MOV     #4$,@#RESVEC        ;SET RESERVED INSTRUCTION TRAP
014754  063737  001004  000010          ADD     @#FACTOR,@#RESVEC
014762  012203                  1$:     MOV     (R2)+,R3            ;GET FIRST RESERVED INSTRUCTION
014764  001454                          BEQ     7$                  ;0 TERMINATES THE TABLE
014766  012204                          MOV     (R2)+,R4            ;GET LAST RESERVED INSTRUCTION IN GROUP
014770  010317                  2$:     MOV     R3,(PC)             ;EXECUTE RESERVED INSTRUCTION
014772  000000                  3$:     .WORD   0                   ;CONTAINS RESERVED INSTRUCTION
014774  104400                          HLT                         ;ERROR! INSTRUCTION IN R3
014776  104400                          HLT                         ;(2$) ABOVE FAILED TO CAUSE A
015000  104400                          HLT                         ;RESERVED INSTRUCTION TRAP
015002  000405                          BR      41$
015004  012716  015014          4$:     MOV     #41$,(SP)           ;ADJUST RETURN PC
015010  063716  001004                  ADD     @#FACTOR,(SP)       ;TO RETURN TO 41$
015014  000002                          RTI                         ;RETURN TO 41$
015016  020304                  41$:    CMP     R3,R4               ;HAS GROUP OF RESERVED INSTRUCTIONS
015020  001760                          BEQ     1$                  ;BEEN EXECUTED
015022  005203                          INC     R3                  ;INCREMENT THIS RESERVED INSTRUCTION
015024  000761                          BR      2$                  ;TO NEXT ONE AND EXECUTE
                                ;TABLE OF 11/40,11/45 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)
```

```
015024  000007                  5$:     7                       ;GROUP 1
015026  000077                          77                      ;  "
015030  000210                          210                     ;GROUP 2
015032  000227                          227                     ;  "
015034  007000                          7000                    ;GROUP 3
015036  007777                          7777                    ;  "
015040  075040                          75040                   ;GROUP 4
015042  076777                          76777                   ;  "
015044  106400                          106400                  ;GROUP 5
015046  106477                          106477                  ;  "
015050  106700                          106700                  ;GROUP 6
015052  107777                          107777                  ;  "
015054  170000                  50$:    170000                  ;GROUP 7        FLOATING POINT
015056  177777                          177777                  ;               INSTRUCTIONS
015060  000000                          0                       ;0 TERMINATES THE TABLE

                                ;TABLE OF 11/05, 11/20 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)
015062  000006                  6$:     6                       ;GROUP 1
015064  000077                          77                      ;  "
015066  000210                          210                     ;GROUP 2
015070  000237                          237                     ;  "
015072  006400                          6400                    ;GROUP 3
015074  007777                          7777                    ;  "
015076  070000                          70000                   ;GROUP 4
015100  077777                          77777                   ;  "
015102  106400                          106400                  ;GROUP 5
015104  107777                          107777                  ;  "
015106  170000                          170000                  ;GROUP 6
015110  177777                          177777                  ;  "
015112  000000                          0                       ;0 TERMINATES THE TABLE
015114  012737  000012  000010  7$:     MOV     #RESVEC+2,@#RESVEC      ;RESTORE RESERVED TRAP TO HALT AT 12
015122  104000                          SCOPE

                                ;CHECK THAT ALL BITS IN THE PROCESSER STATUS WORD (PSW) CAN BE SET AND
                                ;CLEARED.
015124  013767  177776  000152  PSWCHK: MOV     @#PSW,3$                ;SAVE STATUS
015132  005037  177776                  CLR     @#PSW                   ;CLEAR MODE BITS IN PSW
015136  005046                          CLR     -(SP)                   ;ROUTINE TO CLEAR
015140  010746                          MOV     PC,-(SP)                ;STATUS WORD (PSW)
015142  062716  000006                  ADD     #6,(SP)
015146  000002                          RTI                             ;CLEAR PSW & EXECUTE FOLLOWING INST

015150  013746  000016                  MOV     @#TBITVEC+2,-(SP)
015154  012704  177776                  MOV     #PSW,R4                 ;LOAD ADDRESS OF PSW INTO R4
015160  000250                          CLN
015162  005714                          TST     (R4)                    ;CHECK THAT PSW WAS CLEARED
015164  001401                          BEQ     .+4
015166  104400                          HLT                             ;ERROR! PSW FAILED TO CLEAR
015170  113700  014254                  MOVB    @#OPT.CP,R0             ;GET CP TYPE
015174  016000  016570                  MOV     PSWBIT(0),R0            ;GET BIT MASK FOR TEST R0=THE   BITS IN
                                                                        ;THE PSW WHICH CAN BE SET/CLE.  D.
015200  005737  014254                  TST     @#OPT.CP                ;CHECK IF MEM MGMT IS AVAILABLE
015204  100002                          BPL     10$                     ;BRANCH IF NOT AVAILABLE
015206  052700  170000                  BIS     #170000,R0              ;SET BITS 15-12 IF MEM MGMT
015212  012702  000001          10$:    MOV     #1,R2                   ;R2 = TEST BIT
015216  030200                  1$:     BIT     R2,R0                   ;CHECK IF BIT CAN BE SET CLEARED
```

```
015220  001423                          BEQ     2$
015222  005037  000016                  CLR     @#TBITVEC+2
015226  030227  000020                  BIT     R2,#20          ;CHECK IF TEST WILL SET 'T' BIT
015232  001403                          BEQ     20$
015234  012737  000002  000016          MOV     #RTI,@#   TVEC+2;SET RTI INTO RETURN
015242  005014                  20$:    CLR     (R4)            ;CLEAR PSW
015244  050214                          BIS     R2,(R4)         ;SET R2 INTO PSW
015246  011403                          MOV     (R4),R3         ;GET BIT
015250  020203                          CMP     R2,R3           ;CHECK THAT BIT WAS SET IN PSW
015252  001401                          BEQ     .+4
015254  104400                          HLT                     ;ERROR! BIT IN R2 FAILED TO SET IN PSW
015256  000244                          CLZ                     ;CLEAR Z BIT
015260  040214                          BIC     R2,(R4)         ;CLEAR BIT IN PSW
015262  011403                          MOV     (R4),R3         ;GET PSW RESULT
015264  001401                          BEQ     2$              ;BRANCH IF BIC ABOVE CLEARED BIT IN PSW
015266  104400                          HLT                     ;ERROR! BIT IN R2 FAILED TO CLEAR IN PSW
015270  006302                  2$:     ASL     R2              ;SHIFT TEST BIT
015272  103351                          BCC     1$              ;BRANCH IF ALL BITS NOT TESTED
015274  005014                          CLR     (R4)            ;CLEAR STATUS
015276  012637  0000'S                  MOV     (SP)+,@#TBITVEC+2       ;RESTORE T BIT RETURN
015302  012746                          MOV     (PC)+,-(SP)     ;PUSH ORIGINAL STATUS ON STACK
015304  000000                  3$:     .WORD   0               ;CONTAINS ORIGINAL PSW
015306  010746                          MOV     PC,-(SP)        ;SET RETURN PC
015310  062716  000006                  ADD     #6,(SP)
015314  000002                          RTI                     ;RETURN
015316  104000                  4$:     SCOPE

015320  013704  177776                  MOV     @#PSW,R4                ;SAVE PSW IN R4
015324  010446                          MOV     R4,-(SP)                ;PUSH R4 ONTO STACK
015326  112716  000300                  MOVB    #300,(SP)               ;SET PRIORITY LEVEL 6 AND
015332  010746                          MOV     PC,-(SP)                ;CLEAR 'T' BIT AND EXECUTE
015334  062716  000006                  ADD     #6,(SP)                 ;INSTRUCTION FOLLOWING RTI
015340  000002                          RTI

                                ;CHECK THAT ALL BITS IN THE CURRENT STACK PTR CAN BE SET/CLEARED
015342  010603                  CHKSP:  MOV     SP,R3           ;SAVE STACK PTR
015344  000257                          CCC
015346  112706  000377                  MOVB    #377,SP         ;SET STACK PTR = -1
015352  006006                  1$:     ROR     SP              ;ROTATE 0 BIT THROUGH ALL BIT
015354  103776                          BCS     1$              ;BIT POSITIONS
015356  005206                          INC     SP              ;SHOULD INCREMENT SP TO 0
015360  001403                          BEQ     2$
015362  010602                          MOV     SP,R2           ;SAVE ERROR STACK PTR
015364  010306                          MOV     R3,SP           ;SET STACK PTR FOR TRAP
015366  104400                          HLT                     ;ERROR!

015370  010306                  2$:     MOV     R3,SP           ;RESTORE ORIGINAL STACK PTR

                                ;CHECK BYTE OPERATIONS USING THE STACK
015372  010600                  SPCHK:  MOV     SP,R0           ;SAVE STACK PTR
015374  010003                          MOV     R0,R3
015376  005043                          CLR     -(R3)
015400  112746  177777                  MOVB    #-1,-(SP)       ;(SP) = 377
015404  022713  000377                  CMP     #377,(R3)       ;CHECK THAT ONLY EVE  BYTE WAS AFFECTED
015410  001002                          BNE     1$
015412  020306                          CMP     R3,SP           ;CHECK AUTO-DEC
```

```
015414  001401                          BEQ     .+4
015416  104400                  1$:     HLT

015420  105226                          INCB    (SP)+
015422  005723                          TST     -(R3)+          ;CHECK RESULT
015424  001002                          BNE     2$
015426  020006                          CMP     R0,SP           ;CHECK AUTO-INC
015430  001401                          BEQ     .+4
015432  104400                  2$:     HLT

015434  005143                          COM     -(R3)           ;(R3)=177777
015436  144613                          BICB    -(SP),(R3)
015440  022713  177400                  CMP     #177400,(R3)    ;CHECK RESULT
015444  001002                          BNE     3$
015446  020603                          CMP     SP,R3
015450  001401                          BEQ     .+4
015452  104400                  3$:     HLT

015454  132627  000377                  BITB    (SP)+,#377
015460  001002                          BNE     4$
015462  020600                          CMP     SP,R0
015464  001401                          BEQ     .+4
015466  104400                  4$:     HLT

015470  012746  000001                  MOV     #1,-(SP)
015474  062706  000002                  ADD     #2,SP
015500  012702  177401                  MOV     #177401,R2
015504  120246                          CMPB    R2,-(SP)
015506  001004                          BNE     5$
015510  122602                          CMPB    (SP)+,R2
015512  001002                          BNE     5$
015514  020006                          CMP     R0,SP
015516  001401                          BEQ     .+4
015520  104400                  5$:     HLT
015522  010446                          MOV     R4,-(SP)        ;RESTORE ORIGINAL PSW TO STACK
015524  010746                          MOV     PC,-(SP)
015526  062716  000006                  ADD     #6,(SP)
015532  000002                          RTI
015534  104000                          SCOPE

                                ;CHECK THAT 'C' BIT SETS/CLEARS PROPERLY
015536  012727  177776          CBIT:   MOV     #177776,(PC)+   ;LOAD CONSTANT
015542  000000                  1$:     .WORD   0
015544  010700                          MOV     PC,R0           ;GET CURRENT PC
015546  162700  000004                  SUB     #4,R0           ;POINT R0 TO 1$ ABOVE
015552  005520                  2$:     ADC     (R0)+           ;ADD 'C' BIT TO 1$ ABOVE
015554  006340                          ASL     -(R0)           ;SHIFT 1$
015556  102375                          BVC     2$              ;UNTIL 'V' BIT SETS
015560  022767  077776  177754          CMP     #077776,1$      ;CHECK RESULT
015566  001401                          BEQ     .+4
015570  104400                          HLT                     ;ERROR! INCORRECT RESULT IN 1$ ABOVE
                                                                ;R0=ADDRESS OF DATA

                                ;CHECK THAT CONDITION CODES ARE SET PROPERLY WHEN A NUMBER (CURRENT PC)
                                ;AND THAT NUMBER +1 ARE COMPARED, AND VICE VERSA.
015572  010700                  CMPN:   MOV     PC,R0           ;GET CURRENT PC
```

```
015574  010002                          MOV     R0,R2           ;SAVE IN R2
015576  005202                          INC     R2              ;MAKE R2 = R0+1
015600  000277                          SCC
015602  000251                          +CLC!CLN                ;CLEAR C & N BITS
015604  020002                          CMP     R0,R2           ;COMPARE # WITH #+1
015606  103003                          BCC     1$              ;CARRY BIT SHOULD SET
015610  102402                          BVS     1$              ;V BIT SHOULD CLEAR
015612  001401                          BEQ     1$              ;Z BIT SHOULD CLEAR
015614  100401                          BMI     .+4             ;N BIT SHOULD SET
015616  104400                  1$:     HLT                     ;ERROR! COMPARE # WITH #+1 FAILED TO
                                                                ;SET CONDITION CODES IN PSW CORRECTLY

015620  000277                          SCC                     ;SET CONDITION CODES IN PSW
015622  120200                          CMPB    R2,R0           ;COMPARE #+1 WITH #
015624  103403                          BCS     2$              ;C BIT SHOULD CLEAR
015626  102402                          BVS     2$              ;V BIT SHOULD CLEAR
015630  001401                          BEQ     2$              ;Z BIT SHOULD CLEAR
015632  100001                          BPL     .+4             ;N BIT SHOLD CLEAR
015634  104400                  2$:     HLT                     ;ERROR! COMPARE #+1 WITH # FAILED TO SET
                                                                ;CONDITION CODES IN PSW CORRECTLY

                                ;24 NOP (240) INSTRUCTIONS FOLLOW. THESE NOPS MAY
                                ;BE CHANGED TO TEST CODE IF THE NEED ARISES. THE TEST CODE SHOULD
                                ;BE POSITION INDEPENDENT AND SHOULD RUN WHEN RELOCATED BY THE PROGRAM.
015636  000240                          NOP
015640  000240                          NOP
015642  000240                          NOP
015644  000240                          NOP
015646  000240                          NOP
015650  000240                          NOP
015652  000240                          NOP
015654  000240                          NOP
015656  000240                          NOP
015660  000240                          NOP
015662  000240                          NOP
015664  000240                          NOP
015666  000240                          NOP
015670  000240                          NOP
015672  000240                          NOP
015674  000240                          NOP
015676  000240                          NOP
015700  000240                          NOP
015702  000240                          NOP
015704  000240                          NOP
015706  000240                          NOP
015710  000240                          NOP
015712  000240                          NOP
015714  000240                          NOP
015716  104000                          SCOPE

015720  010702                          MOV     PC,R2
015722  062702  000012                    ADD     #12,R2
015726  012707  001132                    MOV     #RELOC,PC       ;GO RELOCATE PROGRAM CODE
015732  000240                          NOP                     ;PROGRAM RETURNS HERE+2
                                ;3333333333333 LAST ADDRESS OF CODE TO BE RELOCATED 33333333333
```

```
                                        ;CHECK TTY INTERRUPT.
015734  005037  001004          TTYCHK: CLR     @#FACTOR
015740  010701                          MOV     PC,R1
015742  032737  000100  177564          BIT     #100,@#TPS      ;CHECK IF TTY IS READY
015750  001374                          BNE     .-6
015752  012737  016026  000064          MOV     #3$,@#TPVEC     ;SET TTY INTERRUPT VECTOR
015760  012737  000200  000066          MOV     #200,@#TPVEC+2  ;PRIORITY LEVEL 4 ON INTERRUPT
015766  012767  016064  000064          MOV     #NULLS,MSG      ;ADDRESS OF MESSAGE TO BE TYPPED
015774  117737  000050  177566          MOVB    @MSG,@#TPB      ;TYPE FIRST CHARACTER OF MESSAGE
016002  105737  177564                  TSTB    @#TPS
016006  100375                          BPL     .-4
016010  006237  177564                  ASR     @#TPS           ;SET IE BIT IN TTY CSR REG
016014  000001                          WAIT                    ;WAIT FOR FIRST INTERRUPT
016016  000424                          BR      KW11
016020  006337  177564          2$:     ASL     @#TPS           ;CLEAR IE BIT
016024  000002                          RTI

016026  122737  000012  000024  3$:     CMPB    #12,@MSG        ;BRANCH IF CHAR IS NOT <LF>
016034  001004                          BNE     4$
016036  004767  163170                  JSR     PC,.PRINT       ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
016042  001744                          $CRLF
016044  000404                          BR      5$
016046  117737  000006  177566  4$:     MOVB    @MSG,@#TPB      ;TYPE CHARACTER
016054  001761                          BEQ     2$              ;BRANCH IF TERMINATOR
016056  005227                  5$:     INC     (PC)+           ;SET MSG TO NEXT CHAR ADDRESS
016060  000000                  MSG:    .WORD   0               ;CONTAINS ADDRESS OF CHAR TO BE TYPED
016062  000002                          RTI                     ;RETURN
016064  020015  000015          NULLS:  .ASCIZ  <15><40><15>
                                        .EVEN

                                        ;ROUTINE TO TURN ON KW11-L LINE CLOCK IF AVAILABLE
016070  012737  000002  000006  KW11:   MOV     #RTI,@#ERRVEC+2 ;SET UP DIRECT RTI ON TRAP
016076  012737  016234  000100          MOV     #4$,@#LKVEC     ;LOAD INTERRUPT VECTOR
016104  012737  000300  000102          MOV     #300,@#LKVEC+2  ;SET PRIORITY LEVEL 6 ON INT.
016112  000262                          SEV                     ;SET TIME OUT INDICATOR
016114  052737  000100  177546          BIS     #100,@#LKS      ;SET INTERRUPT ENABLE
016122  102447                          BVS     5$              ;SKIP PRIORITY ARBITRATION TEST
                                                                ;BELOW IF NO KW11-L

                                        ;ROUTINE TO CHECK PRIORITY ARBITRATION LOGIC
                                        ;THE BELOW TEST WILL INHIBIT INTERRUPTS ON LEVEL 6 AND ABOVE (LOCKING
                                        ;OUT THE LINE CLOCK) AND THEN SET UP THE TTY TO INTERRUPT. NEXT THE
                                        ;PRIORITY LEVEL WILL BE SET TO 0 ALLOWING INTERRUPTS IN WHICH CASE
                                        ;THE LINE CLOCK (AT LEVEL 6) SHOULD INTERRUPT BEFORE THE TTY (AT LEVEL 4).

016124  132737  000020  177776          BITB    #20,@#PSW       ;CHECK IF 'T' BIT IS SET
016132  001043                          BNE     5$              ;DO NOT DO TEST IF SET
016134  112737  000300  177776          MOVB    #300,@#PSW      ;SET PRIORITY LEVEL = 6
016142  013727  000064                  MOV     @#TPVEC,(PC)+   ;SAVE TTY INTERRUPT VECTOR
016146  000000                  1$:     .WORD   0               ;CONTAINS CURRENT TTY VECTOR
016150  105737  177564                  TSTB    @#TPS           ;CHECK IF READY
016154  100375                          BPL     .-4             ;WAIT FOR TTY TO BECOME READY
016156  012737  016204  000064          MOV     #2$,@#TPVEC     ;SET NEW VECTOR
016164  005227                  6$:     INC     (PC)+           ;STALL WAITING FOR LINE CLOCK
016166  000000                          .WORD   0               ;TO BE READY
```

```
016170  001375                          BNE     6$
016172  012737  016210  000100          MOV     #3$,@#LKVEC             ;SET LI      CK VECTOR
016200  105037  177776                  CLRB    @#PSW                   ;SET PR  R   / LEVEL 0
016204  104400                  2$:     HLT                             ;ERROR! -I   ER TTY INTERRUPTED
                                ;BEFORE THE LINE CLOCK OR BOTH FAILED TO INTERRUPT
016206  000415                          BR      5$                      ;EXIT TEST
016210  016737  177732  000064  3$:     MOV     1$,@#TPVEC              ;RESTORE TTY VECTOR
016216  012737  016234  000100          MOV     #4$,@#LKVEC             ;SET LINE CLOCK VECTOR
016224  105037  177776                  CLRB    @#PSW                   ;RESTORE PRIORITY LEVEL 0
016230  012716  016242                  MOV     #5$,(SP)                ;SET RETURN ADDRESS TO 5$ BELOW

016234  005267  162536          4$:     INC     TICKS                   ;INCREMENT TICK COUNT
016240  000002                          RTI                             ;RETURN

016242  005037  000006          5$:     CLR     @#ERRVEC+2              ;RESTORE ERROR TRAP TO HALT AT 6

016246  000240                  END:    NOP
016250  005037  177776          END1:   CLR     @#PSW           ;CLEAR MODE BITS IN PSW
016254  005046                          CLR     -(SP)           ;CLEAR PSW
016256  012746  016264                  MOV     #.+6,-(SP)
016262  000002                          RTI                     ;GO TO NEXT INST WITH PSW=0
016264  012706  000600                  MOV     #KPTR,SP        ;SET KERNEL STACK PTR (NOT APPLICABLE
                                                                ;FOR 11/20,11/05 CP'S)
016270  032737  000100  177564          BIT     #100,@#TPS      ;CHECK IF OUTPUT DEVICE IS BUSY
016276  001374                          BNE     .-6             ;IS AVAILABLE
016300  105737  177570                  TSTB    @#SWR           ;DELETE END OF PASS TYPE OUT IF SW7=0
016304  100020                          BPL     1$              ;BRANCH IF SW7 IS DOWN
016306  016702  162466                  MOV     ICNT,R2         ;GET PASS COUNT
016312  004767  163012                  JSR     PC,$FORMO       ;GO TO FORMAT ROUTINE
016316  012702  001664                  MOV     #DIGITS+2,R2    ;GET ASCII VALUES
016322  012703  001702                  MOV     #PASSES,R3      ;AND MOVE THEM INTO MESSAGE
016326  012223                          MOV     (R2)+,(R3)+
016330  012223                          MOV     (R2)+,(R3)+
016332  012737  001672  016060          MOV     #PASCNT,@#MSG   ;PASS MESSAGE ADRS TO TELETYPE SERVICE
016340  052737  000100  177564          BIS     #100,@#TPS      ;SET IE BIT
016346  012737  000610  000024  1$:     MOV     #PDWN,@#PFVEC   ;ENABLE POWER FAIL TRAP
016354  012737  000340  000026          MOV     #340,@#PFVEC+2  ;PRIORITY 7 ON POWER FAIL
016362  005267  162412                  INC     ICNT
016366  116700  175662                  MOVB    OPT.CP,R0       ;GET CP TYPE
016372  026067  016574  162400          CMP     PASTAB(R0),ICNT ;CHECK IF END OF TEST
016400  001002                          BNE     2$              ;BRANCH IF NOT AT END
016402  000167  000060                  JMP     DONE
016406  016702  162366          2$:     MOV     ICNT,R2         ;GET PASS COUNT
016412  006302                          ASL     R2
016414  046002  016564                  BIC     CPPASS(0),R2    ;LIMIT PASS COUNT TO 0-6
016420  005037  000016                  CLR     @#16            ;CLEAR T BIT TRAP ADDRESS
016424  012737  000040  001122          MOV     #40,@#SCOPEF+2  ;SET ITERATION COUNT = 40
016432  016216  016560                  MOV     PSWTAB(2),(SP)  ;PUSH NEXT PASS PSW ON STACK
016436  032716  000020                  BIT     #20,(SP)        ;WILL 'T' BIT BE SET ON NEXT PASS?
016442  001406                          BEQ     3$              ;BRANCH IF NOT
016444  012737  000002  001122          MOV     #2,@#SCOPEF+2   ;SET ITERATION COUNT = 2 FOR 'T' BIT
016452  016737  000006  000016          MOV     RTI1,@#16       ;SET 'T' BIT TRAP TO RETURN VIA 16
016460  012746  002230          3$:     MOV     #START2,-(SP)   ;RESART PROGRAM AT START2
016464  000002                  RTI1:   RTI                     ;RESTART PROGRAM AT START2 WITH NEW PSW
                                                                ;(FROM TABLE BELOW) NOTE: THE RTI IS
                                                                ;CHANGED TO AN RTT IF NOT AN 11 05,11/20
```

```
                                        ;ROUTINE TO SET UP MEMORY MANAGEMENT TO RELOCATE PROGRAM CODE ABOVE 28K

016466  032737  000200  177564  DONE:   BIT     #200,@#TPS      ;WAIT FOR TTY OUTPUT TO FINISH
016474  001374                          BNE     DONE
016476  105737  177564                  TSTB    @#TPS           ;WAIT FOR LAST CHARACTER TO BE PRINTED
016502  100375                          BPL     .-4
016504  005027                          CLR     (PC)+
016506  000000                  1$:     .WORD   0
016510  005267  177772          2$:     INC     1$              ;DELAY WAITING FOR TELETYPE TO FINISH
016514  001375                          BNE     2$              ;TYPING CHARACTER BEFORE ISSUING RESET
016516  000005                          RESET
016520  105737  177570                  TSTB    @#SWR
016524  100003                          BPL     3$
016526  004767  162500                  JSR     PC,.PRINT       ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
016532  016720                          ENDMSG
016534  013702  000042          3$:     MOV     @#42,R2         ;CHECK DDP/ACT11 MONITOR HOOK
016540  001405                          BEQ     DONE1
016542  000005                          RESET
016544  004712                  LOGICAL:JSR     PC,(R2)         ;GO TO DDP/ACT11 MONITOR VIA 42
016546  000240                          NOP
016550  000240                          NOP
016552  000240                          NOP
016554  000137  002224          DONE1:  JMP     @#START3                ;RESTART PROGRAM

                                        ;THE BELOW TABLE REPRESENTS THE 'NEW' PSW SET BY THE PROGRAM ON
                                        ;SUCCESSIVE PASSES.
                                        ;NOTE THE BELOW TABLE MAY BE MODIFIED TO CAUSE THE PROGRAM TO RUN
                                        ;UNDER USER DEFINED PARAMETERS BY PATCHING IN THE DESIRED PASS PARAMETER
                                        ;FOR EXAMPLE TO CAUSE THE PROGRAM TO RUN WITHOUT SETTING THE 'T' BIT
                                        ;IN ALL PASSES PATCH OUT THE 'T' BIT IN THE TABLE.
016560  000000                  PSWTAB: 000000                  ;ALL 11 FAMILY CP'S
016562  000020                          000020

                                        ;THE BELOW TABLE IS THE 'BIT MASK' USED TO DETERMINE THE INDEX VALUE
                                        ;NEEDED TO SET THE 'NEW' PSW.
016564  177774                  CPPASS: 177774                  ;11/05
016566  177774                          177774                  ;11/20

                                        ;THE BELOW TABLE REPRESENTS THOSE BITS IN THE CP WHICH CAN BE SET/CLEARED
016570  000377                  PSWBIT: 000377                  ;11/05
016572  000377                          000377                  ;11/20

                                        ;THE BELOW TABLE CONTAINS THE # OF PASSES REQUIRED TO COMPLETE TEST
016574  000002                  PASTAB: .WORD   2               ;11/05
016576  000002                          .WORD   2               ;11/20
                                        ;MESSAGES
016600  005015  047514  020127  MSG1:   .ASCIZ  <15><12>'LOW LIMIT?'
016606  044514  044515  037524
016614     000
016615     110  043511  020110  MSG2:   .ASCIZ  'HIGH LIMIT?'
016622  044514  044515  037524
016630     000
016631     015  052012  044510  ILLTEST:.ASCIZ  <15><12>'THIS TEST INVALID FOR 11/40-11/45 PLEASE RUN DCQKC'<15><12>
016636  020123  042524  052123
016644  044440  053116  046101
```

```
016652  042111  043040  051117
016660  030440  027461  030064
016666  030455  027461  032464
016674  050040  042514  051501
016702  020105  052522  020116
016710  041504  045521  006503
016716  000012
016720  005015  042040  050532  ENDMSG: .ASCIZ  <15><12>' DZQKC DONE'<207>
016726  041513  042040  047117
016734  103505     000
        000001                          .END
```

```
ADCB2   004564          1446    1448#
ADCB5   005374          1675    1677#
ADCB6   006062          1828    1829    1831#
ADCB7   006710          2036    2037    2038    2040#
ADC0    002524           857     858     859     861#
ADC1    003400          1080    1081    1082    1084#
ADC2    004374          1378    1380#
ADC5    005202          1602    1603    1605#
ADC6    005672          1773    1774    1776#
ADC7    006604          2000    2001    2003#
ADD0    007376          2201    2202    2203    2205#
ADD1    007602          2273    2274    2276#
ADD1A   010026          2356    2357    2358    2360#
ADD1B   010044          2365    2366    2368#
ADD2    010432          2502    2503    2505#
ADD3    011106          2661    2663#
ADD6    011450          2758    2759    2761#
ADD7    012112          2870    2871    2872    2874#
ASLB1   003742          1212    1213    1215#
ASLB1A  004166          1299    1300    1302#
ASLB3   005364          1669    1670    1672#
ASLB4   004670          1483    1484    1485    1487#
ASLB6   006044          1820    1821    1822    1824#
ASLB7   007006          2068    2069    2071#
ASL0    002646           901     902     903     904     906#
ASL1    003554          1143    1144    1145    1147#
ASL3    005116          1571    1572    1574#
ASL4    004466          1410    1411    1412    1414#
ASL6    005642          1761    1762    1764#
ASL7    006432          1947    1948    1950#
ASRB1   004036          1248    1250#
ASRB1A  004052          1254    1255    1257#
ASRB2   004634          1468    1469    1471#
ASRB2A  004652          1476    1477    1479#
ASRB5   005324          1650    1651    1653#
ASRB6   006162          1860    1861    1863#
ASRB7   007024          2075    2076    2078#
ASR0    002674           915     916     917     919#
ASR1    003442          1100    1101    1102    1104#
ASR2    004410          1384    1385    1387#
ASR3    005102          1565    1567#
ASR6    005524          1723    1724    1726#
ASR7    006466          1961    1962    1964#
BELL    001747           667     690#
BICB1   010220          2426    2427    2429#
BICB1A  010242          2437    2440#
BIC0    007310          2172    2173    2174    2176#
BIC1    007724          2319    2320    2322#
BIC2    010522          2531    2532    2533    2535#
BIC3    011120          2666    2668#
BIC7    012570          3000#   3002
BINB7   012336          2935    2943#
BIN1    010400          2469    2472    2475    2478    2481    2484    2487    2492#
BISB1   010206          2421    2423#
BIS0    007266          2163    2164    2166#
BIS0A   007344          2190    2192#
```

```
BIS1     007712            2313    2314    2316#
BIS2     010460            2515    2517#
BIS2A    010562            2549    2550    2552#
BIS7     012530            2990#   2992
BITB1    010176            2415    2416    2418#
BITB2    010736            2604    2605    2607#
BITB3    011270            2718    2721#
BITB6    011654            2805#
BIT1     007640            2290    2291    2293#
BIT13 =  020000             413#
BIT14 =  040000             412#    439
BIT15 =  100000             411#
BIT2     010546            2542    2543    2545#
BIT6  =  000100             415#
BIT8  =  000400             414#
BPTVEC=  000014             377#
C     =  000001             363#
CBIT     015536            3661#
CC0      002340             781     782     783     784     785     786     787     789#
CC1      002354             793     794     795     797#
CC2      002370             801     802     803     805#
CC3      002402             809     810     812#
CC4      002416             816     817     818     820#
CHKSP    015342            3599#
CLRD     002442             828     829     830     831     833#
CMPB1    010152            2405    2407#
CMPB2    010722            2597    2598    2600#
CMPB3    011302            2724    2727#
CMPN     015572            3675#
CMPO     007170            2126    2127    2128    2130#
CMPOA    007432            2216    2219#
CMP1     007622            2281    2282    2283    2285#
CMP1A    007742            2326    2327    2328    2330#
CMP2     010450            2509    2510    2512#
CMP7     012046            2856    2857    2859#
COMB1    004020            1239    1240    1242#
COMB1A   004200            1305    1306    1308#
COMB2    004546            1437    1438    1440#
COMB5    005302            1640    1642#
COMB6    006114            1841    1842    1844#
COMB7    006724            2043    2044    2046#
COMO     002506             848     849     850     851     853#
COM1     003566            1150    1151    1153#
COM3     005160            1592    1594#
COM4     004276            1341    1343#
COM6     005510            1717    1718    1720#
COM7     006556            1989    1990    1992#
CPCHK    014204            3347#
CPPASS   016564            3832    3880#
DBINB7   012332            2940#   2972*   2973*   2977*   2978*   2979    2983*   2990    2991*   2998*   3000    3001*
DBIN7    011746            2831#   2846*   2850*   2855    2861*   2869*   2876*   2880*   2884
DDATA    011312            2733#   2737*   2739*   2743    2747    2748    2751*   2752    2757*   2764*   2771*   2772    2775*
                           2778    2782
DDATAB   011742            2801*   2802*   2807*   2810    2814*   2815    2819    2826#
DECB1    003770            1225    1226    1228#
DECB1A   004120            1277    1278    1280#
```

```
DECB2    004702      1490   1491   1493#
DECB5    005442      1698   1700#
DECBFA   006214      1872   1873   1874   1876#
DECB7    006772      2062   2063   2065#
DECD     002566       874    875    876    877    879#
DEC1     003360      1073   1075#
DEC1A    003630      1171   1173#
DEC2     004444      1401   1402   1404#
DEC5     005134      1578   1579   1580   1582#
DEC6     005656      1767   1769   1770#
DEC7     006450      1954   1955   1957#
DIGITS   001652       632    635    636    645    651    661    678#  3817
DIGTAB   001652       614    674#
DISPLA=  177570       400#   756*
DONE     016466      3929   3847#  3848
DONE1    016554      3861   3867#
EMTVEC=  000030       380#   757*  3259   3260*  3261*  3263*  3300*  3301*
EMT1     013724      3260   3268#
EMT1B    014004      3268   3272   3273   3276   3277   3281   3282   3285   3286   3297   3292#
EMT1C    014010      3266   3294#
EMT1D    014022      3295   3296   3299#
END      016246      3804#
ENDMSG   016720      3859   3907#
END1     016250      3805#
ERROR    001412       622#   768
ERRPC    001707       643    433#
ERRPCO   001721       659    785#
ERRVEC=  000004       373#   493*   500*   740*   741*   764*  3015*  3016*  3017   3057*  3349*  3363*  3398*
                     3389*  3429*  3475*  3764*  3802*
FACTOR   001004       511#   523    551    654    656    777*  1798   1912*  2735   2793   2902   2903   2907
                     2908   2932*  2961   3016   3140   3152   3160   3175   3199   3239   3261   3308   3380*
                     3482   3490   3501   3734*
FORXOR   000740       493#   517
FPEVEC=  000244       385#
FRSTAC   001010       515#   546    774*  1909*  2574   2929*  3377*
FRSTME   001012       516#   554    749*   756*
GSTST    003160      1018#
HLT    = 104400       418#   564    789    797    805    812    820    833    843    953    961    970    979
                      889    897    906    912    919    926    934    956    966    975    983    990    999
                     1015   1050   1068   1075   1084   1091   1097   1104   1111   1119   1126   1134   1139
                     1147   1153   1161   1168   1173   1191   1197   1203   1209   1215   1221   1229   1235
                     [illegible] 1250   1257   1261   1268   1273   1280   1287   1293   1297   1302   1309   1312
                     [illegible] 1321   1338   1343   1350   1357   1363   1368   1372   1390   1397   1394   1399
                     [illegible] 1408   1414   1418   1421   1440   1444   1449   1456   1464   1471   1479   1487
                     [illegible] 1499   1505   1510   1515   1521   1529   1533   1536   1554   1561   1567   1574
                     1582   1588   1594   1599   1605   1612   1618   1642   1646   1653   1659   1666   1672
                     1677   1684   1688   1694   1700   1714   1720   1726   1734   1742   1748   1756   1764
                     1770   1776   1780   1783   1797   1802   1808   1817   1824   1831   1938   1944   1951
                     1857   1863   1869   1876   1881   1887   1893   1935   1942   1950   1957   1964   1972
                     1979   1985   1992   1996   2003   2008   2015   2032   2040   2046   2053   2059   2065
                     2071   2078   2084   2091   2098   2107   2115   2121   2130   2139   2153   2159   2166
                     2176   2184   2192   2195   2205   2210   2219   2228   2234   2239   2243   2249   2255
                     2276   2285   2293   2302   2309   2316   2322   2330   2341   2348   2353   2360   2368
                     2373   2396   2407   2418   2423   2429   2434   2440   2492   2505   2512   2517   2524
                     2535   2545   2552   2558   2566   2573   2590   2594   2600   2607   2612   2641   2649
                     2654   2657   2663   2658   2673   2705   2708   2714   2721   2727   2745   2750   2755
```

```
                    2761    2767    2774    2780    2784    2805    2809    2812    2817    2821    2849    2853    2859
                    2866    2874    2878    2882    2886    2970    2975    2981    2985    2995    3005    3021    3023
                    3025    3027    3029    3031    3033    3035    3037    3039    3041    3043    3045    3047    3049
                    3075    3095    3111    3129    3137    3148    3173    3197    3207    3219    3227    3231    3253
                    3267    3292    3299    3305*   3316    3318    3321    3325    3335*   3423    3465    3496    3497
                    3498    3555    3574    3579    3608    3621    3628    3636    3642    3653    3670    3685    3694
                    3791
ICNT    001000       508*     633     763*    766     3347    3815    3825*   3827    3830
ILLTES  016631      3369    3897*
INC61   003704      1194    1195    1197*
INC62   004776      1525    1526    1527    1529*
INC63   005334      1656    1658*
INC66   006026      1815    1817*
INC66A  006176      1866    1867    1869*
INC67   006756      2056    2057    2059*
INC0    002606       883     884     885     886     888*
INC1    003474      1115    1116    1117    1119*
INC3    005170      1597    1599*
INC4    004326      1354    1355    1357*
INC6    005620      1752    1753    1754    1756*
INC7    006616      2006    2008*
IOTVEC= 000020       378*    3237*   3238*   3240*   3254*   3255*   3306*   3330    3332*   3333*   3341*   3393*   3453*
IOT1    013624      3237    3244*
IOT1A   013632      3243    3247*
IOT1B   013646      3248    3249    3250    3253*
JMP1    013052      3071    3072    3073    3075*
JMP3    013122      3091    3095*
JMP4    013160      3107    3111*
JMP5    013224      3125    3129*
JMP6    013244      3137*
JMP7    013302      3148*
JSRTST  013306      3152*
JSR1    013370      3159    3161    3164    3165    3166    3167    3173*
JSR1A   013372      3162    3175*
JSR3    013456      3186    3193    3197*
JSR3A   013460      3187    3199*
JSR4    013534      3209    3210    3211    3212    3216    3219*
JSR4A   013536      3208    3222*
JSR6A   013570      3226    3234*
KPTR  = 000600       406*     477     737    3468    3809
KW11    016070      3746    3764*
LKS   = 177546       393*    3768*
LKVEC = 000100       383*    3765*   3766*   3789*   3795*
LOGICA  016544       437    3863*
LPB   = 177516       402*
LPS   = 177514       401*
LSTMEM  002140       557     747*    748*    761*
MMVEC = 000250       386*
MOVB1   010126      2390    2391    2392    2393    2396*
MOV0    007120      2104    2105    2107*
MOV0A   007150      2119    2121*
MOV1    010012      2351    2353*
MOV7    012016      2847    2849*
MSG     016060      3740*   3741    3750    3755    3758*   3821*
MSG1    016600       753    3891*
MSG2    016615       758    3894*
```

```
N       = 000010          366#
NE681     003754         1218    1219    1221#
NE684     004726         1502    1503    1505#
NE686     006132         1848    1849    1851#
NE687     007040         2081    2082    2084#
NE6D      002626          893     894     895     897#
NE6D1     003620         1165    1166    1168#
NE6D2     004350         1366    1368#
NE6D5     005146         1585    1586    1588#
NE6D6     005544         1730    1731    1732    1734#
NE6D7     006524         1976    1977    1979#
NULL    = 000000          536#
NULLS     016064         3740    3760#
OAERR     012616         3008#
OPT.CP    014254         3360#   3366    3405    3461    3483    3486    3556    3559    3826
OVFLW     014332         3384#
PASCNT    001672          638     680#    3821
PASFLS    001702          635*    636*    682#    3[illegible]3
PASTAB    016574         3827    3888#
PC      = 000007          352#    441*    442*    443[illegible]  454*    466*    478*    482*    536*    567*    568*    580*    582*
                          590*    596*    601*    618[illegible]  619*    628*    630*    632*    634*    637*    641*    642*    644*
                          646*    649*    650*    657*    658*    660*    663*    666*    714*    746*    752*    757*    772
                          775     778    1021*   1023*   1035*   1059    1179    1327    1427    1543    1626    1704*   1706
                         1898    1900*   1907    1910    1913    1921    2023    2103    2206    2208    2212    2215    2263
                         2380    2445    2569    2582    2621    2685    2835    2893*   2894    2895    2896*   2897    2898*
                         2899    2900    2921*   2902    2903    2907*   2908*   2910    2911    2912    2913*   2914*   2919
                         2921[illegible]  2927    2930    2933    2943    3009    3063    3078    3098    3113    3201    3224    3225*
                         3233*   3312    3337    3339*   3343    3359*   3368*   3375    3378    3381    3386    3394    3437
                         3441*   3451*   3458    3469    3471    3480    3494*   3546    3584    3586    3594    3655    3661*
                         3663    3675    3726    3728*   3735    3752*   3757*   377[illegible]  3796*   3816*   3851*   3858*   3863*
PDWN      000610          471#    475     3823
PFAIL     000656          483     486#
PFVEC   = 000024          379#    471*    475*    3823*   3824*
PIRQ    = 177772          391#   3352
PIRVEC  = 000240          384#
PRTY4   = 000200          370#
PRTY6   = 000300          369#
PRTY7   = 000340          368#   3331
PSW     = 177776          389#    524*    730*   1018    1033*   1039*   3240    3263    3310    3355    3384    3395*   3543
                         3544*   3551    3591    3778    3780*   3790*   3796*   3805*
PSWBIT    016570         3557    3884#
PSWCHK    015124         3543#
PSWTAB    016560         3835    3875#
PUF       000620          471     475#
RECD      001752          696#    754     759
RELOC     001132          544#   1900    2921    3339    3728
RELR1     001006          513#    528*    529*
REL0      002272          772#
REL1      006302         1907#
REL2      012274         2927#
REL3      014306         3348    3367    3375#
RESTRP    014674         3479#
RESVEC  = 000010          374#   3017*   3058*   3364*   3489*   3490*   3538*
RETPC     001230          553*    569#
RCLB1     003716         1200    1201    1203#
RCLB2     004620         1460    1461    1462    1464#
```

```
ROLB3   005410          1681    1682    1684*
ROLB6   006146          1854    1855    1857*
ROLB6A  006242          1884    1885    1887*
ROLB7   007056          2088    2089    2091*
ROLO    002660           909     910     912*
ROL1    003414          1087    1088    1089    1091*
ROL1A   003426          1094    1095    1097*
ROL3    005216          1609    1610    1612*
ROL4    004424          1391    1392    1394*
ROL6    005474          1710    1711    1712    1714*
ROL7    006634          2012    2013    2015*
RORB1   004004          1232    1233    1235*
RORB1A  004074          1265    1266    1268*
RORB4   004602          1452    1453    1454    1456*
RORB5   005352          1662    1663    1664    1666*
RORB6   006100          1835    1836    1838*
RORB7   006742          2050    2051    2053*
RORO    002544           865     866     867     868     870*
ROR1    003344          1065    1066    1068*
RORIA   003456          1107    1108    1109    1111*
ROR2    004312          1347    1348    1350*
ROR5    005070          1558    1559    1561*
ROR6    005564          1738    173     1740    1742*
ROR7    006506          1969    1970    1972*
RTI1    016464          3839    3841*
R0      =%000000         345*    546*    547     559     560     562     565     572     573*    576     579*    607*    616*
                         696     700*    701*    702     711     712     719*    723     731*     42*    743     745*    746
                         772*    773     774     775*    776*    777     827*    837     947*    856*    864*    873*    882*
                         892*    900*    908*    914*    922*    929*    929*    938*    940*    341     953     961*    962*
                         964*    968*    972*   1027*   1028    1039    1044    1429*   1430    1432*   1546*   1548    1629*
                        1630*   1633    1634*   1635    1636*   1706*   1707    1781    1787*   1788*   1907*   1908    1909
                        1910*   1911*   1912    1921*   1922    1923    1924*   1925    193 *   1927    1928    1929*   1930*
                        2019    2020*   2021    2022*   2103*   2109    2111    2132    2162    2168    217     2178    2180
                        2182    2186    2189    2206*   2207    2208    2212*   2213*   2214    2215    2217    2622*   2623
                        2624    2626    2628    2629    2630    2631    2685*   2686    2687    2690    2692    2694    2695
                        2697    2769*   2770    2772*   2785    2835*   2836    2937    2839    2841    2842    2991*   2896
                        2899    2900    2927*   2928    2929    2930*   2931*   2932    2943*   2944    2945*   2946*   2947*
                        2948    2949*   2950*   295 *   2959*   2960*   3063*   3064*   3066    3083*   3085*   3092    3093
                        3099*   3108    3109    31  *   3126    3127    3176*   3185*   3188*   3224*   3228    3229    3241*
                        3244*   3247*   3251    32  *   3270*   3271*   3275*   3280*   3284*   3289*   3294*   329     3312*
                        3322*   3323    3350*   3353*   3356*   3358*   3359    3375*   3376    3377    3378*   3379*   3390
                        3420*   3432*   3448*   3458    3463    3556*   355 *   3561*   3563    3613*   3614    3626    3640
                        3651    3663*   3664*   3665*   3666*   3675*   3676    3680    3689    3926*   392
R1      =%000001         346*    499*    501     527     528     540*    604*    611*    613*    732*    778*   1020    1029*
                        1029    1045    1053*   1913*   2023*   262 *   2622    2933*   3343*   3381*   3490*   3735*

R10     =%000000         353*
R11     =%000001         354*
R12     =%000002         355*
R13     =%000003         356*
R14     =%000004         357*
R15     =%000005         358*
R2      =%000002         347*    548     550     553     554*    555     559*    562     567     604     605*    609*    633*
                         639*    640     648*    652*    653     656*    733*    941*    942*    943     952     969*    971*
                         974*   1002*   1003*   1004    1006*   1007*   1009*   1029*   1030    1046    1059*   1060*   1061*
                        1064*   1072*   1079*   1086*   1093*   1099*   1106*   1114*   1121*   1129*   1137*   1142*   1149*
                        1156    1164*   1173*   1430*   1431*   1442*   1459*   1473*   1475*   1489*   1501*   1512    1519*
```

1531 1534 1626* 1627 1628 1629 1631 1632 1633* 1635* 1636 1637 1639*
1644 1655* 1668* 1674* 1686* 1696* 1697* 1898* 1899* 1925* 1926 1929 1930
2109* 2111* 2118 2125 2132* 2133 2134 2135* 2136 2142* 2143* 2145* 2147
2149 2154 2156 2160* 2162* 2214* 2217 2383* 2384* 2385* 2386* 2389 2398*
2400* 2404 2414 2420 2425 2431* 2432 2436* 2438 2452* 2453* 2459* 2460
2461* 2462 2463* 2465* 2467 2468 2479* 2480 2486 2498 2501 2514 2520
2526* 2530 2537* 2538 2541 2548 2556 2560* 2561* 2566 2569* 2570 2571
2574 2627* 2631* 2637 2643* 2645 2652 2655 2658* 2660 2665 2671 2735*
2736 2793* 2794 2837* 2838 2839* 2840 2919* 2920* 2952* 2962* 3008* 3020*
3022* 3024* 3026 3028* 3030* 3032* 3034* 3036* 3038* 3046 3077* 3088* 3090*
3097* 3101* 3106* 3121* 3124* 3132* 3135* 3142* 3146* 3153* 3154* 3157 3158
3168* 3179* 3194 3195 3213* 3214* 3215 3337* 3338* 3421* 3431* 3434* 3436*
3438* 3440* 3442* 3448 3481* 3482* 3485* 3491 3493 3562* 3563 3566 3570
3572 3576 3580* 3606* 3646* 3647 3649 3676* 3677* 3680 3689 3726* 3727*
3915* 3817* 3819 3820 3830* 3831* 3832* 3860* 3863

R3 =%000003

348# 550* 560 603* 606* 610* 615* 734* 943* 944* 945 951 978*
979* 991* 1004* 1008 1011* 1030* 1031 1047 1179* 1180* 1181 1183* 1186*
1211* 1217* 1224* 1231* 1244* 1276* 1283* 1298* 1304* 1310* 1314* 1316* 1319
1543* 1544* 1545* 1546 1547 1548* 1549 1552 1557* 1570* 1584* 1596* 1614*
2118* 2125 2133* 2134* 2136 2168* 2171* 2188* 2189* 2193* 2196 2197* 2200
2223* 2224 2225 2231 2251 2266* 2267* 2269* 2272 2280 2289 2298 2304*
2305* 2307 2312 2318 2325 2332* 2334* 2335 2337 2344 2350 2355 2362*
2363 2364 2409* 2411* 2412 2450* 2451* 2452 2462* 2464 2466* 2470 2471
2473* 2474 2476* 2477 2488 2489* 2490 2582* 2593 2594* 2585* 2586 2592
2596 2603 2610 2630* 2632* 2687* 2688 2689 2690* 2691* 2692* 2693 2702*
2703* 2709 2710* 2711* 2715* 2716* 2717 2719 2723 2725 2795* 2796* 2797
2840* 2841 2842* 2954* 2963* 3009* 3012 3013* 3014* 3038 3040 3042 3044*
3078* 3081 3082* 3083 3084* 3085 3086 3093 3113* 3116 3117* 3118* 3119
3123 3127 3190* 3191* 3192 3201* 3204 3206 3398* 3399* 3400 3430 3491*
3494 3503 3505* 3571* 3572 3577* 3599* 3607 3610 3614* 3615* 3617 3619
3624 3630* 3631* 3632 3634 3818* 3819* 3820*

R4 =%000004

349# 548* 549* 555* 557 602* 735* 945* 946* 947 950 985* 986*
983 1008* 1013* 1031* 1032 1048 1181* 1182* 1188* 1193* 1199* 1205* 1238*
1246* 1252* 1253* 1259* 1264* 1271* 1289 1295* 1327* 1328* 1329 1346* 1353*
1360* 1370* 1383* 1390* 1406* 1409* 1416 1419 1547* 1564* 1577* 1591* 1601*
1608* 1615* 2178* 2179* 2180* 2181* 2182 2186* 2187* 2188 2196* 2199* 2200*
2224* 2225* 2226 2230* 2231* 2232 2236* 2237 2241 2245* 2246* 2247 2251*
2252* 2253 2263* 2264 2265* 2266 2270* 2272* 2280 2289 2297* 2298* 2307*
2310* 2312* 2318* 2325 2335* 2336* 2337* 2344* 2350* 2355* 2363* 2364* 2370*
2371 2448* 2449* 2450 2460* 2470* 2471 2474 2480 2483 2485* 2486* 2489*
2496 2586* 2587* 2588* 2592* 2596 2603 2610* 2628* 2633* 2651* 2693* 2695*
2696* 2697* 2700* 2701* 2702 2703 2710 2711 2715 2716 2717 2719 2723
2725 2794* 2795 2955* 2964* 3098* 3099 3103 3104 3105 3109 3169* 3170*
3171 3175* 3178 3181 3184* 3192 3199* 3200 3206* 3215 3217 3218* 3493*
3503 3551* 3553 3569* 3570* 3571 3576* 3577 3582* 3591* 3592 3654

R5 =%000005

350# 547* 549 565 697* 706* 707* 708* 716 717* 720* 721* 722*
723* 736* 754* 759* 947* 948* 949 992* 993* 995* 997 1032* 1034
1049 1329* 1330* 1334 1340* 1365* 1375* 1377* 1396* 1400* 1419 1427* 1429*
1429 1436* 1445* 1451* 1467* 1482* 1495* 1507 1524* 1534 1637* 1649* 1651*
1680* 1691* 2147* 2149 2154* 2156 2380* 2381 2382* 2383 2389* 2394* 2401*
2402* 2404 2412* 2414 2420* 2425* 2432 2436 2445* 2446 2447* 2448 2464*
2467* 2468 2477 2482* 2483 2490* 2496* 2497* 2501* 2508 2514* 2520* 2527*
2530* 2538* 2539* 2541 2548* 2554* 2555* 2556 2570* 2571 2624* 2625 2626*
2627 2637* 2644* 2645* 2652 2655 2660* 2665* 2670* 2671 2688* 2705 2712
2736* 2797* 2956* 2965* 3152* 3154 3157* 3160 3163* 3171 3177* 3178* 3179
3180* 3181* 3184 3195 3200*

```
SBCB1   003730          1206    1207    1209*
SBCB3   005430          1692    1694*
SBCB4   004714          1496    1497    1499*
SBCB6   006002          1805    1806    1808*
SBCB7   006670          2028    2029    2030    2032*
SBC0    002710           923     924     926*
SBC1    003510          1122    1123    1124    1126*
SBC1A   003526          1130    1131    1132    1134*
SBC5    005230          1616    1618*
SBC6    005600          1745    1746    1748*
SBC7    006412          1939    1940    1942*
SBINB7  012322          2936*   2967*   2968    2972    2979    2983    2987*   2990*   2993    2997*   3000*   3003
SBIN7   011744          2830*   2845*   2846    2855    2861    2868*   2869    2876    2880    2884
SCOPE = 104000           419*    821     935    1054    1174    1322    1422    1537    1619    1701    1784    1897    2016
                        2099    2220    2256    2374    2493    2567    2576    2613    2674    2728    2787    2822    2887
                        2917    3006    3059    3149    3270    3256    3302    3334    3476    3539    3589    3658    3724
SCOPEA  001016           521*    767
SCOPEB  001046           527*    538     541
SCOPEC  001102           526     534*
SCOPED  001114           537*    539*    3479*
SCOPEE  001126           535     540*
SCOPEF  001120           539*   3834*   3838*
SDATA   011310          2732*   2738*   2739    2740*   2741*   2746*   2747*   2748    2752    2757    2763*   2764    2770*
                        2775    2776*   2777*   2778    2781*   2782    2785
SDATAB  011740          2799*   2800*   2801    2803    2807    2810    2814    2815    2825*
SLR   = 177774           390*
SP    = 000006           351*    447     448*    449*    450*    451*    452*    453*    459     460     461     462     463
                         464     465     477*    499     501*    522     527*    540     572*    573     574*    576*    578
                         579     583     586*    588*    595     628     630     696*    737*    765*    949*    950*    951*
                         952*    953*    954    1020*   1021    1034*   1035    1037*   1044*   1045    1046*   1047*   1048*
                        1049*   1052*   1053    2910    2911    2961*   2962    2963    2964    2965    3052*   3053*   3056*
                        3170    3191    3214    3229    3259*   3300    3323    3326    3327    3386*   3387*   3388    3390*
                        3391    3392*   3393    3394*   3395*   3400*   3403    3404    3412*   3413*   3414*   3415*   3420
                        3421    3422*   3430*   3433*   3435*   3437*   3444*   3457*   3468*   3469*   3471*   3472*   3474*
                        3500*   3501*   3545*   3546*   3547*   3550*   3583    3584*   3586*   3587*   3592*   3593*   3594*
                        3595*   3599    3601*   3602*   3604*   3606    3607*   3610*   3613    3616*   3619    3623*   3626
                        3631    3634    3638    3640    3644*   3645*   3647    3649    3651    3654*   3655*   3656*   3797*
                        3806*   3807*   3809*   3835*   3836    3840*
SPCHK   015372          3613*
SR0   = 177572           398*
START   002066           441     484     730*
START1  002160           442     751*
START2  002230           764*   3840
START3  002224           443     750     763*   3867
STATUS  001714           647     684*
STKPTR= 000500           405*    765    3056    3422    3457    3474
SUB0    007136          2112    2113    2115*
SUB1    007660          2299    2300    2302*
SUB1A   007766          2338    2339    2341*
SUB1B   010002          2345    2346    2348*
SUB2    010476          2521    2522    2524*
SUB2A   010616          2563    2564    2566*
SUB3    011032          2638    2639    2641*
SUB3A   011054          2646    2647    2649*
SUB6    011470          2765    2767*
SUB7    012060          2862    2863    2864    2866*
```

```
SWAB0   002726            930    931    932    934*
SWAB1   004134           1284   1285   1287*
SWAB2   004340           1361   1363*
SWAB4   004760           1519   1521*
SWAB6   006226           1879   1881*
SWAB7   006540           1983   1985*
SWR   = 177570            399*   525    530    532    534    544    622    626    664    668   3813   3856
T     = 000020            367*
TBITVE= 000014            375*  3550   3565*  3568*  3583*
TICKS   000776            506*  3799*
TKB   = 177562            395*   700
TKS   = 177560            394*   698
TPB   = 177566            397*   595*   704*   711*  3741*  3755*
TPS   = 177564            396*   593   3736   3742  3744*  3747*  3783   3811   3822*  3847   3849
TPVEC = 000064            382*  3738*  3739*  3781  3785*  3794*
TRAPVE= 000034            381*   768*   769*  3306  3307*  3308*  3310*  3330*  3331*
TRAP1   014112           3307   3320*
TRAP1C  014136           3319   3330*
TRTVEC= 000014             376*
TSTB1   004146           1290   1291   1293*
TSTB2   004736           1508   1510*
TSTB2A  004746           1513   1515*
TSTB6   005754           1793   1794   1795   1797*
TSTJ    002464            838    839    840    841   843*
TST1    003604           1157   1158   1159   1161*
TST2    004266           1335   1336   1338*
TST6    006256           1890   1891   1893*
TTYCHK  015734           3734*
UBM6    006262           1787   1789*  1792   1800  1804*  1811*  1813*  1819*  1827*  1834*  1840*  1847*  1853*
                         1859*  1865*  1871*  1878*  1883*  1889   1896*
UBREAK= 177770            392*   532*
UB7     011754           2824   2835*
UWM6    005450           1705*  1709*  1716*  1722*  1729*  1737*  1744*  1751*  1758*  1760*  1766*  1772*  1778*
UWM7    006332           1918*
UW7     006336           1916   1921*
V     = 000002            364*
Z     = 000004            365*
$CRLF   001744            689*   715   3753
$FILLS  001002            509*   586
$FORMO  001330            600*   634    641    649    657   3816
$RESTR  000240            459*   618    663
$SAVR   000214            447*   601    632
.     = 016737            436*   438*   440*   469*   490*   505*   507*   563    594    623    669    691*   744
                          776    788    796    804    811    819    832    842    852    860    869    879    887
                          896    905    911    918    925    933    955    965    975    982    989    999   1014
                         1057   1067   1074   1083   1090   1096   1103   1110   1118   1125   1133   1139   1146
                         1152   1160   1167   1172   1177   1190   1196   1202   1208   1214   1220   1227   1234
                         1241   1249   1256   1260   1267   1272   1279   1286   1292   1296   1301   1307   1311
                         1317   1320   1325   1337   1342   1349   1356   1362   1367   1371   1379   1386   1393
                         1397   1403   1407   1413   1417   1420   1425   1439   1443   1447   1455   1463   1470
                         1478   1486   1492   1498   1504   1509   1514   1520   1529   1532   1535   1540   1553
                         1560   1566   1573   1581   1587   1593   1598   1604   1611   1617   1622   1641   1645
                         1652   1657   1665   1671   1676   1683   1687   1693   1699   1713   1719   1725   1733
                         1741   1747   1755   1763   1769   1775   1779   1782   1796   1801   1807   1816   1823
                         1830   1837   1843   1850   1856   1862   1868   1875   1880   1886   1892   1895   1911
                         1934   1941   1949   1956   1963   1971   1978   1984   1991   1995   2002   2007   2014
```

```
                    2031    2039    2045    2052    2058    2064    2070    2077    2083    2090    2097    2106    2114
                    2120    2129    2137    2157    2165    2175    2183    2191    2194    2204    2209    2218    2227
                    2233    2238    2242    2248    2254    2260    2275    2284    2292    2301    2308    2315    2321
                    2329    2340    2347    2352    2359    2367    2372    2377    2395    2399    2406    2417    2422
                    2428    2433    2439    2442    2491    2504    2511    2516    2523    2534    2544    2551    2557
                    2565    2572    2579    2589    2593    2599    2606    2611    2640    2648    2653    2656    2662
                    2667    2672    2704    2707    2713    2720    2726    2731    2744    2749    2753    2760    2766
                    2773    2779    2783    2804    2808    2811    2816    2820    2848    2851    2852    2858    2865
                    2873    2877    2881    2885    2915    2931    2969    2974    2980    2984    2994    3004    3010
                    3046    3048*   3067    3069    3074    3079    3087    3089    3094    3110    3114    3128    3136
                    3147    3230    3245    3252    3290    3291    3298    3315    3317    3320    3324    3379    3387
                    3487    3554    3573    3620    3627    3635    3641    3652    3669    3684    3693    3737    3743
                    3794    3807    3812    3850
.PRINT  001232       482     572*    637     642     644     646     650     658     660     666     714     752     757
                    3368    3752    3858
```

```
FORMAT    428#    634     641     649     657    3816
GETDAT    435#    692
KW11L     430#
PRINT     426#    482     637     642     644     646     650     658     660     666     714     751     757    3368    3752
         3858
RELOCA    432#    770    1905    2925    3373
RELOCB    433#   1898    2919    3337    3726
RELOCC    434#    543
RESTOR    424#    618     662
SAVE      422#    600     632
SCOP4     421#    518
WHICHC    430#   3344
.ERRO     435#    621
.FORMA    429#    599
.RESTO    425#    457
.SAVE     423#    445
..PRIN    427#    571
```

```
ADC      856  1079  1137  1377  1601  1772  1999  2893  3665
ADCS    1186  1246  1442  1445  1674  1813  1927  2035  2095  3036  3270  3294
ADD      555   574   986  1788  1899  2134  2200  2213  2272  2355  2364  2370  2501  2644  2660
        2757  2775  2869  2896  2901  2908  2920  2962  2963  2964  2965  3016  3022  3052  3064
        3084  3118  3140  3154  3170  3178  3181  3191  3214  3238  3261  3308  3322  3338  3387
        3390  3395  3429  3437  3472  3482  3485  3490  3501  3547  3587  3595  3645  3656  3727
ASL        5   609   720   721   722   900   942   944   946   948   993   995  1041  1142  1406
           3  1570  1760  1946  2135  2145  2740  2741  2751  2763  2776  2850  3024  3358  3434
          36  3438  3440  3442  3580  3666  3747  3831
ASLB       1  1298  1482  1668  1819  2067  2398  3247
ASR       37   708   914   979  1007  1099  1383  1564  1722  1758  1960  2868  3744
ASRB      44  1252  1253  1467  1475  1649  1859  2074  2479  2489  3289
BCC      809   848   857   874   893   901   915   923   980   987   996  1073  1087  1100  1115
        1143  1150  1165  1171  1200  1206  1212  1225  1232  1239  1254  1277  1305  1341  1347
        1366  1384  1391  1397  1410  1437  1452  1483  1490  1502  1525  1571  1578  1585  1592
        1597  1609  1640  1650  1656  1662  1669  1698  1717  1730  1738  1761  1767  1815  1820
        1841  1848  1860  1866  1872  1884  1939  1947  1961  1976  1989  2012  2028  2043  2050
        2056  2062  2068  2075  2088  2104  2150  2163  2172  2190  2201  2273  2290  2313  2319
        2326  2338  2356  2390  2415  2421  2426  2502  2509  2515  2531  2542  2549  2604  2646
        2666  2742  2758  2847  2862  2870  2992  3002  3071  3164  3209  3248  3285  3295  3581
        3681
BCS      498   781   828   838   865   883   909   930   963  1080  1107  1122  1130  1157  1194
        1218  1248  1265  1284  1290  1299  1335  1361  1407  1446  1460  1468  1476  1496  1508
        1519  1558  1602  1616  1675  1692  1710  1723  1745  1752  1773  1793  1805  1828  1835
        1954  1879  1890  1954  1983  2000  2036  2081  2112  2119  2126  2281  2299  2365  2521
        2563  2597  2638  2704  2765  2808  2856  3315  3603  3690
BEQ      523   526   531   563   655   665   783   850   867   876   885   895   955   965   975
         982   989   998  1014  1109  1117  1124  1132  1172  1202  1214  1220  1241  1249  1260
        1296  1311  1317  1320  1337  1355  1371  1412  1417  1420  1443  1454  1462  1485  1498
        1509  1513  1527  1532  1535  1553  1580  1611  1617  1645  1693  1699  1712  1732  1740
        1754  1782  1801  1807  1868  1874  1886  1892  1934  2014  2030  2038  2070  2083  2097
        2114  2120  2137  2183  2194  2203  2209  2218  2227  2233  2238  2242  2248  2254  2275
        2283  2292  2301  2328  2351  2372  2392  2395  2417  2427  2433  2439  2469  2472  2481
        2491  2504  2523  2533  2544  2557  2572  2589  2606  2611  2640  2653  2672  2707  2713
        2726  2744  2749  2753  2766  2773  2779  2793  2804  2811  2816  2820  2852  2864  2972
        2877  2885  2969  2974  2980  2984  2994  3004  3094  3110  3128  3136  3147  3172  3196
        3230  3252  3266  3298  3317  3324  3407  3460  3464  3487  3492  3504  3554  3564  3567
        3573  3578  3605  3620  3627  3635  3641  3652  3669  3683  3692  3756  3837  3861
BGE      794   804   860   896   911   933  1067
BGT      795   811   817   869   887
BHI      558   788   810   818
BIC      701   719  2171  2180  2318  2530  2554  2665  2747  2876  2914  3030  3576  3832
BICB    1038  2225  2425  2436  2486  2490  2610  2715  2716  2807  2814  2983  3000  3631
BIS      524   723   950   951   952   953  2162  2189  2312  2412  2514  2548  2643  2739  2802
        2880  3053  3413  3415  3444  3448  3561  3570  3768  3822
BISB    1033  2231  2251  2420  2467  2470  2592  2710  2711  2972  2990  3034
BIT      522   525   530   534   544   626   664  1018  2149  2156  2289  2541  2652  2748  2778
        2895  3327  3563  3566  3736  3811  3836  3847
BITB    2241  2414  2468  2471  2480  2603  2717  2719  2803  2819  2900  2903  2911  2968  3046
        3486  3638  3778
BLE      786   803   819   832   878   910   925  1074  3484
BLOS     787   842   905  1065  1094  1681  1969  2345
BLT      589   785   796   802   852   918  3367  3406
BMI      784   831   841   877   904  1110  1118  1125  1138  1152  1187  1196  1208  1267  1279
        1292  1301  1342  1356  1362  1367  1379  1386  1393  1413  1439  1455  1470  1486  1504
        1514  1520  1560  1566  1573  1587  1604  1641  1652  1671  1676  1687  1719  1725  1741
```

```
        1755  1763  1775  1779  1812  1830  1837  1843  1856  1862  1875  1880  1941  1943  1956
        1963  1978  1984  1991  1995  2007  2031  2045  2058  2064  2077  2090  2146  2157  2191
        2284  2315  2321  2340  2352  2367  2406  2422  2511  2516  2534  2551  2565  2593  2648
        2662  2667  2848  2858  2865  2881  3226  3282  3320  3684
BNE      481   535   538   545   552   561   566   577   584   612   617   627   703   713   816
         830   840   859   903   917   932   973  1010  1012  1019  1040  1042  1083  1089  1102
        1145  1160  1167  1227  1256  1447  1664  1795  1822  2106  2128  2152  2165  2174  2216
        2358  2437  2475  2478  2484  2487  2575  2718  2720  2724  2786  3073  3091  3107  3125
        3159  3161  3166  3186  3193  3211  3216  3250  3296  3348  3462  3618  3625  3633  3639
        3648  3650  3737  3751  3779  3788  3812  3828  3848  3854
BPL      594   623   669   699   793   851   868   886  1066  1082  1090  1096  1103  1133  1146
        1159  1234  1286  1307  1349  1403  1463  1478  1492  1528  1581  1598  1657  1665  1683
        1713  1733  1747  1769  1796  1823  1850  1971  2002  2039  2052  2129  2175  2204  2308
        2329  2347  2359  2393  2428  2599  2760  2873  3074  3167  3212  3273  3277  3287  3560
        3693  3743  3784  3814  3850  3857
ER       517   541   591   608   709   724   744   750  1043  1057  1177  1189  1247  1325  1425
        1540  1622  1814  1895  1916  2144  2260  2377  2442  2579  2616  2677  2731  2824  2915
        2935  3010  3050  3067  3069  3079  3087  3089  3100  3102  3114  3120  3122  3131  3133
        3141  3143  3162  3182  3187  3202  3203  3208  3222  3223  3243  3291  3319  3409  3411
        3416  3426  3445  3499  3506  3746  3754  3793
BVC      801   866   875   884   894   902   916   994  1081  1088  1101  1108  1131  1144  1195
        1201  1213  1233  1245  1255  1266  1300  1348  1354  1402  1469  1491  1559  1565  1579
        1593  1603  1610  1663  1724  1746  1753  1768  1774  1821  1829  1836  1855  1885  2006
        2013  2051  2057  2069  2202  2339  2357  2366  2399  2405  2598  2647  2661  2759  2851
        2863  2871  3072  3210  3245  3249  3268  3272  3276  3281  3286  3667
BVS      782   829   839   849   858   924   931  1095  1116  1123  1151  1158  1166  1190  1207
        1219  1226  1240  1272  1278  1285  1291  1306  1336  1378  1385  1392  1407  1411  1438
        1453  1461  1477  1484  1497  1503  1525  1572  1586  1651  1670  1682  1711  1718  1731
        1739  1762  1794  1806  1816  1842  1849  1861  1867  1873  1891  1940  1948  1955  1962
        1970  1977  1990  2001  2029  2037  2044  2063  2076  2082  2089  2105  2113  2127  2151
        2164  2173  2274  2282  2291  2300  2314  2320  2327  2346  2391  2416  2503  2510  2522
        2532  2543  2550  2564  2605  2639  2656  2857  3165  3290  3682  3691  3769
CCC      780   845   881   960  1005  1070  1077  1352  1466  1551  1563  1607  1799  1952  2010
        2123  2155  2403  2499  2659  2756  3274  3600
CLC      705  1230  1237  1345  1435  1660  1945  1966  2048  2073  2296  3679
CLN     1155  1569  1981  2279  2529  3068  3552  3679
CLR      478   500   603   615   697   730   731   732   733   734   735   736   741   742   763
         827   938   969  1002  1061  1183  1330  1370  1432  1545  1630  1704  1789  1924  2022
        2142  2160  2230  2265  2267  2332  2382  2384  2409  2560  2632  2633  2737  2891  2892
        2953  2959  2960  3077  3097  3168  3176  3241  3255  3301  3333  3357  3363  3364  3385
        3414  3432  3433  3488  3544  3545  3565  3569  3582  3615  3734  3802  3805  3806  3833
        3851
CLRB    1295  2304  2447  2449  2451  2453  2987  3790  3796
CLRD    3044
CLRF    3028
CLV      872   891  1192  1270  1750  1777  1826  3156  3278  3283  3288
CLZ      826   836  1333  1791  1932  2117  2170  2288  2296  2519  2609  2636  3575
CMP      557   560   562   565   954  1039  1319  1416  1419  1531  1534  1707  1781  2125  2136
        2182  2207  2208  2215  2217  2226  2232  2237  2247  2280  2325  2508  2556  2574  2623
        2625  2655  2671  2686  2706  2709  2712  2743  2752  2782  2785  2836  2838  2841  2855
        2884  2899  2902  2910  2912  2944  2993  3093  3103  3108  3109  3127  3160  3171  3192
        3195  3204  3215  3228  3229  3404  3458  3463  3503  3572  3617  3619  3626  3632  3634
        3640  3651  3668  3680  3827
CMPB     583   702   712  2404  2432  2474  2477  2483  2571  2596  2723  2725  2810  2815  2894
        2979  3323  3326  3366  3405  3483  3647  3649  3689  3750
COM      847   974  1003  1149  1340  1591  1716  1988  2179  2181  2187  2197  2269  2297  2336
```

```
        2362  2527  2539  2555  2561  2658  2670  2777  2781  3088  3101  3121  3132  3142  3188
        3630
COMB    1238  1304  1436  1639  1840  2042  2096  2245  2386  2431  3269
DEC      536   611   616   873   972  1009  1072  1170  1400  1577  1766  1953
DECB     588  1224  1259  1271  1276  1489  1697  1871  2061  2401  2485  2977  3280
EMT      419  3265
HALT     436   472   624   670  3370
INC      480   882   964   981  1011  1013  1026  1114  1182  1353  1431  1596  1614  1634  1751
        2005  2020  2143  2193  2270  2385  2691  2696  2796  2947  2951  2978  3014  3090  3106
        3124  3135  3146  3185  3505  3604  3677  3757  3786  3799  3825  3853
INCB    1188  1193  1314  1473  1524  1655  1811  1865  2055  2093  2094  2252  2394  2473  2973
        3284  3435  3623
IOT     3242  3305  3439
JMP      484  3066  3086  3105  3123  3134  3144  3829  3867
JSR      482   582   590   601   618   632   634   637   641   642   644   646   649   650   657
         658   660   663   666   714   752   754   757   759  3157  3184  3206  3225  3368  3441
        3752  3816  3858  3863
MFPD    3026  3042
MOV      441   442   443   447   448   449   450   451   452   453   454   459   460   461   462
         463   464   465   466   471   475   477   493   499   501   527   528   539   540   546
         547   548   550   553   554   559   567   568   572   573   579   586   602   60-   607
         613   628   630   633   635   636   639   648   652   696   737   740   746   7-3   756
         761   764   765   766   767   768   769   772   774   775   777   778   941   943   945
         947   949   968   978   992  1004  1008  1020  1021  1023  1027  1028  1029  1030  1031
        1032  1034  1035  1037  1044  1045  1046  1047  1048  1049  1052  1053  1059  1179  1181
        1327  1329  1427  1429  1430  1543  1546  1548  1549  1626  1629  1633  1635  1636  1637
        1706  1787  1898  1900  1907  1909  1910  1912  1913  1921  1925  1926  1929  1930  2023
        2103  2109  2118  2132  2133  2147  2154  2168  2178  2186  2188  2196  2206  2212  2214
        2223  2224  2263  2266  2307  2335  2350  2363  2380  2383  2445  2448  2450  2452  2496
        2497  2498  2537  2538  2569  2570  2582  2586  2621  2622  2624  2626  2627  2628  2630
        2631  2685  2687  2688  2690  2692  2693  2695  2697  2700  2735  2736  2738  2746  2770
        2771  2793  2794  2795  2797  2799  2801  2835  2837  2839  2840  2842  2845  2846  2893
        2919  2921  2927  2929  2930  2932  2933  2943  2945  2946  2949  2950  2954  2955  2956
        2961  3008  3009  3013  3015  3017  3056  3057  3058  3063  3078  3082  3083  3085  3098
        3099  3113  3117  3119  3139  3152  3153  3169  3175  3177  3179  3180  3190  3199  3200
        3201  3213  3224  3237  3240  3254  3259  3260  3263  3300  3306  3307  3310  3330  3331
        3332  3337  3339  3343  3349  3350  3359  3375  3377  3378  3390  3381  3384  3386  3388
        3389  3391  3392  3393  3394  3398  3399  3400  3408  3410  3412  3420  3421  3422  3430
        3431  3453  3457  3468  3469  3471  3474  3475  3479  3480  3481  3489  3491  3493  3494
        3500  3538  3543  3546  3550  3551  3557  3562  3568  3571  3577  3583  3584  3586  3591
        3592  3594  3599  3606  3607  3610  3613  3614  3644  3646  3654  3655  3661  3663  3675
        3676  3726  3728  3735  3738  3739  3740  3764  3765  3766  3781  3785  3789  3794  3795
        3797  3807  3809  3815  3817  3818  3819  3820  3821  3823  3824  3830  3834  3835  3838
        3839  3840  3860
MOVB     532   576   595   614   700   704   711   961   985  2236  2246  2389  2460  2462  2464
        2489  2584  2585  2587  2588  2701  2702  2703  2769  2800  2967  2998  3312  3556  3593
        3601  3616  3741  3755  3780  3826
NEG      892   928  1164  1365  1584  1729  1975  2526  2651
NEGB    1217  1501  1696  1847  2080  3048  3275
NOP      502   521  1901  2904  2909  2922  3340  3700  3701  3702  3703  3704  3705  3706  3707
        3708  3709  3710  3711  3712  3713  3714  3715  3716  3717  3718  3719  3720  3721  3722
        3723  3729  3804  3864  3865  3866
RESET   3855  3862
ROL      606   610   908   940   962  1086  1093  1390  1608  1709  2011
ROLB    1199  1459  1680  1853  1883  2087  2459  2461  2463  2465  2466  2476  2991  3244
ROR      706   864   971  1006  1064  1106  1346  1375  1557  1737  1968  2199  2334  2411  3602
```

```
RORB     1231  1264  1451  1661  1834  2049  2400  2402  2482  3001  3271
RTI       493   503   533   625   671   727  3054  3246  3293  3328  3343  3396  3449  3454  3473
         3502  3548  3568  3588  3596  3657  3748  3759  3764  3800  3808  3841
RTS       580   596   619   717  3163  3189  3218  3233  3451
SBC       922  1121  1129  1396  1615  1744  1938  3353  3356
SBCB     1205  1316  1495  1691  1804  2027
SCC       825   835   871   890   921   939  1223  1282  1332  1434  1458  1474  1523  1576  1679
         1708  1728  1736  1790  1931  1937  1944  1998  2034  2102  2148  2161  2169  2278  2287
         2295  2343  2388  2518  2528  2635  3019  3065  3155  3205  3678  3688
SEC       494   808   855   863   899   970  1063  1071  1078  1113  1128  1136  1141  1185  1263
         1275  1315  1359  1374  1376  1450  1517  1556  1690  1759  1810  1833  2026  2198  2311
         2333  2410  2413  2457  2602  2664  2699  2798  2844  3239  3264  3279  3311  3351  3354
SEN       792  2124  2306  2413  3232  3309
SEV       800   846  1163  1382  1389  1450  1481  1590  1648  1846  1959  1967  1974  1987  2026
         2086  2110  2124  2271  2311  2324  2413  2500  2507  2540  2547  3262  3767
SEZ       815   846  1481  3264  3313
SUB       529   549   656   745   748   776  1060  1180  1328  1428  15[illegible]  1911  2111  2298  2337
         2344  2520  2562  2637  2645  2764  2772  2861  2907  2913  29[illegible]  3020  3032  3038  3379
         3664
SWAB      929  1283  1310  1360  1518  1686  1778  1878  1982  1994  2305  2310  2997
TRAP      418  3314  3335
TST       495   551   578   622   640   653   654   668   716   743   773   837   988   997  1156
         1334  1547  1552  1627  1628  1631  1632  1889  1908  1922  1923  1927  1928  1933  2019
         2021  2253  2264  2371  2381  2446  2583  2629  2689  2694  2928  2948  3003  3012  3040
         3081  3092  3104  3116  3126  3158  3194  3217  3251  3297  3347  3352  3376  3403  3553
         3559  3624
TSTB      593   698  1289  1507  1512  1644  1792  1800  2438  2897  3355  3461  3742  3783  3813
         3849  3856
WAIT     3745
.ABS      340
.ASCII    680   681
.ASCIZ    486   678   682   683   684   685   689   690  3760  3891  3894  3897  3907
.END     3910
.EVEN     691  3761
.LIST     339   436
.MACR     422   423   424   425   426   427   428   429   430   432   433   434   435
.MACRO    421   430
.NLIST    338   436
.REM        1
.REPT     436  3700
.TITLE    341   728
.WORD     479   506   509   515   516   629   631   747   755   760  1022  1024  1036  1058  1178
         1326  1426  1541  1542  1623  1624  1625  1705  1896  1917  1918  1919  2261  2262  2378
         2379  2443  2444  2580  2581  2617  2618  2619  2620  2679  2679  2680  2681  2682  2683
         2732  2733  2825  2826  2830  2831  2832  2833  2936  2937  2938  2939  2940  2941  3011
         3080  3115  3145  3183  3360  3459  3470  3495  3585  3662  3758  3782  3787  3852  3888
         3889
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*,DZQKCF.SEQ/SOL/CRF=DZQKCF.DOC,DZQKCF.P11/DS:ERFZ
RUN-TIME: 12 21 6 SECONDS

REAL-TIME RATIO: 157.4/62.8
CORE USED: 11K (72 PAGES)

Spooler runtime 13 Seconds, 389 KCS, 333 disk reads, 3 disk writes, 92 pages