

IDENTIFICATION

PRODUCT CODE	MAINDEC-11-DZQKC-D-D
PRODUCT NAME	11 FAMILY INSTRUCTION EXERCISER
DATE CREATED	JULY 21, 1974
MAINTAINER	DIAGNOSTIC GROUP
AUTHOR	J. ADAMS

COPYRIGHT (c) 1973, 1974
DIGITAL EQUIPMENT CORPORATION

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

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



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 (FILL'S) A 4400.

LOAD ADDRESS = 200

PRESS START

SET OPERATING SWITCHES

PASS COUNT IS PRINTED AFTER EACH PASS (SEE SEC 6,4)

"DZQKC 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 (ERROR), 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 DETECTION

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

0000000000000000FIRST ADDRESS TO BE RELOCATED0000000000

CODE TO BE MOVED AND EXECUTED

0000000000000000LAST ADDRESS OF CODE TO BE RELOCATED 00000000

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 20K, 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
.ASS
.TITLE FRONT END
;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS

;GENERAL REGISTER ASSIGNMENTS

000000 R0=X0
000001 R1=X1
000002 R2=X2
000003 R3=X3
000004 R4=X4
000005 R5=X5
000006 SP=X6
000007 PC=X7
000008 R0=X0
000001 R11=X1
000002 R12=X2
000003 R13=X3
000004 R14=X4
000005 R15=X5

;STATUS REGISTER (PSW) BIT ASSIGNMENTS

000001 C=1 ;C BIT
000002 V=2 ;V BIT
000004 Z=4 ;Z BIT
000010 N=0 ;N BIT
000020 T=0 ;T BIT
000300 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 YBITVEC=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 HMVEC=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 TKR= 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 SHR= 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 DUFFER

;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
240000 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

000200 ;=200
00200 012707 002066 MOV #START,PC ;GO TO START OF TEST
00204 012707 002160 MOV #START1,PC ;GO GET LOWER/UPPER RELOCATION BOUNDARY
00220 012707 002224 MOV #START3,PC ;START WITH LAST TYPED BOUNDARY LIMITS

000214 012667 000010 ;ROUTINE TO SAVE REGISTERS ON THE STACK
;CALLED BY SAVE MACRO OR JBR PC,SSAVR
000220 010546 SSAVR1 MOV (S)+,IS ;SAVE RETURN PC
000222 010446 MOV X5[=SP)
MOV X4[=SP)

```

000224 010346      MOV      X37,(SP)
000226 010246      MOV      X42,(SP)
000230 010146      MOV      X14,(SP)
000232 010046      MOV      X0,(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 MAGRO OR JSR PC,SRESTR
000240 012667 000016  SRESTR1 MOV      (SP)+,X5      ;SAVE RETURN PC
000244 012600      MOV      (S2)+,X0
000246 012601      MOV      (S2)+,X1
000250 012602      MOV      (S2)+,X2
000252 012603      MOV      (S2)+,X3
000254 012604      MOV      (S2)+,X4
000256 012605      MOV      (S2)+,X5
000260 012707      MOV      (PC)+,PC      ;RETURN
000262 000000      1$|      0      ;CONTAINS RETURN ADDRESS

000610      ;=610
000610 012737 000620 000024 ;POWER FAIL SUBROUTINE
000616 000000      PDWN1  MOV      #PUP,#PFVEC
000616 000000      PDWN1  HALT

000620 012737 000610 000024 ;POWER UP SUBROUTINE
000626 012706 000600  PDUP1  MOV      #PDWN,#PFVEC      ;RESTORE POWER FAIL TRAP TO POWER
000632 005027      PDUP1  CLR      #KPTR,SP      ;DOWN ROUTINE ABOVE
000634 000000      PDUP1  CLR      (PC)+      ;SET STACK PTR
000636 005267 177772  1$|      ;WORD 0      ;KILL TIME
000636 005267 2$|      INC      1$
000642 001375      BNE     2$
000644 004767 000362  JSR     PC7,PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
000650 000636      PFAIL  PFAIL
000652 000137 002066      JMP     #START      ;RESTART TEST

000656 005015 047520 042527 PFAIL1 ;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 FORXOR1 MOV      #RT1,#ERRVEC+2      ;SET TIME OUT TRAP TO RETURN
000746 000261      FORXOR1 SEC
000750 005737 177060      FORXOR1 TST      #377060      ;IF A TIME OUT OCCURS THEN WHEN NEXT
;INSTRUCTION IS EXECUTED 'G' WILL BE SET
;AND IF NO TIME OUT 'G' WILL BE CLEARED
;BRANCH IF 'G' SET (TIMED OUT)
000754 103401      BCS     1$      ;ADDRESS OF NEXT SUBTEST TO R1
000756 011601      MOV      (SP),R1      ;RESTORE TIME OUT TRAP
000760 005037 000006  1$|      CLR      #ERRVEC+2
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 000000      TICKS1 ;=776
000776 000000      TICKS1 ;WORD 0      ;CONTAINS CLOCK TICK COUNT
001000 000000      ICNT1  ;=1000
001002 000000      SFILLS ;WORD 0      ;CONTAINS PASS COUNT
;CONTAINS FILLS COUNT IN ODD BYTE
;AND FILLER CHARACTER IN EVEN BYTE
001004 000000      FACTOR1 ;CONTAINS RELOCATION FACTOR
;SUBTRACT # IN FACTOR FROM PC TO GET PC OF ORIGINAL CODE
001006 000000      RELR11 ;CONTAINS RELOCATED R1 (THE R1 OF THE
;ORIGINAL CODE MOVED)
001010 000000      FRSTAD1 ;WORD 0      ;CONTAINS FIRST ADRS OF CODE TO BE MOVED
001012 000000      FRSTMEM1 ;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      SCOPEA1 NOP
001020 032706 004000 000002      SCOPEA1 BIT      #4000,2(SP)      ;HAS REGISTER SET BIT SET ON TRAP
001026 001403      SCOPEA1 BEQ     2$      ;BRANCH IF NOT
001030 052737 004000 177776      SCOPEA1 BIT      #4000,#PSW      ;RETAIN REGISTER SET
001036 032737 040000 177570 2$|      BIT      #4000,#SWR      ;CHECK BIT 14 (CONTINUOUS LOOP)
001044 001416      SCOPEA1 BEQ     SCOPEC
001046 010116      SCOPEB1 MOV      R1,(SP)      ;LOAD RETURN ADDRESS
001050 010137 001006      SCOPEB1 MOV      R1,#REL1      ;LOAD REL1
001054 163737 001004 001006      SCOPEB1 SUB      #FACTOR,#REL1      ;REL1 CONTAINS UNRELOCATED R1
001062 032737 004000 177570      SCOPEB1 BIT      #400,#SWR      ;LOAD PDP11/45 MICRO BREAK REG?
001070 001403      SCOPEB1 BEQ     1$
001072 113737 177570 177770      SCOPEB1 MOVB   #SWR,#UBREAK      ;LOAD MICRO BREAK REG WITH SR0=7
001080 000002      SCOPEB1 RTI      ;RETURN TO SUBTEST
001102 032737 004000 177570      SCOPEC1 BIT      #4000,#SWR      ;SUBTEST ITERATION DESIRED?
001110 001006      SCOPEC1 BNE     SCOPEE      ;BRANCH IF NO ITERATION DESIRED?
001112 005327      SCOPEC1 DEC      (PC)+      ;DECREMENT SUBTEST ITERATION COUNT
001114 000040      SCOPEE1 ;CONTAINS SUBTEST ITERATION COUNT
001116 001353      SCOPEE1 BNE     SCOPEB
001120 012767 000040 177766      SCOPEE1 MOV      #40,SCOPEE      ;RESET ITERATION COUNT
001126 011601      SCOPEE1 MOV      (SR),R1      ;GET ADDRESS OF NEXT TEST
001130 000746      SCOPEE1 BR      SCOPEB

;ROUTINE TO RELOCATE PROGRAM CODE
001132 032737 010000 177570      RELOC1 BIT      #10000,#SWR      ;CHECK IF RELOCATION DESIRED (BIT12)
001140 001031      RELOC1 BNE     3$      ;BRANCH IF NO RELOCATION DESIRED
001142 013700 001010      RELOC1 MOV      #FRSTAD,R0      ;GET FIRST ADDRESS OF CODE TO BE MOVED
001146 010005      RELOC1 MOV      #0,R0      ;SAVE
001150 010204      RELOC1 MOV      R2,R4      ;GET LAST ADDRESS OF CODE TO BE MOVED
001152 160904      RELOC1 SUB      #R4,R4      ;R4 CONTAINS # OF WORDS TO RELOCATE
001154 010203      RELOC1 MOV      R2,R3      ;SAVE LAST ADDRESS OF CODE TO BE MOVED
001156 005737 001004      RELOC1 TST      #FACTOR      ;FIRST RELOCATION IS TO 20000
001162 001004      RELOC1 BNE     1$
001164 010237 001230      RELOC1 MOV      #R2,#RETPC      ;SAVE RETURN PC TO NEXT SECTION OF CODE
001170 013702 001012      RELOC1 MOV      #FRSTMEM,R2      ;SET FIRST ADDRESS
001174 000204      RELOC1 ADD      R2,R4      ;R4 CONTAINS LAST MEMORY ADDRESS
;TO BE USED
001176 020437 002140      RELOC1 CMP      R4,#LSTMEM      ;CHECK IF SUFFICIENT MEMORY REMAINS

```

```

001202 001011      BHI    4S
001204 012022      MOV    (R0)+,(R2)+ ;RELOCATE PROGRAM CODE
001206 020003      CMP    R0,R3 ;CHECK IF DONE
001210 001375      BNE   1S
001212 024042      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   2S
001224 010207      MOV    R2,PC ;GO EXECUTE RELOCATED CODE
001226 011727      MOV    (PC),PC ;RETURN TO NEXT SECTION OF CODE
001230 000000      RETPC ;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 017000      MOV    #2(SP),R0 ;GET MESSAGE ADDRESS
001240 062766      ADD    #2(2(SP),R0 ;ADJUST RETURN PC

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

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

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

001312 105737      177564      5S:   TSTB  #*TPS ;WAIT FOR OUTPUT DEVICE
001316 100375      BPL   ,+4 ;TO BECOME READY
001320 116637      000002      177566      MOV    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      SFORM0

001330 004767      176660      JSR    PC,$$SAVR ;GO SAVE REGISTERS ON THE STACK
001334 012774      001662      MOV    #DIGITS,R4 ;ADDRESS WHERE ASCII VALUES ARE STORED
001340 005003      CLR   R3 ;WORKING & INDEX REGISTER
001342 010201      MOV    R2,R1 ;SAVE
001344 006302      1S:   ASL   R2 ;FIRST DIGIT TO R3
001346 006103      ROL   R3
001350 012700      000006      MOV    #6,R0 ;DIGIT COUNT
001354 000404      BR    3S ;PRINT FIRST DIGIT
001356 006302      2S:   ASL   R2
001360 006103      ROL   R3
001362 005301      DEC   R1

```

```

001364 001374      000003      3S:   BNE   2S
001366 012701      MOV    #3,R1 ;DIGIT SHIFT COUNT
001372 116324      001652      MOV    DIGTAB(3),(4)+ ;LOAD DIGIT INTO MESSAGE
001376 005003      CLR   R3 ;CLEAR INDEX
001400 005300      DEC   R0 ;DEC DIGIT COUNT
001402 001365      BNE   2S
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   ,+4
001420 000000      HALT
001422 000002      RTI
001424 032737      020000      177570      BIT   #2000,*#SWR ;PRINT OUT DESIRED?
001432 001073      BNE   1S ;BRANCH IF NO PRINTOUT
001434 011627      MOV    (SP),(PC)+ ;SAVE PC
001436 000000      11S:  ,WORD 0 ;CONTAINS SAVED PC
001440 016627      000002      MOV    2(SP),(PC)+ ;GET STATUS ON TRAP
001444 000000      12S:  ,WORD 0 ;CONTAINS STATUS (PSW) AT TIME OF TRAP
001446 004767      176542      JSR    PC,$$SAVR ;GO SAVE REGISTERS ON THE STACK
001452 013702      001000      MOV    #*ICNT,R2 ;GET PASS COUNT
001456 004767      177646      JSR    PC,$$FORM0 ;GO TO FORMAT ROUTINE
001462 016767      000176      002212      MOV    DIGITS+2,PASSES ;LOAD ASCII VALUES
001470 016767      000172      002206      MOV    DIGITS+4,PASSES+2
001476 004767      177530      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001502 001672      PASCNT
001504 016702      177726      MOV    11S,R2 ;GET PC OF ERROR CALL
001510 005742      TST   =(R2) ;DECREMENT PC TO HLT
001512 004767      177612      JSR    PC,$$FORM0 ;GO TO FORMAT ROUTINE
001516 004767      177510      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001522 001707      ERRPC
001524 004767      177502      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001530 001662      DIGITS
001532 004767      177474      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001536 001714      STATUS
001540 016702      177700      MOV    12S,R2 ;GET STAUUS AT TIME OF ERROR
001544 004767      177560      JSR    PC,$$FORM0 ;GO TO FORMAT ROUTINE
001550 004767      177456      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001554 001662      DIGITS
001556 016702      177654      MOV    11S,R2 ;GET PC OF ERROR
001562 005742      TST   =(R2)
001564 005737      001004      TST   #*FACTOR
001570 001412      BEQ   10S
001572 163702      001004      SUB   #*FACTOR,R2 ;FORM PC OF ORIGINAL CODE
001576 004767      177526      JSR    PC,$$FORM0 ;GO TO FORMAT ROUTINE
001602 004767      177424      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001606 001721      ERRPC0
001610 004767      177416      JSR    PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001614 001662      DIGITS
001616      10S:
001616 004767      176416      JSR    PC,$$RESTR ;RESTORE REGISTERS FROM STACK
001622 032737      002000      177570      1S:   BIT   #2000,*#SWR ;RING BELL ON ERROR
001630 001403      BEQ   2S

```

```

001632 004747 177374 JSR PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
001636 001747 BELL
001640 005737 177570 2s) TST @NSHR ;HALT AFTER PRINT OUT
001644 100001 BPL ,+3
001646 000000 HALT
001650 000002 RTI
    
```

```

;DIGIT TABLE
001652 030460 DIGTAB1 "01
001654 031462 "23
001656 032464 "45
001660 033466 "67
001662 030060 030060 030060 DIGITS1 ,ASCIZ '000000 '
001670 000040
001672 005015
001674 044440 047103 031524 PASCNT1 ,ASCII <15><12>
001702 030060 030060 000 PASSE1 ,ASCII / 'CNTs'
001707 040 041520 000075 ERRPC1 ,ASCIZ / 'C='
001714 051520 030527 000 STATUS1 ,ASCIZ / 'PSW='
001721 120 020103 042522 ERRPC01 ,ASCIZ 'PC RELOCATED FROM '
001726 047514 040503 042524
001734 020104 031106 040517
001742 000040
001744 005015 000 SCRLF1 ,ASCIZ <15><12>
001747 007 000 BELLI ,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
RECD1 MOV R0,=(SP) ;SAVE R0 ON THE STACK
CLR (R5) ;CLEAR OLD DATA
1s) TSTB @#TKS ;WAIT FOR USER TO TYPE CHARACTER
BPL 15
MOVVB @#TKB,R0 ;GET CHARACTER
BIC #250,R0 ;STRIP MSB
CMPB #177,R0 ;CHECK IF RUBOUT
BNE 25 ;BRANCH IF NOT RUBOUT
MOVVB #1,R0#TPB ;TYPE \
CLC ;CLEAR CARRY
ROR (R5) ;SHIFT LAST TYPED CHARACTER
ASR (R5) ;OUT OF DATA WORD
ASR (R5)
BR 15 ;GO WAIT FOR NEXT CHARACTER

002022 110037 177566 2s) MOVVB R0,#TPB ;ECHO CHARACTER TYPED
002026 122700 000015 CMPB #15,R0 ;CHECK IF CARRIAGE RETURN
002032 001005 BNE 35 ;BRANCH IF NOT CARRIAGE RETURN
002034 004747 177172 JSR PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
002040 001744 SCRLF
002042 005725 TST (R5)+ ;STEP RETURN ADDRESS
002044 000205 RTS ;RETURN

002046 042700 177770 3s) BIC #177770,R0 ;STRIP NONESSENTIAL 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 15 ;WAIT FOR NEXT CHARACTER

002064 000002 RTI ;RETURN
    
```

TITLE DEQKC=C BASIC 11 FAMILY INSTRUCTION EXER,

```

002066 005037 177776 START CLR #PSM ;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 000000 MOV #KPTR,SP ;SET KERNEL STACK PTR

;ROUTINE TO DETERMINE LAST MEMORY ADDRESS
002112 012737 002132 000004 MOV #15,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 102700 000002 1S: SUB #2,R0
002136 010027 MOV R0,(PC)+ ;SET VALUE INTO LSTMEM
002140 000000 LSTMEM,WORD 0 ;CONTAINS VALUE OF LAST MEMORY ADDRESS
002142 102737 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 004767 177046 START1: JSR PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
002164 016574 MSG1
002166 004567 177560 JSR R5,RECD ;GET LOWER LIMIT
002172 000000 1S: ,WORD 0 ;CONTAINS TYPED LOWER LIMIT
002174 016737 177772 001012 MOV 15,#FRSTMEM ;SET IN LOWER LIMIT
002202 004767 177024 JSR PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
002206 016611 MSG2
002210 004567 177536 2S: JSR R5,RECD ;GET UPPER LIMIT
002214 000000 2S: ,WORD 0 ;CONTAINS UPPER LIMIT
002216 016737 177772 002140 MOV 25,#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 #SIPTR,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 #EMROR,#TRAPVEC ;SET TRAP (HLT) VECTOR
002264 012737 000200 000036 MOV #25,#TRAPVEC+2 ;PRIORITY LEVEL 4 ON TRAP

;0000000000000000 FIRST ADDRESS TO BE RELOCATED 0000000000
002272 010700 REL0: MOV PC,R0 ;GET PC
002274 005740 TST =(R0) ;R0 CONTAINS THE ADDRESS OF REL0
002276 010037 001010 MOV R0,#FRSTAD ;SAVE
002302 010700 MOV PC,R0 ;GET CURRENT PC
002304 102700 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 CC0 ;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 CC1
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 CC2
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 CC1
002414 003401 BLE ,+4
002416 104400 CC4: HLT ;ERROR! ONE OF THE ABOVE BRANCHES FAILED
002420 104000 SCOPE

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

002444 000277 SCC

```

002446	000244	CLZ			
002450	000700	TSY	R0		IR0=0,CC'S=0100
002452	103404	BCC	TSY0		
002454	102403	BVS	TSY0		
002456	001002	BNE	TSY0		
002460	100401	BMI	TSY0		
002462	101401	BLOS	,+4		
002464	104400	HLT			ERROR! INCORRECT CC'S AFTER TSY
002466	000257	CCC			
002470	000266	+SEZ:SEV			
002472	000100	COM	R0		IR0=1,CC'S=1001
002474	103004	BCC	COM0		
002476	102403	BVS	COM0		
002500	001402	BEQ	COM0		
002502	100001	BPL	COM0		
002504	002401	BLT	,+4		
002506	104400	HLT			ERROR! INCORRECT CC'S AFTER COM
002510	000261	SEC			
002512	000300	ADC	R0		IR0=00000,CC'S=0101
002514	103003	BCC	ADC0		
002516	102402	BVS	ADC0		
002520	001001	BNE	ADC0		
002522	002001	BGE	,+4		
002524	104400	HLT			ERROR! INCORRECT CC'S AFTER ADC
002526	000261	SEC			
002530	000000	ROR	R0		IR0=10000,CC'S=1010
002532	103404	BCC	ROR0		
002534	102003	BVC	ROR0		
002536	001402	BEQ	ROR0		
002540	100001	BPL	ROR0		
002542	003001	BGT	,+4		
002544	104400	HLT			ERROR! INCORRECT CC'S AFTER ROR
002546	000277	SCC			
002550	000242	CLV			
002552	000300	DEC	R0		IR0=07777,CC'S=0011
002554	103004	BCC	DEC0		
002556	102003	BVC	DEC0		
002560	001402	BEQ	DEC0		
002562	100401	BMI	DEC0		
002564	003401	BLE	,+4		
002566	104400	HLT			ERROR! INCORRECT CC'S AFTER DEC
002570	000257	CCC			
002572	000200	INC	R0		IR0=10000,CC'S=1010
002574	103404	BCC	INC0		
002576	102003	BVC	INC0		
002600	001402	BEQ	INC0		
002602	100001	BPL	INC0		
002604	003001	BGT	,+4		
002606	104400	HLT			ERROR! INCORRECT CC'S AFTER INC

002610	000277	SCC			
002612	000242	CLV			
002614	000400	NEG	R0		IR0=10000,CC'S=1011
002616	103003	BCC	NEG0		
002620	102002	BVC	NEG0		
002622	001401	BEQ	NEG0		
002624	002001	BGE	,+4		
002626	104400	HLT			ERROR! INCORRECT CC'S AFTER NEG
002630	000261	SEC			
002632	000300	ASL	R0		IR0=00000,CC'S=0111
002634	103004	BCC	ASL0		
002636	102003	BVC	ASL0		
002640	001002	BNE	ASL0		
002642	100401	BMI	ASL0		
002644	101401	BLOS	,+4		
002646	104400	HLT			ERROR! INCORRECT CC'S AFTER ASL
002650	000100	ROL	R0		IR0=00001,CC'S=0000
002652	103402	BCC	ROL0		
002654	003401	BLE	ROL0		
002656	002001	BGE	,+4		
002660	104400	HLT			ERROR! INCORRECT CC'S AFTER ROL
002662	000200	ASR	R0		IR0=00000,CC'S=0111
002664	103003	BCC	ASR0		
002666	102002	BVC	ASR0		
002670	001001	BNE	ASR0		
002672	002401	BLT	,+4		
002674	104400	HLT			ERROR! INCORRECT CC'S AFTER ASR
002676	000277	SCC			
002700	000300	SBC	R0		IR0=1,CC'S=1001
002702	103002	BCC	SBC0		
002704	102401	BVS	SBC0		
002706	003401	BLE	,+4		
002710	104400	HLT			ERROR! INCORRECT CC'S AFTER SBC
002712	000400	NEG	R0		IR0=00001,CC'S=00001
002714	000300	SWAB	R0		IR0=000400,CC'S=0100
002716	103403	BCC	SWAB0		
002720	102402	BVS	SWAB0		
002722	001001	BNE	SWAB0		
002724	002001	BGE	,+4		
002726	104400	HLT			ERROR! INCORRECT CC'S AFTER SWAB
002730	104000	SCOPE			
002732	000000	ICHECK REGISTER SELECTION			
002734	000277	CLR	R0		
002736	000100	SCC			IR0=1
002740	010002	ROL	R0		IR2=2
002742	000302	MOV	R2		IR2=2
002744	010203	MOV	R2;R3		

```

002746 006303 ASL R3 JR3=4
002750 010304 MOV R3,R4
002752 006304 ASL R4 JR4=10
002754 010405 MOV R4,R5
002756 006305 ASL R5 JR5=20
002760 010546 MOV R5,(SP) ;SET BITS SET IN REGISTERS
002762 050416 B18 R4,(SP) ;INTO STACK ADDRESS
002764 050316 B18 R3,(SP)
002766 050216 B18 R2,(SP)
002770 050016 B18 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 1S1 ROL R0 ;ROTATE A 0 THROUGH ALL BIT
003012 103776 BCS 15 ;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 2S1 SEC
003032 006002 ROR R2 ;ROTATE I THROUGH ALL BIT POSITS
003034 005300 DEC R0 ;DECREMENT SHIFT COUNT
003036 001374 BNE 25
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 ASR R3 ;EXTEND I BIT THROUGH ALL POSITIONS
003054 103376 BCC 35
003056 005203 INC R3
003060 001401 BEQ ,+4
003062 104400 HLT ;ERROR!

003064 112704 177401 4S1 MOVB #177401,R4
003070 000404 ADD R4,R4 ;R4=1
003072 103376 BCC 45 ;HAS THE AFFECT OF SHIFTING A BIT
003074 005704 TST R4 ;THROUGH ALL POSITIONS
003076 001401 BEQ ,+4 ;RESULT SHOULD BE 0
003100 104400 HLT

003102 012705 000001 5S1 MOV #17,R5
003106 006305 ASL R5
003110 102376 BVC 55
003112 006305 ASL R5
003114 103002 BCC 65
003116 005705 TST R5

```

```

003120 001401 BEQ ,+4
003122 104400 HLT

;CHECK REGISTER VOLITILITY
003124 005002 CLR R2
003126 005102 COM R2 JR2=-1
003130 010203 MOV R2,R3
003132 000257 CCC
003134 006002 ROR R2 JR2=LOOP COUNT
003136 006202 ASR R2
003140 010304 7S1 MOV R3,R4
003142 005302 DEC R2 ;DECREMENT LOOP COUNT
003144 001375 BNE 75
003146 005203 INC R3 ;CHECK R3
003150 001002 BNE 85
003152 005204 INC R4 ;CHECK R4
003154 001401 BEQ ,+4
003156 104400 HLT

;CHECK TRANSFER OF REGISTER DATA BETWEEN THE CS AND GD REGISTERS (11/45)
003160 032737 000020 177776 1S1 BIT #20,#PSW ;CHECK IF IT' BIT IS SET
003166 001052 BNE 75 ;SKIP TEST IF IT' BIT SET
003170 010146 MOV R1,(SP) ;SAVE SCOPE PTR
003172 010627 MOV SP,(PC)+ ;SAVE STACK PTR
003174 000000 2S1 ,WORD 0 ;CONTAINS SAVED STACK PTR
003176 010727 MOV PC,(PC)+ ;LOAD DATA, THE CURRENT PC IS USED AS
003200 000000 ,WORD 0 ;DATA, IF THIS TEST FAILS RS CON-
;TAINS THE DATA BEING USED;
003202 005267 177772 INC 25 ;MAKE ODD TO CHECK BIT 0
003206 016700 177766 MOV 25,R0 ;LOAD GD REGISTER 0
003212 010001 MOV R0,R1 ;TRANSFER CS REG 0 TO GD REG 1
003214 010102 MOV R1,R2 ;AND CS REG 1 TO GD REG 2
003216 010203 MOV R2,R3 ;ETC.;
003220 010304 MOV R3,R4
003222 010405 MOV R4,R5
003224 102705 000340 177776 B18B #30,#PSW
003232 010506 MOV R5,SP ;SET PRIORITY LEVEL 7
003234 010607 MOV SP,(PC)+ ;TRANSFER CS REG 5 TO GD STK PTR
003236 000000 ,WORD 0 ;TRANSFER CS STK PTR TO MEMORY
003240 016706 177730 MOV 15,SP ;CONTAINS CS STACK PTR
003244 142737 000340 177776 B18B #30,#PSW ;RESTORE STK PTR NEEDED FOR HLT/SCOPE
003252 026707 177760 MOV 45,R0 ;SET PRIORITY LEVEL 0
003256 010004 BNE 55 ;COMPARE CS/GD STKPTR WITH CS REG 0
003260 006307 177714 ASL 25 ;BRANCH IF THEY WERE NOT =
003264 001390 BNE 35 ;SHIFT TEST DATA UNTIL = 000000
003266 000411 BR 65
003270 010946 5S1 MOV R0,(SP) ;GET CS 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 CS STK PTR NOT = CS REG 0
;CS REG 0=GD REG 5 ARE ON THE STACK

```

003306	016706	177662		MOV	157SP	RESTORE STACK PTR
003312	012601		6S	MOV	(SP)+,R1	RESTORE SCOPE PTR
003314	104000		7S	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	102702	000004		SUB	#4,R2	R2 POINTS TO RESERVED WORD
003330	000012			CLR	(R2)	PRESET (R2)
003332	000261			SEC		
003334	000012			ROR	(R2)	(R2)=100000,CC=1010
003336	101402			BLOS	ROR1	
003340	100001			BPL	ROR1	
003342	002001			BGE	,+3	
003344	104400		ROR1	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003346	000257			CCC		
003350	000261			SEC		
003352	000312			DEC	(R2)	(R2)=077777,CC=0011
003354	100001			BCC	DEC1	
003356	003401			BLE	,+3	
003360	104400		DEC1	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003362	000257			CCC		
003364	000261			SEC		
003366	000512			ADC	(R2)	(R2)=100000,CC=1010
003370	103403			BCS	AD1	
003372	102002			BVC	AD1	
003374	100001			BPL	AD1	
003376	001001			BNE	,+4	
003400	104400		AD1	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	,+3	
003426	104400		ROL1A	HLT		
003430	006212			ASR	(R2)	(R2)=000000,CC=0111
003432	100003			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			BEO	ROR1A	
003454	100401			BMI	,+3	
003456	104400		ROR1A	HLT		
003460	000261			SEC		
003462	000212			INC	(R2)	(R2)=100001,CC=1001
003464	100003			BCC	INC1	
003466	102402			BVS	INC1	
003470	001401			BEO	INC1	
003472	100401			BMI	,+3	
003474	104400		INC1	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003476	000612			SBC	(R2)	(R2)=100000,CC=1000
003500	103403			BCS	SBC1	
003502	102402			BVS	SBC1	
003504	001401			BEO	SBC1	
003506	100401			BMI	,+3	
003510	104400		SBC1	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003512	000261			SEC		
003514	000612			SBC	(R2)	(R2)=077777,CC=0010
003516	103403			BCS	SBC1A	
003520	102002			BVC	SBC1A	
003522	001401			BEO	SBC1A	
003524	100001			BPL	,+3	
003526	104400		SBC1A	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003530	000261			SEC		
003532	000512			ADC	(R2)	(R2)=100000,CC=1010
003534	100401			BMI	,+4	
003536	104400			HLT		
003540	000261			SEC		
003542	000312			ASL	(R2)	(R2)=000000,CC=0111
003544	100003			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	000112			COM	(R2)	(R2)=177777,CC=1001
003560	100002			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	000712			TSY	(R2)	(R2)=177777,CC=1000
003574	103403			BCS	TSY1	
003576	102402			BVS	TSY1	
003600	100001			BPL	TSY1	

003602	001001		BNE	,+4	
003604	104400	TS11	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003606	000202		SEV		
003610	005412		NEG	(R2)	(R2)=000001,CC=0000
003612	103002		BCC	NEG1	
003614	102401		BVS	NEG1	
003616	001001		BNE	,+2	
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	,+2	
003630	104400	DEC1A	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
003632	104000		SCOPE		
ICHECK 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	PCTR3	
003642	162703	000004	SUB	#4,R3	R3 POINTS TO EVEN BYTE OF WORD
003644	010304		MOV	R3,R4	R4 POINTS TO ODD BYTE OF WORD
003650	005204		INC	R4	
003652	005013		CLR	(R3)	PRESET DATA
003654	000201	1s1	SEC		
003656	105513		ADCB	(R3)	ADD CARRY TO EVEN BYTE
003660	100402		BMI	2S	UNTIL EVEN BYTE BECOMES NEGATIVE
003662	105214		INCB	(R4)	INCREMENT ODD BYTE
003664	000773		BR	1S	
003666	102401	2s1	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		BVS	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)=177400=[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		BVS	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	,+2	
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	,+2	
004004	104400	RORB1	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
004006	000241		CLC		
004010	105114		COMB	(R4)	(R3)=000197,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	1s1	ASRB	(R3)	SHIFT EVEN BYTE UNTIL V CLEARS
004024	102002		BVC	2S	
004026	105514		ADCB	(R4)	AND ADD CARRY TO ODD BYTE
004030	000774		BR	1S	
004032	103401	2s1	BVS	ASRB1	
004034	001401		BEQ	,+4	
004036	104400	ASRB1	HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
004040	106214		ASRB	(R4)	
004042	106214		ASRB	(R1)	(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	,+2	
004060	104400		HLT		ERROR! INCORRECT CC'S AS SHOWN ABOVE
004062	000201		SEC		
004064	106014		RORB	(R4)	(R3)=100000,CC=1010
004066	103402		BVS	RORB1A	
004070	102001		BVC	RORB1A	
004072	100401		BMI	,+4	

004074	104400	RORB1A	HLT			ERROR: INCORRECT CC'S AS SHOWN ABOVE
004076	000242		CLV			
004100	103314		DECB	(R4)		(R3)=077400,CC=0100
004102	102401		BVS	,+4		
004104	104400		HLT			
004106	000261		SEC			
004110	103313		DECB	(R3)		(R3)=077777,CC=1001
004112	103002		BCC	DELB1A		
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		BVS	SWAB1		
004130	102401		BVS	SWAB1		
004132	100001		BPL	,+4		
004134	104400	SWAB1	HLT			ERROR: INCORRECT CC'S AS SHOWN ABOVE
004136	105714		TSYB	(R4)		(R3)=177577=[1774][177],CC=1000
004140	103402		BVS	TSYB1		
004142	102401		BVS	TSYB1		
004144	100401		BMI	,+4		
004146	104400	TSYB1	HLT			ERROR: INCORRECT CC'S AS SHOWN ABOVE
004150	105014		CLRB	(R4)		(R3)=000177=[0000][177],CC=0100
004152	001401		BEQ	,+3		
004154	104400		HLT			
004156	100313		ASLB	(R3)		(R3)=000376,CC=1010
004160	103402		BVS	ASHB1A		
004162	102001		BVC	ASHB1A		
004164	100401		BMI	,+3		
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	,+2		
004220	104400		HLT			
004222	022713	000400	CMP	#400,(R3)		ICHECK REMAINING RESULT
004226	001401		BEQ	,+4		
004230	104400		HLT			

004232	104000		SCOPE			
004234	000401		ICHECK	UNARY WORD OPS USING ADDRESS MONES 2 AND 4 (AUTO INC/DEC)		
004236	000000		BR	,+2		
004240	010704		,WORD	0		ADDRESS RESERVED FOR TESTS
004242	162704	000004	MOV	PCTR4		
004246	010405		SUB	#44R4		R4 AND R5 POINT TO RESERVED WORD
004250	005015		MOV	R4R5		PRESET DATA=0
			CLR	(R2)		
004252	000277		SCC			
004254	000244		CLZ			
004256	005725		TSY	(R5)+		(R5)=000000,CC=0100
004260	103402		BVS	TSI2		
004262	102401		BVS	TSI2		
004264	001401		BEQ	,+3		
004266	104400	TSY2	HLT			ERROR: INCORRECT CC'S AS SHOWN ABOVE
004270	005145		COM	=(R5)		(R5)=177777,CC=1001
004272	103001		BCC	COM4		
004274	100401		BMI	,+4		
004276	104400	COM4	HLT			ERROR: INCORRECT CC'S AS SHOWN ABOVE
004300	000241		CLC			
004302	000024		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		BVS	SWAB2		
004336	100401		BMI	,+2		
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	005044		CLR	=(R4)		(R4)=000000,CC=0100
004354	001401		BEQ	,+4		
004356	104400		HLT			
004360	000261		SEC			
004362	000045		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	ADC21	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	,+2	
004410	104400	ASR21	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	ROL41	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		BCC	DEC2	
004440	102001		BVC	DEC2	
004442	100001		BPL	,+4	
004444	104400	DEC21	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004446	006324		ASL	(R4)+	;(R4)=177776,CC=1010
004450	102401		BVS	,+2	
004452	104400		HLT		
004454	006344		ASL	=(R4)	;(R4)=177774,CC=1001
004456	103003		BCC	ASL4	
004460	102402		BVS	ASL4	
004462	001401		BEO	ASL4	
004464	100401		BMI	,+2	
004466	104400	ASL41	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004470	022724	177774	CHP	#177774,(R4)+	
004474	001401		BEQ	,+4	
004476	104400		HLT		
004500	020405		CHP	R4,R5	
004502	001401		BEQ	,+2	
004504	104400		HLT		
004506	104000		SCOPE		
004510	000401		BR	,+4	;CHECK UNARY BYTE OPS USING ADDRESS MODES 2 AND 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)+	;(R5)=000397,CC=1001
004540	103002		BCC	COMB2	
004542	102401		BVS	COMB2	
004544	100401		BMI	,+2	
004546	104400	COMB21	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004550	105542		ADCB	=(R2)	;(R2)=000000,CC=0101
004552	001401		BEQ	,+4	
004554	104400		HLT		;ERROR! INCORRECT RESULT AS SHOWN ABOVE
004556	105525		ADCB	(R5)+	;(R5)=000400,CC=0000
004560	103401		BCC	ADCB2	
004562	001001		BNE	,+4	
004564	104400	ADCB21	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004566	000263		+SEC:SEV		
004570	106045		RORB	=(R5)	;(R5)=100000,CC=1001
004572	103003		BCC	RORB4	
004574	102402		BVS	RORB4	
004576	001401		BEQ	RORB4	
004600	100401		BMI	,+2	
004602	104400	RORB41	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004604	000277		SCC		
004606	106122		ROLB	(R2)+	;(R2)=100001,CC=0000
004610	103403		BCC	ROLB2	
004612	102402		BVS	ROLB2	
004614	001401		BEQ	ROLB2	
004616	100001		BPL	,+2	
004620	104400	ROLB21	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004622	000257		CCO		
004624	106225		ASRB	(R5)+	;(R5)=140001, CC=1010
004626	103402		BCC	ASRB2	
004630	102001		BVC	ASRB2	
004632	100401		BMI	,+2	
004634	104400	ASRB21	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004636	105242		INCB	=(R2)	;(R2)=140002,CC=0000
004640	000277		SCC		
004642	106222		ASRB	(R2)+	;(R2)=140001,CC=0000
004644	103402		BCC	ASRB2A	
004646	102401		BVS	ASRB2A	
004650	100001		BPL	,+4	
004652	104400	ASRB2A1	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
004654	000266		+SEC:SEV		;SET Z,V
004656	106345		ASLB	=(R5)	;(R5)=100001,CC=1001
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 SHOWN ABOVE
004704	105645		SBCB	={R3}	{R0}=077400, CC=0100
004706	103402		BOS	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		BOS	TSTB2	
004734	001401		BEQ	,+3	
004736	104400	TSTB2:	HLT		
004740	105722		TSTB	{R2}+	{R0}=100400, CC=1000
004742	001401		BEQ	TSTB2A	
004744	100401		BMI	,+3	
004746	104400	TSTB2A:	HLT		
004750	000261		SEC	={R2}	{R0}=000201, CC=1000
004752	000342		SWAB	SWAB4	
004754	103401		BOS	SWAB4	
004756	100401		BMI	,+2	
004760	104400	SWAB4:	HLT		
004762	000277		SCC	{R5}+	{R0}=000201+{0004}[201], CC=0000
004764	105225		INCB	INCB2	
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	ICHECK END RESULT
005004	001401		BEQ	,+2	
005006	104400		HLT		
005010	020205		CMP	R2,R5	ICHECK REGISTERS
005012	001401		BEQ	,+4	
005014	104400		HLT		
005016	104000		SCOPE		

005020	000402		ICHECK	UNARY WORD OPS USING ADDRESS MODES 3 AND 5	
005022	000000		BR	,+2	RESERVE 2 WORDS
005024	000000		,WORD	0	11 FOR THE ADDRESS
005026	010703		,WORD	0	AND 1 FOR DATA
005030	102703	000004	MOV	PC,R3	
005034	005013		SUB	#4,R3	
005036	010300		CLR	{R2}	PRESET DATA
005040	005743		MOV	R3,R0	R0 POINTS TO DATA WORD
005042	010013		TST	={R3}	
005044	010304		MOV	R0,{R3}	
			MOV	R3,R4	
005046	000257		CCC	={R3}+	{R0}=000000, CC=0100
005050	005733		TST	,+4	
005052	001401		BEQ		
005054	104400		HLT		
005056	000261		SEC	={R3}	{R0}=100000, CC=1010
005060	000003		ROR	ROR5	
005062	103402		BOS	ROR5	
005064	102001		BVC	ROR5	
005066	100401		BMI	,+4	
005070	104400	ROR5:	HLT		
005072	000257		CCC	={R4}+	{R0}=140000, CC=1010
005074	006234		ASR	ASR3	
005076	102001		BVC	ASR3	
005100	100401		BMI	,+2	
005102	104400	ASR3:	HLT		
005104	000250		CLN	={R3}+	{R0}=100000, CC=1001
005106	006333		ASL	ASL3	
005110	103002		BCC	ASL3	
005112	102401		BVS	ASL3	
005114	100401		BMI	,+4	
005116	104400	ASL3:	HLT		
005120	000277		SCC	={R4}	{R0}=077777, CC=0010
005122	005334		DEC	DEC5	
005124	103003		BCC	DEC5	
005126	102002		BVC	DEC5	
005130	001401		BEQ	DEC5	
005132	100001		BPL	,+2	
005134	104400	DEC5:	HLT		
005136	005453		NEB	={R3}	{R0}=100001, CC=1001
005140	103002		BCC	NEB5	
005142	102401		BVS	NEB5	
005144	100401		BMI	,+4	
005146	104400	NEB5:	HLT		
005150	000262		SEV	={R4}+	{R0}=077776, CC=0001
005152	005134		COM		

005154	103001	BCC	COM3		
005156	102001	BVC	,*4		
005160	104400	HLT			
		COM3:			
005162	005233	INC	*(R3)+		;(R0)=077777, CC=0001
005164	103001	BCC	INC3		
005166	100001	SPL	,*4		
005170	104400	HLT			
		INC3:			
005172	005554	ADC	*(R4)		;(R0)=100000, CC=1010
005174	103402	BCS	ADCS		
005176	102001	BVC	ADCS		
005200	100401	BMI	,*4		
005202	104400	HLT			
		ADCS:			
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	HLT			
		ROL3:			
005220	005253	INC	*(R3)		;(R0)=000001, CC=0001
005222	005654	SBC	*(R4)		;(R0)=000000, CC=0100
005224	103401	BCS	SBC5		
005226	001401	BEQ	,*4		
005230	104400	HLT			
005232	104000	SCOPE			
		SBC5:			
		ICHECK	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)		IBACK 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)		IBACK 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	010200	MOV	R2,R0		
		COMB	*(R2)		;(R0)=177400, CC=1001
005274	105152	BCC	COMB5		
005276	103001	BMI	,*4		
005300	100401	HLT			
005302	104400	HLT			
		COMB5:			
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	HLT			
		ASRB5:			
005326	105232	INCB	*(R2)+		;(R0)=177401, CC=0000
005330	103001	BCC	INCB3		
005332	100001	SPL	,*4		
005334	104400	HLT			
		INCB3:			
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	SPL	,*4		
005352	104400	HLT			
		RORB5:			
005354	106332	ASLB	*(R2)+		;(R0)=177000, CC=1001
005356	103002	BCC	ASLB3		
005360	102401	BVS	ASLB3		
005362	100401	BMI	,*4		
005364	104400	HLT			
		ASLB3:			
005366	105552	ADCB	*(R2)		;(R0)=177400, CC=1000
005370	103401	BCS	ADCB5		
005372	100401	BMI	,*4		
005374	104400	HLT			
		ADCB5:			
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	SPL	,*4		
005410	104400	HLT			
		ROLB3:			
005412	000352	SHAB	*(R2)		;(R0)=000777, CC=1000
005414	100401	BMI	,*4		
005416	104400	HLT			
		SHAB:			
005420	000261	SEC			
005422	105435	SBCB	*(R5)+		;(R0)=000377, CC=0100
005424	103401	BCS	SBCB3		
005426	001401	BEQ	,*4		
005430	104400	HLT			
		SBCB3:			
005432	105432	NECB	*(R2)+		;(R0)=000001
005434	105352	DECB	*(R2)		;(R0)=000000, CC=0101
005436	103001	BCC	DECB5		
005440	001401	BEQ	,*4		

005442 104400
005444 104000

DECB5: HLT
SCOPE

005446 005027
005450 000000
005452 010700
005454 024040
005456 000277
005460 006167 177764
005464 103403
005466 102402
005470 001401
005472 100001
005474 104400

JCHECK UNARY WORD OPS USING ADDRESS MODE 6 (PC)
CLR (PC)* ;PRESET DATA = 0
UNM6: ,WORD 0 ;RESERVED FOR DATA
MOV PC,R0 ;R0 POINTS TO DATA WORD
CMP =(R0),=(R0)
SCC
ROL UNM6 ;(R0)=000001,CC=0000
BCS ROL6
BVS ROL6
BEQ ROL6
BPL ,+4
ROL6: HLT

005476 005167 177746
005502 103002
005504 102401
005506 100401
005510 104400

COM UNM6 ;(R0)=177776, CC=1001
BCC COM6
BVS COM6
BMI ,+4
COM6: HLT

005512	006267	177732	ASR	UHM6	;(R0)=177777, CC=1010
005516	103402		BCS	ASR6	
005520	102001		BVC	ASR6	
005522	100401		BMI	,*3	
005524	104400		HLT		
			ASR6i		
005526	000277		SCC		
005530	005467	177714	NEG	UHM6	;(R0)=000001, CC=0001
005534	103003		BCC	NEG6	
005536	102402		BVS	NEG6	
005540	001401		BEQ	NEG6	
005542	100001		BPL	,*4	
005544	104400		HLT		
			NEG6i		
005546	000277		SCC		
005550	006067	177674	ROR	UHM6	;(R0)=100000, CC=1001
005554	103003		BCC	ROR6	
005556	102402		BVS	ROR6	
005560	001401		BEQ	ROR6	
005562	100401		BMI	,*4	
005564	104400		HLT		
			ROR6i		
005566	005667	177656	SBC	UHM6	;(R0)=077777, CC=0010
005572	103402		BCS	SBC6	
005574	102001		BVC	SBC6	
005576	100001		BPL	,*2	
005600	104400		HLT		
			SBC6i		
005602	000242		CLV		
005604	005267	177640	INC	UHM6	;(R0)=100000, CC=1011
005610	103403		BCC	INC6	
005612	102002		BVC	INC6	
005614	001401		BEQ	INC6	
005616	100401		BMI	,*2	
005620	104400		HLT		
			INC6i		
005622	006267	177622	ASR	UHM6	;(R0)=140000, CC=1010
005626	000261		SEC		
005630	006367	177614	ASL	UHM6	;(R0)=100000, CC=1001
005634	103002		BCC	ASL6	
005636	102401		BVS	ASL6	
005640	100401		BMI	,*4	
005642	104400		HLT		
			ASL6i		
005644	005367	177600	DEC	UHM6	;(R0)=077777, CC=0011
005650	103002		BCC	DEC6	
005652	102001		BVC	DEC6	
005654	100001		BPL	,*4	
005656	104400		HLT		
			DEC6i		
005660	005567	177564	ADC	UHM6	;(R0)=100000, CC=1010
005664	103402		BCS	ADC6	

005666	102001		BVC	ADC6	
005670	100401		BMI	,*4	
005672	104400		HLT		
			ADC6i		
005674	000242		CLV		
005676	000367	177546	SWAB	UHM6	
005702	100401		BMI	,*2	
005704	104400		HLT		
005706	022710	000200	CHP	#200,(R0)	
005712	001401		BEQ	,*7	
005714	104400		HLT		
005716	104000		SCOPE		
			ICHECK	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	,*2	
005754	104400		HLT		
			TSTB6i		
005756	000257		CCC		
005760	105767	000277	TSTB	UBM6+1	;(TEST ODD BYTE
005764	001401		BEQ	,*2	
005766	104400		HLT		
			SBCB6i		
005770	105667	000266	SBCB	UBM6	;(R0)=000000, CC=0100
005774	103402		BCC	SBCB6	
005776	102401		BVS	SBCB6	
006000	001401		BEQ	,*2	
006002	104400		HLT		
			SBCB6i		
006004	000261		1\$	SEC	
006006	105267	000250	INCB	UBM6	;(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		BCC	INCB6	
006024	102401		BVS	,*4	
006026	104400		HLT		
			INCB6i		
006030	106367	000226	ASLB	UBM6	;(R0)=077400, CC=0111
006034	103003		BCC	ASLB6	
006036	102002		BVC	ASLB6	
006040	001001		BNE	ASLB6	
006042	100001		BPL	,*2	
006044	104400		HLT		
			ASLB6i		
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		HLT		
		ADCB6i			
006064	000261		SEC		
006066	106067	000171	RORB	UBM6+1	;(R0)=140000, CC=1010
006072	103402		BVC	RORB6	
006074	102001		BVC	RORB6	
006076	100401		BMI	,+3	
006100	104400		HLT		
		RORB6i			
006102	105167	000154	COMB	UBM6	;(R0)=140377 CC=1001
006106	103002		BCC	COMB6	
006110	102401		BVS	COMB6	
006112	100401		BMI	,+4	
006114	104400		HLT		
		COMB6i			
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		HLT		
		NEGB6i			
006134	106167	000123	ROLB	UBM6+1	;(R0)=100777, CC=1010
006140	103402		BVS	ROLB6	
006142	102001		BVC	ROLB6	
006144	100401		BMI	,+3	
006146	104400		HLT		
		ROLB6i			
006150	106267	000106	ASRB	UBM6	;(R0)=100777, CC=1001
006154	103002		BCC	ASRB6	
006156	102401		BVS	ASRB6	
006160	100401		BMI	,+4	
006162	104400		HLT		
		ASRB6i			
006164	105267	000072	INCB	UBM6	;(R0)=100400, CC=0101
006170	103002		BCC	INCB6A	
006172	102401		BVS	INCB6A	
006174	100401		BEQ	,+4	
006176	104400		HLT		
		INCB6Ai			
006200	105367	000057	DECB	UBM6+1	;(R0)=100000, CC=1001
006204	103003		BCC	DECB6A	
006206	102402		BVS	DECB6A	
006210	100401		BEQ	DECB6A	
006212	100401		BMI	,+4	
006214	104400		HLT		
		DECB6Ai			
006216	000367	000040	SHAB	UBM6	;(R0)=000200, CC=1000
006222	103401		BVS	SHAB6	
006224	100401		BMI	,+4	
006226	104400		HLT		
		SHAB6i			

006230	106167	000026	ROLB	UBM6	;(R0)=000000, CC=0111
006234	103002		BCC	ROLB6A	
006236	102001		BVC	ROLB6A	
006240	100401		BEQ	,+3	
006242	104400		HLT		
		ROLB6Ai			
006244	005767	000012	TST	UBM6	;(R0)=000000, CC=0100
006250	103402		BVS	TST6	
006252	102401		BVS	TST6	
006254	100401		BEQ	,+4	
006256	104400		HLT		
		TST6i			
006260	000401		BR	,+4	!RESERVE A WORD
006262	000000		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

10000000000000 LAST ADDRESS OF CODE TO BE RELOCATED 0000000000

006626 103002
006630 102001
006632 001401
006634 104400
006636 104000

ROL7:
BCC ROL7
BVC ROL7
BEQ
HLT
SCOPE

006640 005720
006642 005210
006644 005740
006646 005010
006650 010701

ICHECK UNARY BYTE OPS USING ADDRESS MODE 7
TSY (R0)+
INC (R0) ;WORD FOLLOWING UWM7 CONTAINS ADDRESS
TSY -(R0) ;OF ODD BYTE, R0 POINTS TO DATA WORD
CLR (R0) ;PRESET DATA
MOV PC,R1 ;SET SCOPE PTR
INOTE: #2(2) REFERENCES THE ODD BYTE, AND #4(2) REFERENCES THE EVEN BYTE.

006652 000263
006654 105672 000002
006660 103003
006662 102402
006664 001401
006666 100401
006670 104400

+SECISEV ;SET C AND V
SBCB #2(2) ;(R0)=177400, CC=1001
BCC SBCB7
BVS SBCB7
BEQ SBCB7
BHI
SBCB7: HLT ,+4

006672 000277
006674 105572 177776
006700 103403
006702 102402
006704 001401
006706 100001
006710 104400

SCC ;SET CONDITION CODES
ADCB #2(2) ;(R0)=177401, CC=0000
BCC ADCB7
BVS ADCB7
BEQ ADCB7
BPL
ADCB7: HLT ,+4

006712 105172 177776
006716 103002
006720 102401
006722 100401
006724 104400

COMB #2(2) ;(R0)=177776, CC=1001
BCC COMB7
BVS COMB7
BHI
COMB7: HLT ,+4

006726 000241
006730 106072 000002
006734 103002
006736 102001
006740 100001
006742 104400

CLC ;CLEAR CARRY
RORB #2(2) ;(R0)=077776, CC=0011
BCC RORB7
BVC RORB7
BPL
RORB7: HLT ,+4

006744 105272 000002
006750 103002
006752 102001
006754 100401
006756 104400

INCB #2(2) ;(R0)=100376, CC=1011
BCC INCB7
BVC INCB7
BHI
INCB7: HLT ,+4

006760 105372 177776
006764 103002
006766 102401
006770 100401
006772 104400

DECB #2(2) ;(R0)=100375, CC=1001
BCC DECB7
BVS DECB7
BHI
DECB7: HLT ,+4

006774 106372 000002
007000 103002
007002 102001
007004 001401
007006 104400

ASLB #2(2) ;(R0)=000375, CC=0111
BCC ASLB7
BVC ASLB7
BEQ
ASLB7: HLT ,+4

007010 000241
007012 106272 177776
007016 103002
007020 102401
007022 100401
007024 104400

CLC ;CLEAR CARRY
ASRB #2(2) ;(R0)=000376, CC=1001
BCC ASRB7
BVS ASRB7
BHI
ASRB7: HLT ,+4

007026 105472 000002
007032 103402
007034 102401
007036 001401
007040 104400

NEGB #2(2) ;(R0)=000376, CC=0100
BCC NEGB7
BVS NEGB7
BEQ
NEGB7: HLT ,+4

007042 000262
007044 106172 177776
007050 103002
007052 102401
007054 100401
007056 104400

SEV ;(R0)=000374, CC=1001
ROLB #2(2)
BCC ROLB7
BVS ROLB7
BHI
ROLB7: HLT ,+4

007060 105272 177776
007064 105272 177776
007070 105572 177776
007074 105172 177776
007100 001401
007102 104400
007104 104000

INCB #2(2) ;(R0)=000375, CC=1001
INCB #2(2) ;(R0)=000376, CC=1001
ADCB #2(2) ;(R0)=000377, CC=1000
COMB #4(2) ;(R0)=000000, CC=0100
BEQ
HLT
SCOPE

007106 000277
007110 010700
007112 103002
007114 102401
007116 001001
007120 104400

ICHECK BINARY OPS USING ADDRESS MODE 8
SCC ;SET CONDITION CODES
MOV PC,R0 ;R0=PC, CC=X001
BCC MOV0
BVS MOV0
BNE
MOV0: HLT ,+4

007122 010002
007124 000262
007126 100002
007130 103402
007132 102401
007134 001401
007136 104400

MOV R0,R2 ;R2=R0
SEV ;SET V
SUB R0,R2 ;R2=000000, CC=0100
BCC SUB0
BVS SUB0
BEQ
SUB0: HLT ,+4

007140 000244
007142 010203
007144 103401

CLC ;R2=R3=000000, CC=0100
MOV R2,R3
BCC MOV0A

007146	001401		BEQ	,#4	
007150	104400	MOV#A:	HLT		
007152	000257		CCC		
007154	000272		+SEV:SEN		;SET V & N
007156	000203		CMF	R2;R3	;R2=R3=000000, CC=0100
007160	103403		BCS	CMF#	
007162	102402		BVS	CMF#	
007164	001001		BNE	CMF#	
007166	100001		BPL	,#3	
007170	104400	CMF#:	HLT		
007172	010002		MOV	R0;R2	;R0=R2
007174	010203		MOV	R2;R3	;R0=R2=R3
007176	000203		ADD	R2;R3	;R2=R0
007200	006302		ASL	R2	;R2=2*R0
007202	020203		CMF	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 R3 AND DOES A
BIT TEST (BIT) USING R2 AND R3,

007210	005002		CLR	R2	
007212	005202		INC	R2	
007214	000402		BR	25	
007216	006302	1S:	ASL	R2	
007220	100407		BMI	45	
007222	010205	2S:	MOV	R2;R5	
007224	000277		SCC		
007226	030205		BIT	R2;R5	;R2=R5
007230	103002		BCC	35	
007232	102401		BVS	35	
007234	001370		BNE	15	
007236	104400	3S:	HLT		
007240	010205	4S:	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	BIS#	
007262	102401		BVS	BIS#	
007264	001001		BNE	,#3	
007266	104400	BIS#:	HLT		
007270	010003		MOV	R0;R3	
007272	000277		SCC		
007274	000244		CLZ		
007276	040003		BIC	R0;R3	
007300	103003		BCC	BIS#	
007302	102402		BVS	BIC#	

007304	001001		BNE	BIC#	
007306	100001		BPL	,#4	
007310	104400	BIC#:	HLT		
007312	010004		MOV	R0;R4	
007314	005104		COM	R4	
007316	040004		BIC	R0;R4	
007320	005104		COM	R4	
007322	020004		CMF	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	103002		BCC	BIS#A	
007342	100401		BMI	,#3	
007344	104400	BIS#A:	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=17777
007360	000261		SEC		;SET C
007362	006004		ROR	R4	;R4=100000
007364	000304		ADD	R3;R4	;R3=17777;R4=07777, CC=0011
007366	103003		BCC	ADD#	
007370	102002		BVC	ADD#	
007372	001401		BEQ	ADD#	
007374	100001		BPL	,#4	
007376	104400	ADD#:	HLT		
007400	010700		MOV	PC;R0	
007402	022020		CMF	(R0);*(R0);*	
007404	020007		CMF	R0#PC	
007406	001401		BEQ	,#2	
007410	104400		HLT		
007412	010700		MOV	PC;R0	
007414	002700	000010	ADD	#10;R0	
007420	010002		MOV	R0;R2	
007422	020700		CMF	PC;R0	
007424	001002		BNE	CMF#A	
007426	020200		CMF	R2;R0	
007430	001401		BEQ	,#4	
007432	104400	CMF#A:	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	CMF	#125000;R4	
007452	001401		BEQ	,#1	

007454	104400		HLT		ERROR! B10B FAILED
007456	005004		CLR	R4	R3=125252, R4=0
007460	190304		B10B	R3,R4	R3=125252, R4=000252
007462	022704	000252	CHP	#252,R4	
007466	001401		BEQ	,+4	
007470	104400		HLT		ERROR! B10B FAILED
007472	110404		MOV B	R4,R4	R4=177652
007474	022704	177652	CHP	#177652,R4	MOV B EXTENDS THE SIGN
007500	001401		BEQ	,+4	
007502	104400		HLT		ERROR! MOV B FAILED
007504	132704	177525	B10B	#177525,R4	
007510	001401		BEQ	,+4	
007512	104400		HLT		ERROR! B10B FAILED
007514	105104		COM B	R4	R4=177525
007516	110404		MOV B	R4,R4	R4=000125
007520	022704	000125	CHP	#125,R4	
007524	001401		BEQ	,+4	
007526	104400		HLT		
007530	190304		B10B	R3,R4	R3=125252, R4=000377
007532	105204		INCB	R4	
007534	005704		TSY	R4	
007536	001401		BEQ	,+4	
007540	104400		HLT		
007542	104000		SCOPE		

007544	000402		ICHECK	BINARY OPS USING ADDRESS	MODE 1
007546	000000		BR	,*6	RESERVE TWO WORDS
007550	000000		WORD	0	RESERVED FOR SOURCE DATA
007552	010704		WORD	0	RESERVED FOR DESTINATION DATA
007554	003744		MOV	PC,R4	
007556	005044		TSY	-(R4)	R4 POINTS TO DESTINATION DATA
007560	010403		CLR	-(R4)	
007562	005043		MOV	R4,R3	R3 POINTS TO SOURCE DATA
			CLR	-(R3)	
007564	005113		COM	(R3)	R(R3)=177777
007566	005214		INC	(R3)	R(R4)=000001
007570	000262		SEV		SET V
007572	061314		ADD	(R3),(R4)	R(R3)=177777, (R4)=000000, CC=0101
007574	103002		BCC		
007576	102401		ADD1		
007600	001401		BEQ	,+4	
007602	104400		HLT		
		ADD1:			
007604	000277		SCC		
007606	000250		CLN		
007610	021314		CMP	(R3),(R4)	R(R3)=177777, (R4)=000000, CC=1000
007612	103403		BCS	CMP1	

007614	102402		BVS	CMP1	
007616	001401		BEQ	CMP1	
007620	100401		BH1	,+4	
007622	104400		HLT		
		CHP1:			
007624	000277		SCC		
007626	000244		CLZ		
007630	031314		BIT	(R3),(R4)	R(R3)=177777, (R4)=000000, CC=0101
007632	103002		BCC	BIT1	
007634	102401		BVS	BIT1	
007636	001401		BEQ	,+3	
007640	104400		HLT		
		BIT1:			
007642	000277		SCC		
007644	000245		*CLZ	CLZ	
007646	005114		COM	(R4)	R(R4)=177777
007650	161314		SUB	(R3),(R4)	R(R3)=177777, (R4)=000000, CC=0100
007652	103402		BCS	SUB1	
007654	102401		BVS	SUB1	
007656	001401		BEQ	,+3	
007660	104400		HLT		
		SUB1:			
007662	105013		CLRB	(R3)	R(R3)=177400
007664	000313		SHAB	(R3)	R(R3)=000377
007666	000270		SEN		
007670	011314		MOV	(R3),(R4)	R(R3)=(R4)=000377
007672	100001		BPL	,+4	
007674	104400		HLT		
007676	000314		SHAB	(R4)	R(R3)=000377, (R4)=177400
007700	000263		*SEV	SEV	SET C & V
007702	051314		BIS	(R3),(R4)	R(R3)=000377, (R4)=177777, CC=1001
007704	103002		BCC	BIT1	
007706	102401		BVS	BIT1	
007710	100401		BH1	,+3	
007712	104400		HLT		
		BIS1:			
007714	041314		BIT	(R3),(R4)	R(R3)=000377, (R4)=177400, CC=1001
007716	103002		BCC	BIT1	
007720	102401		BVS	BIT1	
007722	100401		BH1	,+4	
007724	104400		HLT		
		BIT1:			
007726	000262		SEV		SET V
007730	021314		CHP	(R3),(R4)	R(R3)=000377, (R4)=177400, CC=0001
007732	103003		BCC	CHP1A	
007734	100402		BVS	CHP1A	
007736	001401		BEQ	CHP1A	
007740	100001		BPL	,+4	
007742	104400		HLT		
		CHP1A:			
007744	005013		CLR	(R3)	R(R3)=000000
007746	000261		SEC		
007750	000013		ROR	(R3)	R(R3)=100000
007752	011314		MOV	(R2),(R4)	R(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	BM	,+4	
007766	104400	SUB1A	HLT	
007770	000277	SCC		
007772	161314	SUB	(R3),(R4)	(R3)=100000,(R4)=077777, CC=0000
007774	101402	BLDS	SUB1B	BRANCH IF C OR E 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	MOX1	
010010	100401	BM	,+2	
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	BVS	ADD1B	
010040	102001	BVC	ADD1B	
010042	100401	BM	,+3	
010044	104400	ADD1B	HLT	
010046	062714	ADD	#27(R4)	
010052	005714	TST	(R4)	;CHECK FINAL RESULT
010054	001401	BEQ	,+4	
010056	104400	HLT		
010060	104000	SCOPE		
010062	000402	ICHECK	BINARY BYTE OPS USING ADDRESS MODE I	
010064	000000	BR	,+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	MOV8	(R2),(R5)	(R2)=177400,(R5)=000377,CC=1001

010112	103005	BCC	MOV8	
010114	102404	BVS	MOV8	
010116	001403	BEQ	MOV8	
010120	100002	BPL	MOV8	
010122	102115	INCB	(R5)	;CHECK RESULT
010124	001401	BEQ	,+4	
010126	104400	MOV8	HLT	
010130	106312	ASLB	(R2)	SHIFT (R2) UNTIL
010132	102376	BVC	,=2	(R2)=000000
010134	106012	RORB	(R5)	(R2)=100000
010136	105315	DECB	(R5)	(R5)=00377
010140	106015	RORB	(R5)	(R5)=000177
010142	000277	CCC		
010144	121512	CHPB	(R5),(R2)	(R5)=000177,(R2)=100000, CC=1010
010146	102001	BVC	CHPB1	
010150	100401	BM	,+2	
010152	104400	CHPB1	HLT	
010154	005003	CLR	R3	
010156	000201	SEC		
010160	006003	ROR	R3	R3=100000
010162	000315	BIS	R3,(R5)	(R5)=100177
010164	000273	+SECTSEV	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	BITB	(R2),(R5)	(R2)=100000,(R5)=100377, CC=1001
010202	103001	BCC	BITB1	
010204	100401	BM	,+4	
010206	104400	BITB1	HLT	
010210	141215	BITB	(R2),(R5)	(R2)=100000,(R5)=100177, CC=0001
010212	103002	BCC	BICB1	
010214	001401	BEQ	BICB1	
010216	100001	BPL	,+2	
010220	104400	BITB1	HLT	
010222	105112	COMB	(R2)	(R2)=077400,(R5)=100177
010224	121215	CHPB	(R2),(R5)	
010226	001401	BEQ	,+4	
010230	104400	HLT		
010232	141512	BITB	(R5),(R2)	(R5)=100177,(R2)=000000, CC=0100
010234	021002	BNE	BITB1A	
010236	105712	TSTB	(R5)	
010240	001401	BEQ	,+4	
010242	104400	BITB1A	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	R2,R5	
010254	009745		YST	=(R5)	
010256	109045		CLRB	=(R5)	R5 POINTS TO DEST ODD BYTE
010260	010504		MOV	R5,R4	
010262	109044		CLRB	=(R4)	R4 POINTS TO DEST EVEN BYTE
010264	010403		MOV	R4,R3	
010266	109043		CLRB	=(R3)	R3 POINTS TO SOURCE ODD BYTE
010270	010302		MOV	R3,R2	
010272	109042		CLRB	=(R2)	R2 POINTS TO SOURCE EVEN BYTE

ICOMMENTS 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		ISET CARRY
010276	106112		ROLB	(R2)	I(R2),(R3),(R4),(R5)
010300	111214		MOVB	(R2),(R4)	I0001,0000,0000,0000
010302	106112		ROLB	(R2)	I0010,0000,0001,0000
010304	111213		MOVB	(R2),(R3)	I0010,0010,0001,0000
010306	106112		ROLB	(R2)	I0100,0010,0001,0000
010310	111315		MOVB	(R2),(R5)	I0100,0010,0001,0010
010312	106112		ROLB	(R2)	I1000,0010,0001,0010
010314	106113		ROLB	(R3)	I1000,0100,0001,0010
010316	151215		BISB	(R2),(R5)	I1000,0100,0001,1010
010320	131512		BITB	(R2),(R2)	I1000,0100,0001,1010
010322	001426		BEQ	BIN1	
010324	151314		BISB	(R3),(R4)	I1000,0100,0101,1010
010326	131413		BITB	(R2),(R3)	I1000,0100,0101,1010
010330	001423		BEQ	BIN1	
010332	109213		INCB	(R2)	I1000,0101,0101,1010
010334	121314		CMPB	(R2),(R4)	I1000,0101,0101,1010
010336	001020		BNE	BIN1	
010340	106113		ROLB	(R3)	I1000,1010,0101,1010
010342	121315		CMPB	(R2),(R5)	I1000,1010,0101,1010
010344	001015		BNE	BIN1	
010346	106212		ASRB	(R2)	I0100,1010,0101,1010
010350	131214		BITB	(R2),(R4)	I0100,1010,0101,1010
010352	001412		BEQ	BIN1	
010354	106015		RORB	(R5)	I0100,1010,0101,0101
010356	121415		CMPB	(R4),(R5)	I0100,1010,0101,0101
010360	001007		BNE	BIN1	
010362	109314		DECB	(R4)	I0100,1010,0100,0101
010364	141214		BICB	(R2),(R4)	I0100,1010,0000,0101
010366	001004		BNE	BIN1	
010370	111314		MOVB	(R2),(R4)	I0100,1010,1010,0101
010372	106213		ASRB	(R2)	I0100,0101,1010,0101
010374	141315		BICB	(R2),(R5)	I0100,0101,1010,0101
010376	001401		BEQ	,+4	
010400	104400		BIN1	HLT	
010402	104000		SCOPE		

ICHECK BINARY WORD OPS USING ADDRESS MODE 2 & 4
MOV R4,R5 ISET DESTINATION REGISTER
MOV #1,(R5)

010412	012712	177777	MOV	#=1,(R2)	
010416	000297		CCC		
010420	000262		SEV		
010422	062225		ADD	(R2)+,(R5)+	I(R2)=177777,(R5)=000000, CC=0101
010424	103002		BCC	ADD2	
010426	102401		BVS	ADD2	
010430	001401		BEQ	,+2	
010432	104400		ADD2:	HLT	
010434	000262		SEV		ISET V
010436	024527	000001	CHP	=(R5),#1	I(R5)=000000, CC=1001
010442	103002		BCC	CMP2	
010444	102401		BVS	CMP2	
010446	100401		BMI	,*	
010450	104400		CMP2:	HLT	
010452	054225		BIS	=(R2),(R5)+	I(R2)=177777,(R5)=177777, CC=1001
010454	103001		BCC	BIS2	
010456	100401		BMI	,+2	
010460	104400		BIS2:	HLT	
010462	000277		SCC		
010464	000244		CLZ		
010466	102245		SUB	(R2)+,(R5)	I(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)	I(R2)=000001
010502	005115		COM	(R2)	I(R5)=177777
010504	000277		SCC		
010506	000250		CLN		
010510	042225		BIC	(R2)+,(R5)+	I(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	(R2)+	I(R5)=052525
010534	000262		SEV		
010536	034245		BIT	=(R2),(R5)	I(R2)=125252,(R5)=052525, CC=0101
010540	103002		BCC	BIT2	
010542	102401		BVS	BIT2	
010544	001401		BEQ	,+2	
010546	104400		BIT2:	HLT	
010550	000262		SEV		
010552	052225		BIS	(R2)+,(R5)+	I(R2)=125252,(R5)=177777, CC=1001
010554	103002		BCC	BIS2A	
010556	102401		BVS	BIS2A	
010560	100401		BMI	,+4	

010562	104400		B102A1	HLT		
010564	042745	125252		BIG	#125252,=(R5)	(R5)=052525
010570	005125			COH	(R5)+	(R5)=125252
010572	024245			CKP	=(R2),=(R5)	
010574	001401			BEQ	,+4	
010576	104400			HLT		
010600	005012			CLR	(R3)	
010602	005122			COH	(R3)+	(R2)=177777
010604	102742	000001		SUB	#17,=(R2)	(R2)=177776, CC=1000
010610	103402			BCS	SUB2A	
010612	102401			BVS	SUB2A	
010614	100401			BMI	,+4	
010616	104400		SUB2A1	HLT		
010620	104000			SCOPE		
010622	010702			MOV	PC;R2	IGET CURRENT PC
010624	010205			MOV	R2;R5	IMOVE TO R5
010626	124245		1S1	CHPB	=(R2),=(R5)	ICOMPARE ALL PREVIOUS MEMORY ADDRESSES
010630	001401			BEQ	,+4	
010632	104400			HLT		IFERROR1
010634	020237	001010		CHP	R2;#FRSTAD	ICHECK FOR LOW LIMIT
010640	001372			BNE	13	
010642	104000			SCOPE		
010644	000402			ICHECK	BINARY BYTE OPS USING ADDRESS MODES 2 & 4,	
010646	000000			BR	,*6	RESERVE TWO WORDS
010650	000000			,WORD	0	SOURCE DATA
010652	010703			,WORD	0	DESTINATION DATA
010654	005743			MOV	PC;R5	
010656	127443	000200		TST	=(R3)	
010662	127443	000377		MOV	#200,=(R3)	(R3)=100377
010666	010304			MOV	#377,=(R3)	
010670	127444	000177		MOV	R3;R4	
010674	127444	000000		MOV	#177,=(R4)	(R4)=077400
010700	001401			MOV	#07,=(R4)	
010702	104400			BEQ	,+4	
010704	152324			HLT		
010706	100401			BIB	(R3)+,(R4)+	(R3)=100377,(R4)=077777
010710	104400			BMI	,+4	
010712	123324			HLT		
010714	103402			CHPB	(R3)+,(R4)+	
010716	102001			BCS	CHPB2	
010720	100001			BVC	CHPB2	
010722	104400		CHPB21	BPL	,+4	
010724	000261			HLT		
010726	134344			SEC		
010730	103002			BITB	=(R3),=(R4)	
010732	102401			BCC	BITB2	
				BVS	BITB2	

010734	001401			BEQ	,+4	
010736	104400		B102B1	HLT		
010740	000244			CLZ		
010742	144344			BICB	=(R3),=(R4)	(R3)=100377,(R4)=077400
010744	001401			BEQ	,+4	
010746	104400			HLT		
010750	104000			SCOPE		
010752	000404			ICHECK	BINARY WORD OPS USING ADDRESS MODES 3 & 5,	
010754	000000			BR	23	RESERVE SPACE FOR DATA AND ADDRESSES
010756	000000			,WORD	0	CONTAINS ADDRESS OF SOURCE DATA
010760	000000			,WORD	0	CONTAINS ADDRESS OF DEST DATA
010762	000000			,WORD	0	CONTAINS SOURCE DATA
010764	010701		2S1	,WORD	0	CONTAINS DEST DATA
010766	010100			MOV	PC;R1	
010770	024040			MOV	R1;R0	ISET SCOPE PTR
010772	010005			CHP	=(R0),=(R0)	ADJUST R0
010774	024545			MOV	R0;R5	R5 POINTS TO DEST DATA
010776	010015			CHP	=(R5),=(R5)	SUB 4 FROM R5
011000	010502			MOV	R0,(R5)	R5 POINTS TO ADDRESS OF DEST DATA
011002	010004			MOV	R5;R2	
011004	005740			MOV	R0;R4	R4 POINTS TO DEST DATA
011006	010003			TST	=(R0)	
011010	010042			MOV	R0;R3	R3 POINTS TO SOURCE DATA
011012	005013			MOV	R0;=(R2)	R2 POINTS TO ADDRESS OF SOURCE DATA
011014	005014			CLR	(R3)	PRESET SOURCE DATA
				CLR	(R4)	PRESET DEST DATA
011016	000277			SCC		
011020	000244			CLZ		
011022	103235			SUB	0(R2)+,0(R5)+	(R3)=000000,(R4)=000000, CC=0100
011024	103402			BCS	SUB3	
011026	102401			BVS	SUB3	
011030	001401			BEQ	,+4	
011032	104400		SUB31	HLT		
011034	005752	100000		BIS	#100000,0=(R2)	(R3)=100000
011040	002755	000001		ADD	#100,=(R5)	(R4)=000001
011044	103235			SUB	0(R2)+,0(R5)+	(R3)=100000,(R4)=100001, CC=1011
011046	103002			BCC	SUB3A	
011050	102001			BVC	SUB3A	
011052	100401			BMI	,+4	
011054	104400		SUB3A1	HLT		
011056	005414			NEG	(R4)	(R4)=077777
011060	005255			BIT	0=(R2),0=(R5)	(R3)=100000,(R4)=077777
011062	001401			BEQ	,+4	
011064	104400			HLT		
011066	023235			CHP	0(R2)+,0(R5)+	
011070	102401			BVS	,+4	
011072	104400			HLT		
011074	005132			COH	0=(R2)	
011076	000277			CCC		

011100	063255		ADD	0(R2)+,0=(R5)	
011102	102001		BVC	ADD3	
011104	100401		BMI	,+4	
011106	104400		HLT		
011110	000261		SEC		
011112	049235		BIC	0=(R2),0(R5)+	; (R3)=077777,(R4)=100000
011114	103001		BISB		
011116	100401		BMI	,+4	
011120	104400		HLT		
011122	005155		COM	0=(R5)	; (R4)=077777
011124	023235		CMF	0(22)+,0(R5)+	; (R3)=077777,(R4)=077777
011126	001401		BEQ	,+4	
011130	104400		HLT		
011132	104000		SCOPE		

ICHECK BINARY BYTE OPS USING ADDRESS MODES 3 & 5

011134	000400		BR	15	;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		151	MOV	PCTR0	
011154	024040		CMF	=(R0),=(R0)	;R0=ADDRESS OF DEST DATA	
011156	010003		MOV	R0,R3	;R3 " " "	
011160	010305		MOV	R3,R5	;R5 " " "	
011162	009743		TST	=(R3)	;SUB 2 FROM R3	
011164	010043		MOV	R04=(R3)	;R3 POINTS TO ADDRESS OF DEST DATA	
011166	009213		INC	(R3)	;ODD BYTE	
011170	010043		MOV	R0,=(R3)	;EVEN BYTE	
011172	010304		MOV	R3,R4		
011174	009740		TST	=(R0)	;R0=ADDRESS OF SOURCE DATA	
011176	010044		MOV	R07=(R4)	;R4 POINTS TO ADDRESS OF SOURCE DATA	
011200	009214		INC	(R4)	;ODD BYTE	
011202	010044		MOV	R07=(R4)	;EVEN BYTE	
011204	000261		SEC		;SET CARRY	
011206	012734	177001	MOV	#177001,0(R4)+		
011212	112734	000200	MOVB	#200,0(R4)+	;SOURCE DATA=100001	
011216	119433		MOVB	0=(R4),0(R3)+		
011220	119433		MOVB	0=(R4),0(R3)+	;DEST DATA=000600	
011222	103401		BCS	,+4		
011224	104400		HLT		;ERROR! MOV DOES AFFECT C BIT IN PSW	
011226	022715	000600	CMF	#600,(R5)	;CHECK DEST DATA	
011232	001401		BEQ	,+3		
011234	104400		HLT		;ERROR! INCORRECT RESULT	
011236	024343		CMF	=(R3),=(R3)	;POINT R4 BACK TO EVEN BYTE	
011240	153433		BISB	0(R4)+,0(R3)+		
011242	153433		BISB	0(24)+,0(R3)+	;DEST DATA=100601	
011244	022715	100601	CMF	#100601,(R5)	;CHECK RESULT	
011250	001401		BEQ	,+4		

011252	104400		HLT		;ERROR! INCORRECT DEST DATA AFTER BISB
011254	149453		BICB	0=(R4),0=(R3)	
011256	149453		BICB	0=(R4),0=(R3)	
011260	133433		BITB	0=(R4),0=(R3)+	
011262	001002		BNE	BITB3	
011264	139433		BITB	0=(R4),0(R3)+	
011266	001001		BNE	,+4	
011270	104400		HLT		
011272	123453		CMPB	0(R4)+,0=(R3)	
011274	001002		BNE	CMPB3	
011276	123453		CMPB	0(R4)+,0=(R3)	
011300	001401		BEQ	,+3	
011302	104400		HLT		
011304	104000		SCOPE		

ICHECK BINARY OPS USING ADDRESS MODE 6

011306	000402		BR	,+6	;RESERVE TWO LOCATIONS	
011310	000000		SDATA	,WORD	;RESERVED FOR SOURCE DATA	
011312	000000		DDATA	,WORD	;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	009065	011312	CLR	DDATA(5)	;PRESET DESTINATION DATA	
011326	012762	000001	011310	MOV	#1,SDATA(2)	;THIS ROUTINE PUT A 1 BIT INTO EVERY
011334	006265	011310	011312	BIS	SDATA(2),DDATA(5)	;OTHER BIT POSITION IN THE DEST-
011342	006362	011310	ASL	SDATA(2)	;INATION ADDRESS (92925)	
011346	006362	011310	ASL	SDATA(2)		
011352	103379		BCC	15		
011354	022765	052525	011312	CMF	#52525,DDATA(5)	;CHECK RESULT
011362	001401		BEQ	,+2		
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	030265	011310	011312	BIF	SDATA(2),DDATA(5)	
011410	001401		BEQ	,+3		
011412	104400		HLT		;ERROR! BIT INST FAILED	
011414	006362	011312	011312	ASL	DDATA(5)	;DDATA=125252
011420	020265	011310	011312	CMF	SDATA(2),DDATA(5)	
011426	001401		BEQ	,+4		

011430	104400		HLT		;ERROR! CMP INST FAILED	
011432	000297		CCC			
011434	066265	011310	011312	ADD	SDATA(2),DDATA(5)	
011442	103002		BCC	ADD6		
011444	100001		BVC	ADD6		
011446	100001		BPL	,+2		
011450	104400		HLT			
011452	006362	011310	011312	ASL	SDATA(2)	;SDATA=52524
011456	100265	011310	011312	SUB	SDATA(2),DDATA(5)	
011464	103401		BCS	SUB6		
011466	001401		BEQ	,+4		
011470	104400		HLT			

011472	112700	000377		MOVW	#377,R0	;R0=177777 (MOVW XR EXTENDS SIGN)
011476	010062	011310		MOV	R0,SDATA(2)	
011502	012765	177777	011312	MOV	#0,DDATA(5)	
011510	166500	011312		SUB	DDATA(5),R0	
011514	001401			BEQ	,+3	
011516	104400			HLT		
011520	066265	011310	011312	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	,+3	
011564	104400			HLT		
011566	026200	011310		CMP	SDATA(2),R0	
011572	001352			BNE	1\$	
011574	104000			SCOPE		

ICHECK BINARY 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 DATA

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	MOVW	#177125,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),SDATAB(3)	
011652	001401			BEQ	,+4	
011654	104400			BITB6:	HLT	
011656	146264	011740	011742	BITCB	SDATAB(2),DDATAB(4)	
011664	103401			BCC	,+3	
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	BITCB	SDATAB(3),DDATAB(5)	
011710	126265	011740	011742	CMPB	SDATAB(2),DDATAB(5)	
011716	001401			BEQ	,+3	
011720	104400			HLT		
011722	136564	011742	011742	BITB	DDATAB(5),DDATAB(4)	
011730	001401			BEQ	,+3	
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
ICHECK BINARY WORD OPS USING ADDRESS MODE 7						
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	,+3	
012026	001401			BEQ	,+3	
012030	104400			HLT		
012032	027777	177706	177706	CMP	SBIN7,DBIN7	;{(R2)=100000, (R3)=000000
012040	103402			BCC	CMP7	
012042	102401			BVS	CMP7	
012044	100401			BMI	,+3	
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			BMI	,+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	,+3	
012112	104400			ADD7:	HLT	

012114	047777	177624	177624	BIC	#SBIN7,#DBIN7	;(R2)=140000,(R3)=000000
012122	001401			BEQ	,+2	
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	CMF	#SBIN7,#DBIN7	
012146	001401			BEQ	,+4	
012150	104400			HLT		
012152	104000			SCOPE		

ISOME MISCELLANEOUS OPERATION INVOLVING THE PC
NOTE: NONE OF THESE OPERATIONS SHOULD AFFECT THE PC

012154	005000			CLR	R0	
012156	005067	000072		CLR	1S	
012162	010707			MOV	PC,PC	
012164	120707			CMFB	PC,PC	
012166	030707			BIT	PC,PC	
012170	060007			ADD	R0,PC	
012172	105707			TSTB	PC	
012174	005507			ADC	PC	
012176	021007			CMF	(R0),PC	
012200	131007			BITB	(R4),PC	
012202	062707	000000		ADD	#0,PC	
012206	023707	001004		CMF	#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			CMF	=(SP),PC	
012234	132607			BITB	(SP)+,PC	
012236	026707	000012		CMF	1S,PC	
012242	166707	000006		SUB	1S,PC	
012246	046707	000002		BIC	1S,PC	
012252	000401			BR	,+4	;BRANCH OVER 1S
012254	000000			0		
012256	104000			SCOPE		
012260	010702			MOV	PC,R2	
012262	062702	000012		ADD	#12,R2	
012266	012707	001132		MOV	#RELOC,PC	;GO RELOCATE PROGRAM CODE
012272	000240			NOP		;PROGRAM RETURNS HERE+2

1111111111111111 LAST ADDRESS OF CODE TO BE RELOCATED 111111111111

1222222222222222	FIRST ADDRESS TO BE RELOCATED	2222222222
REL2:	MOV	PC,R0 ;GET PC
	TST	=(R0) ;R0 CONTAINS THE ADDRESS OF REL2
	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
012320	000406			BC	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			CMF	=(R0),=(R0)	;R0 = ADDRESS OF DEST DATA
012342	010060	177772		MOV	R0=#6(R0)	;LOAD ADDRESS 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


```

013000 012706 000500 2S1 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 002700 000012 ADD #10,R0 ;SET ADDRESS FOR JMP INST
013030 000277 SCC ;SET CC'S
013032 000110 JMP (R0)
013034 000402 BR ,+2
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 ,+2
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 042713 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 ,=1
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 15
013134 005102 COM R2 ;COMPLEMENT INDICATOR
013136 000403 BR 25
013140 022424 15i CMP (R4)+,(R4)* ;R4=JMP ADDRESS
013142 005724 TST (R4)+ ;USE R4 AS ADDRESS
013144 000144 JMP =(R4) ;CHECK INDICATOR
013146 005202 25i INC R2
013150 001003 BNE JMP4
013152 022020 CMP (R0)+,(R0)* ;CHECK AUTO=DEC R4
013154 020004 CMP R0,R4
    
```

```

013156 001401 BEQ ,+4
013160 104400 JMP4 HLT

013162 010703 MOV PC,R3
013164 000401 BR ,=1 ;RESERVE WORD FOR JMP ADDRESS
013166 000000 ,WORD 0 ;CONTAINS JUMP ADDRESS
013170 005723 TST (R3)+
013172 010313 MOV R3,(R3)
013174 042723 000016 ADD #10,(R3)+ ;LOAD CHECK REGISTER
013200 010300 MOV R3,R0
013202 000402 BR 35
013204 005102 25i COM R2
013206 000401 BR 45
013210 000193 35i JMP @={R3} ;JUMP TO 25 VIA 15 ABOVE
013212 005202 45i 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 25
013230 005102 15i COM R2 ;COMPLEMENT INDICATOR
013232 000402 BR 35
013234 000167 177770 25i JMP 15
013240 005202 35i INC R2
013242 001401 BEQ ,+4
013244 104400 JMP6 HLT

013246 012767 013264 000020 MOV #15,75 ;SET UP JMP ADDRESS
013254 063767 001004 000012 ADD @#FACTOR,75 ;ADD RELOCATION FACTOR
013262 000402 BR 25 ;GO TO JMP 075 INST
013264 005102 15i COM R2 ;COMPLEMENT INDICATOR
013266 000403 BR 35 ;GO TO CHECK ROUTINE
013270 000177 000000 25i JMP 075 ;JMP TO 15 ABOVE VIA 75
013274 000000 ,WORD 0 ;CONTAINS JMP ADDRESS
013276 005202 35i INC R2 ;CHECK INDICATOR
013300 001401 BEQ ,+4
013302 104400 JMP7 HLT
013304 104000 SCOPE

;CHECK JSR INSTRUCTIONS
013306 013705 001004 JSR R5 ;SET RELOCATION FACTOR
013312 012702 013344 MOV @#FACTOR,R5 ;FORM DEST ADDR
013316 040502 ADD #35,R2 ;ADD RELOCATION FACTOR
013320 000277 SCC ;PRESET CC'S
013322 000242 CLV
013324 004512 JSR R5,(R2) ;GO TO 35 VIA R2
013326 005702 15i TST R2 ;CHECK INDICATOR
013330 001017 JSR1 ;R2 SHOULD BE 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 RTS R5 ;RETURN FROM SUBROUTINE
    
```

013344	103011		3S1	BCC	JSR1		ICHECK THAT JSR DID NOT
013346	102410			BVS	JSR1		
013350	001007			BNE	JSR1		IAFFECT CC'S
013352	100006			BPL	JSR1		
013354	005002			CLR	R2		ICLEAR INDICATOR
013356	012704	013326		MOV	#15,R4		IGET UNRELOCATED RETURN ADDRESS
013362	001604			ADD	(SP),R4		IADD RELOCATION FACTOR (OLD R5)
013364	020405			CHP	R4,R5		ICHECK THAT OLD R5 WAS PLACED ON THE
013366	001765			BEQ	25		ISTACK, & THAT NEW R5 CONTAINS RETURN PC
013370	104400			HLT			ERROR! ABOVE
013372	013704	001004		MOV	##FACTOR,R4		IGET RELOCATION FACTOR
013376	005000			CLR	R0		ISET INDICATOR
013400	012705	013420		MOV	#15,R5		
013404	000405			ADD	R4,R5		ISET UP JSR DEFERRED ADDR
013406	010502			MOV	R5,R2		
013410	012715	013436		MOV	#55,(R5)		
013414	000415			ADD	R4,(R5)		(R5)=DEST ADDR
013416	000401			BR	25		RESERVE WORD FOR ADDRESS
013420	000000		1S1	,WORD	0		CONTAINS DEST ADDR FOR JSR
013422	004435		2S1	JSR	R4,(R5)+		JSR TO 55 VIA 15 ABOVE
013424	005200		3S1	INC	R0		ICHECK INDICATOR
013426	001013			BNE	JSR3		
013430	000413			BR	JSR3A		
013432	005100		4S1	COM	R0		ICOMPLEMENT INDICATOR
013434	000204			RTS	4		IRETURN FROM SUBROUTINE
013436	012703	013424	5S1	MOV	#35,R3		IGET UNRELOCATED RETURN ADDRESS
013442	001603			ADD	(SP),R3		IADD RELOCATION FACTOR (OLD R4)
013444	020403			CHP	R4,R3		
013446	001003			BNE	JSR3		
013450	005722			TST	(R6)+		
013452	020205			CHP	R2,R5		ICHECK AUTO-INC R5
013454	001766			BEQ	45		IGO TO RTS
013456	104400			HLT			ERROR ABOVE
013460	013704	001004		MOV	##FACTOR,R4		
013464	010405			MOV	R4,R5		
013466	010703			MOV	PC,R3		
013470	000401			BR	25		
013472	000405		1S1	BR	45		
013474	022323		2S1	CHP	(R3)+(R3)+		
013476	000277			SCC			
013500	004443			JSR	R4,(R3)		IGO TO 25
013502	104400		3S1	HLT			
013504	000414			BR	JSR4A		
013506	103012		4S1	BCC	JSR4		
013510	102011			BVC	JSR4		
013512	001010			BNE	JS64		
013514	100007			BPL	JSR4		
013516	012702	013502		MOV	#37,R2		IGET UNRELOCATED RETURN ADDRESS
013522	001602			ADD	(SP),R2		IADD RELOCATION FACTOR (OLD R4)
013524	020204			CHP	R2,R4		ICHECK THAT CALCULATED RETURN
013526	001002			BNE	JSR4		IPC = NEW R4
013530	005724			TST	(R7)+		

013532	000204			RTS	R4		
013534	104400			HLT			
013536	000401			BR	25		
013540	000405			BR	35		
013542	010700			MOV	PC,R0		
013544	004767	177770		JSR	PC,15		
013550	100407			BMI	JSR6A		
013552	104400			HLT			
013554	022020		3S1	CHP	(R0)+(R0)+		
013556	020016			CHP	R0,(SP)		ICHECK THAT RETURN ADDRESS IS ON THE
013560	001401			BEQ	,*4		ISTACK
013562	104400			HLT			
013564	000270			SEN			ISET N
013566	000207			RTS	PC		
013570	104000			SCOPE			
013572	012737	013624	000020	ICHECK	DOT TRAP		
013600	003737	001004	000020	MOV	(AND R0LB/ASLB)		
013606	000261			ADD	#10T1,##IOTVEC		
013610	013737	177776	000022	SEC	##FACTOR,##IOTVEC		IADD RELOCATION FACTOR
013616	005000			MOV	##PSW,##IOTVEC+2		ISET CARRY
013620	000004			CLR	R0		IRETAIN CURRENT PSW ON TRAP
013622	000403			ROT			IPRESET R0
013624	106100		10T11	BR	10T1A		
013626	102376			ROLB	R0		IRotate R0
013630	000002			BVC	,#2		UNTIL V SETS (R0=200)
013632	106300		10T1A1	RTI	R0		ISHIFT SHOULD SET CARRY
013634	103004			BCC	10T1B		
013636	102003			BVC	10T1B		
013640	001002			BNE	10T1B		
013642	005700			TST	R0		IR0 SHOULD =0
013644	001401			BEQ	,*4		
013646	104400		10T1B1	HLT			ERROR! ROL/ASL FAILED TO SET CC'S PROPERLY
013650	012737	000022	000020	MOV	##IOTVEC+2,##IOTVEC		IRESTORE DOT TRAP
013656	005037	000022		CLR	##IOTVEC+2		IVECTOR
013662	104000			SCOPE			
013664	013746	000030		ICHECK	EMT TRAP SEQUENCE		
013670	012737	013724	000030	MOV	##EMTVEC,##(SP)		ISAVE SCOPE PTR
013676	003737	001004	000030	MOV	##EMT1,##EMTVEC		ISET EMT TRAP VECTOR
013704	000262			ADD	##FACTOR,##EMTVEC		IADD RELOCATION FACTOR
013706	013737	177776	000032	SEV			ISET V
013714	000265			MOV	##PSW,##EMTVEC+2		IRETAIN CURRENT PSW ON TRAP
013716	104000			+SEI=ISEC			
013720	001433			EMT			ITRIP TO EMT1
013722	104400			BEQ	EMT1C		IGO TO EMT1C
013724	102027		EMT11	HLT			ERROR! INCORRECT CC'S WERE SET ON RETURN
013726	105100			BVC	EMT1B		IVV SHOULD'VE SET ON EMT TRAP
013730	105500			COMB	R0		IR0=000377,CC'S=1001
013732	106000			ADCB	R0		IR0=000000,CC'S=0101
				RORB	R0		IR0=000200,CC'S=1010

013734	102023			BVC	EMT10	
013736	100022			BPL	EMT10	
013740	000257			CCC		
013742	105400			NEGB	R0	IR0=000200,CC'S=1010
013744	102017			BVC	EMT10	
013746	100016			BPL	EMT10	
013750	000242			CLV		ICLEAR /V/
013752	000261			SEC		IAND SET /0/
013754	105300			DECB	R0	IR0=000177,CC'S=0011
013756	102012			BVC	EMT10	
013760	100411			BMI	EMT10	
013762	000242			CLV		ICLEAR /V/
013764	105200			INCB	R0	IR0=000200,CC'S=1011
013766	103006			BCC	EMT10	
013770	102005			BVC	EMT10	
013772	100004			BPL	EMT10	
013774	000242			CLV		ICLEAR /V/
013776	106200			ASRB	R0	ISHIFT R0 UNTIL /V/ CLEARS
014000	102776			BVS	,+2	
014002	000401			BR	,+4	
014004	104400			EMT10:	HLT	IERROR:
014006	000002			RTI		IEXIT WITH R0=000377
014010	105500			EMT10:	ADCB	R0
014012	103003			BCC	EMT10	IR0=000000
014014	001002			BNE	EMT10	
014016	005700			TST	R0	
014020	001401			BEQ	,+4	
014022	104400			EMT10:	HLT	
014024	012637	000030		MOV	(SP)+,EMTVEC	IRESTORE SCOPE PTR
014030	005037	000032		CLR	EMTVEC+2	
014034	104000			SCOPE		

ICHECK TRAP INSTRUCTION TRAP SEQUENCE

014036	013737	000034	000020	HLT=IOT	IREDEFINE HLT	
014044	012737	014112	000034	MOV	EMTVEC,EMTVEC	ISET IOT (HLT) TRAP VECTOR
014052	003737	001004	000034	MOV	#TRAP1,EMTVEC	ISET TRAP VECTOR
014060	000270			ADD	EMTVEC,EMTVEC	IADD RELOCATION FACTOR
014062	013737	177776	000036	SEN		ISET N
014070	000261			MOV	EMTVEC,EMTVEC+2	IRETAIN CURRENT PSW ON TRAP
014072	110700			SEC		ISET CARRY
014074	000264			MOV	PCTR0	
014076	104400			SEZ		ISET Z BIT
014100	103401			TRAP		ITRAP TO TRAP1
014102	000004			BCS	,+4	
014104	001401			HLT		
014106	000004			BEQ	,+4	
014110	000412			HLT		
014112	100401			BR	TRAP1C	
014114	000004			EMT10:	HLT	IN BIT GOT SET ON TRAP
014116	002700	000004		ADD	#4,R0	
014122	120016			CMQB	R0,(SP)	ICHECK LOW BYTE OF RETURN PC ON
014124	001401			BEQ	,+4	ISTACK
014126	000004			HLT		

014130	124646			CMQB	-(SP),-(SP)	
014132	032626			BIT	(SC)+,(SP)*	
014134	000002			RTI		IRETURN TO INST FOLLOWING TRAP (IS)
014136	013737	000020	000034	TRAP1C:	MOV	EMTVEC,EMTVEC
014144	012737	000200	000036	MOV	EMTVEC+2,EMTVEC+2	IRESTORE TRAP (HLT) TRAP VECTOR
014152	012737	000022	000020	MOV	EMTVEC+2,EMTVEC	
014160	005037	000022		CLR	EMTVEC+2	
014164	104000			SCOPE		
014166	104400			HLT=TRAP		IRESTORE HLT TO A TRAP INST
014166	010702			MOV	PCTR2	
014170	002702	000012		ADD	#12,R2	
014174	012707	001132		MOV	#RRELOC,PC	IGO RELOCATE PROGRAM CODE
014200	000240			NOP		IPROGRAM RETURNS HERE+2
014202	010701			MOV	PCTR1	ISET SCOPE PTR
014204	000000					LAST ADDRESS OF CODE TO BE RELOCATED 2222222222
014204	005767	164570				
014210	001036					ITHE BELOW ROUTINE ASCERTAINS WHICH CP & CP OPTIONS THE PROGRAM IS RUN-
014212	012737	000002	000006	CPCHK:	TST	ICNT
014220	012700	000003		BNE	REJ3	ICHECK IF PASS 0
014224	000261			MOV	EMTVEC,EMTVEC+2	IDO NOT EXECUTE ROUTINE IF NOT PASS 0
014226	005737	177772		SEC	#3,R0	ISET UP ERROR TRAP TO RETURN
014232	005600			TST	EMTVEC	IR0=3 IF 11/45
014234	000261			SBC	R0	IR0=2 IF 11/40
014236	105737	177777		SEC		
014242	005600			TSTB	EMTVEC+1	IR0=1 IF 11/20
014244	005037	177700		SBC	R0	IR0=0 IF 11/05
014250	006300			CLR	EMTVEC+1	IR0=0 IF 11/05
014252	010027			ASL	R0	ISHIFT INDICATOR
014254	000000			MOV	R0,(PC)+	ISET CP INDICATOR
014256	005037	000006		WORD	0	ICONTAINS OPTION & CP INDICATORS
014262	005037	000012		JEVEN BYTE	0	IF EVEN BYTE
014264	005037	000012		JEVEN BYTE	0	IF EVEN BYTE
014266	120727	177762	000004	CMQB	OPI,CP,#4	IBRANCH IF 11/05 OR 11/20
014274	002404			BLT	RELS	
014276	004767	164730		JSR	PC,PRINT	IPRINT MESSAGE BEGINING AT FOLLOWING ADRS
014302	016625			ILLTEST		
014304	000000			HALT		
014306	010700			REL3:	MOV	PCTR0
014310	005740			TST	-(R0)	IR0 CONTAINS THE ADDRESS OF REL3
014312	010037	001010		MOV	EMTVEC,EMTVEC	ISAVE
014316	010700			MOV	PCTR0	IGET CURRENT PC
014320	102700	014320		SUB	EMTVEC,EMTVEC	ISUBTRACT RELOCATION FACTOR

```

014324 010037 001004      MOV      R0,#FACTOR      ;SAVE RELOCATION FACTOR
014330 010701              MOV      PC,R1           ;SET NEW SCOPE PTR

014332 013767 177776 000306  ;CHECK STACK OVERFLOW
014340 005037 177776      MOV      #0,PSW,75     ;SAVE STATUS IN 75 BELOW
                                CLR      #0,PSW         ;SET KERNEL MODE
014344 010746              MOV      PC,=(SP)      ;PUSH CURRENT PC ONTO STACK
014346 062716 000136      ADD      #25,=(SP)     ;FORM ADDRESS OF 25 BELOW
014352 011637 000004      MOV      (SP),#ERRVEC  ;SET ERROR VECTOR
014356 012737 000340 000006  ;340,#ERRVEC+2        ;SET PRIORITY LEVEL 7 ON TRAP
014364 062716 000074      ADD      #415=25,(SP)  ;FORM ADDRESS OF 415 BELOW
014370 012637 000020      MOV      (SP),#IOTVEC  ;SET IOT TRAP VECTOR TO 415
014374 012746 000340      MOV      #370,=(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       ;LOAD 376 INTO ADDRESS 376
014420 010313              MOV      R3,(R3)
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      123          ;BRANCH IF 11/40 OR 11/45
014442 001404              BEQ      113          ;BRANCH IF 11/20
014444 012767 000014 000144  ;MOV     #14,515       ;CHANGE CHECK WORD IN 515 IF 11/05
014452 000407              BR      103
014454 012767 000034 000134  ;MOV     #34,515       ;CHANGE CHECK WORD IN 515 IF 11/20
014462 000493              BR      105
014464 012656              MOV      (SP),#=(SP)  ;BECAUSE OF ADDRESS MODE 5
014466 054676 000000      BIS      =(SP),#(SP)  ;BECAUSE OF ADDRESS MODE 7
014472 005066 000004      CLR      4(SP)        ;BECAUSE DEST ADDRESS IS > 376
014476 0057636 000000     BIS      @ (SP),#(SP)+ ;BECAUSE OF ADDRESS MODE 3
014502 000406              BR      35            ;BRANCH OVER NON KERNEL MODE TESTS

;ERROR SERVICE ROUTINE
014504 012600              MOV      (SP),#R0     ;SAVE PC OF INSTRUCTION THAT TRAPPED
014506 012692              MOV      (SP),#R2     ;SAVE PSW
014510 012706 000500      MOV      #STKPTR,SP  ;SET STACK PTR
014514 104400              HLT                    ;ERROR! AN INSTRUCTION THAT WAS NOT
                                ;SUSPESED TO TRAP TRAPPED
                                ;R0 CONTAINS PC, R2 CONTAINS PSW

014516 000450              BR      65            ;EXIT TEST

;THE BELOW INSTRUCTIONS WILL CAUSE A STACK OVERFLOW
;STACK PTR IS AT 376
014520 062737 000066 000004  ;ADD     #43=25,#ERRVEC ;SET ERROR VECTOR TO 43
014526 010396              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            ;SETS BIT 2 IN R0
014546 060746              ADD      PC,=(SP)     ;SETS BIT 2 IN R0
014550 006302              ASL      R2            ;SETS BIT 3 IN R0
014552 000004              IOT
014554 006302              ASL      R2            ;SETS BIT 4 IN R0
014556 004767 000014      JSR      PC,40$       ;NOTE! 11/05 WITHOUT ECO # KD11A=00005
014562 006302              ASL      R2            ;DOES NOT SET BIT 4,
                                ;SETS BIT 5 IN R0

014564 006666 177776      BIS      SP,=2(SP)    ;SETS BIT 5 IN R0
014570 000407              BR      55

;PROGRAM WILL TRAP HERE ON OVERFLOW TRAP
014572 050200              45$     BIS      R2,R0   ;SET APPROPRIATE BIT IN R0
014574 000002              RTI                    ;RETURN FROM TRAP

014576 000207              40$     RTS      PC

014600 012737 000022 000020  ;MOV     #IOTVEC+2,#IOTVEC
014606 000002      RTI

;CHECK THAT ABOVE INSTRUCTIONS 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
014616 000000      51$     ;CONTAINS CHECK WORD
014620 001407              BEQ      65           ;R0= 77 IF 40 OR 45,14 IF 05,34 IF 20
014622 105737 014254      TSYB    #OPT,CP      ;CHECK IF 11/05
014626 001003              BNE     52$          ;BRANCH IF NOT AN 11/05
014630 022700 000034      CMP      #34,R0      ;USE ECO KD11A=00005 CHECK WORD
014634 001401              BEQ     63$
014636 104400              HLT

;EXIT ROUTINE
014640 012706 000600      6$     MOV      #KPTR,SP  ;SET KERNEL STACK PTR
014644 012746              MOV      (PC),#(SP)  ;PUSH OLD PSW ONTO STACK
014646 000000      7$     ;CONTAINS SAVED PSW
014650 010746              ;WORD 0
014652 062716 000006      MOV      PC,=(SP)   ;PUSH CURRENT PC ONTO STACK
014656 000002              ADD      #6,(SP)     ;ADD OFFSET
014660 012706 000500      RTI                    ;SET STACK PTR
014664 012737 000006 000004  ;MOV     #STKPTR,SP
014672 104000              MOV     #ERRVEC+2,#ERRVEC

;CHECK THAT ALL RESERVED INSTRUCTIONS TRAP (TO LOCATION 10)
014674 012737 000002 001114  ;RESTRP; MOV #2,#SCOPED ;INIT TO TWO ITERATIONS
014702 010701              MOV      PC,R2       ;SET SCOPE POINTER
014704 012702 015024      MOV      #55,R2     ;GET ADDRESS OF RESERVED INSTRUCTION TABLE
014710 063702 001004      ADD      #55,R2
014714 122737 000004 014254  ;CMPB    #55,PC       ;ADJUST TABLE ADDRESS IF 11/20, 11/05
014722 003402              BLE     113          ;55=11/45, 11/40 TABLE, 65=11/05
014724 062702 000036      ADD      #63=55,R2   ;11/20 TABLE
014730 132737 000040 014295  ;BITB   #40,#OPT,CP+1 ;CHECK IF 11/45 FLOATING POINT IS AVAILA
014736 001402              BEQ     *0          ;BRANCH IF NOT AVAILABLE
    
```

```

014740 005067 000110 CLR 50$ ;SET TABLE TERMINATOR AT GROUP 7
014744 012737 015002 000010 MOV #4$,#RESVEC ;SET RESERVED INSTRUCTION TRAP
014752 063737 001004 000010 ADD #FACTOR,#RESVEC
014760 012203 1S) MOV (R2),R3 ;GET FIRST RESERVED INSTRUCTION
014762 001494 BEQ 7$ ;TERMINATES THE TABLE
014764 012204 MOV (R2),R4 ;GET LAST RESERVED INSTRUCTION IN GROUP
014766 010317 2S) MOV R3,(PC) ;EXECUTE RESERVED INSTRUCTION
014770 000000 3S) ;CONTAINS RESERVED INSTRUCTION
014772 104400 HLT ;ERROR! INSTRUCTION IN R3
014774 104400 HLT ;(2S) ABOVE FAILED TO CAUSE A
014776 104400 HLT ;RESERVED INSTRUCTION TRAP
015000 000405 BR 41$
015002 012716 015014 4S) MOV #41$, (SP) ;ADJUST RETURN PC
015006 063716 001004 ADD #FACTOR, (SP) ;TO RETURN TO 41$
015012 000002 RTI ;RETURN TO 41$
015014 020304 41S) CMP R3,R4 ;HAS GROUP OF RESERVED INSTRUCTIONS
015016 001760 BEQ 1$ ;BEEN EXECUTED
015020 005203 INC R3 ;INCREMENT THIS RESERVED INSTRUCTION
015022 000761 BR 2$ ;TO NEXT ONE AND EXECUTE

;TABLE OF 11/40, 11/45 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)
015024 000007 5S) 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 50S) 170000 ;GROUP 7 FLOATING POINT
015056 177777 ; INSTRUCTIONS
015060 000000 0 ;TERMINATES THE TABLE

;TABLE OF 11/05, 11/20 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)
015062 000006 6S) 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 ;TERMINATES THE TABLE
015114 012737 000012 000010 7S) MOV #RESVEC+2,#RESVEC ;RESTORE RESERVED TRAP TO HALT AT 12
015122 104000 SCOPE

```

ICHECK THAT ALL BITS IN THE PROCESSOR STATUS WORD (PSW) CAN BE SET AND

```

015124 013767 177776 000152 ;Cleared, PSMCHKI 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 #67,(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 ,+2 ;
015166 104400 HLT ;ERROR! PSW FAILED TO CLEAR
015170 113700 014254 MOV#B #OPT,CP,R0 ;GET CP TYPE
015174 010000 016564 MOV PSMBIT(0),R0 ;GET BIT MASK FOR TEST R0=THOSE BITS IN
;THE PSW WHICH CAN BE SET/CLEARED,
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 10S) MOV #1,R2 ;R2 = TEST BIT
015216 030200 1S) 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 '1' BIT
015232 001403 BEQ 20$ 20$
015234 012737 000002 000016 MOV #RTI,#TBITVEC+2 ;SET RTI INTO RETURN
015242 005014 CLR (R2) ;CLEAR PSW
015244 050214 BIS R2,(R4) ;SET R2 INTO PSW
015246 011403 MOV (R2),R3 ;GET BIT
015250 020203 CMP R2,R3 ;CHECK THAT BIT WAS SET IN PSW
015252 001401 BEQ ,+2 ;
015254 104400 HLT ;ERROR! BIT IN R2 FAILED TO SET IN PSW
015256 000244 CLE ;CLEAR Z BIT
015260 040214 BIC (R4),R3 ;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 000302 2S) ASL R2 ;SHIFT TEST BIT
015272 103391 BCC 1$ ;BRANCH IF ALL BITS NOT TESTED
015274 005014 CLR (R4) ;CLEAR STATUS
015276 012637 000016 MOV (SP)+,#TBITVEC+2 ;RESTORE T BIT RETURN
015302 012746 MOV (PC)+,=(SP) ;PUSH ORIGINAL STATUS ON STACK
015304 000000 ;CONTAINS ORIGINAL PSW
015306 010746 MOV PC,(SP) ;SET RETURN PC
015310 062716 000006 ADD #61,(SP)
015314 000002 RTI ;RETURN
015316 104000 4S) SCOPE

015320 013704 177776 MOV #PSW,R4 ;SAVE PSW IN R4
015324 010446 MOV R4,(SP) ;PUSH R4 ONTO STACK
015326 112716 000300 MOV#B #300,(SP) ;SET PRIORITY LEVEL 6 AND
015332 010746 MOV PC,(SP) ;CLEAR '1' BIT AND EXECUTE
015334 062716 000006 ADD #61,(SP) ;INSTRUCTION FOLLOWING RTI

```



```

015340 000002 RTI
ICHECK THAT ALL BITS IN THE CURRENT STACK PTR CAN BE SET/CLEARED
CHKSP: MOV SP,R3 ;SAVE STACK PTR
CCC
MOV #377,SP ;SET STACK PTR = #1
ROR SP ;ROTATE 8 BIT THROUGH ALL BIT
15i ROR SP ;BIT POSITIONS
;SHOULD INCREMENT SP TO 0
ROR SP
BEQ 25
MOV SP,R2 ;SAVE ERROR STACK PTR
MOV R3,SP ;SET STACK PTR FOR TRAP
HLT ;ERROR!

015370 010306 25i MOV R3,SP ;RESTORE ORIGINAL STACK PTR

ICHECK BYTE OPERATIONS USING THE STACK
SPOHK: MOV SP,R0 ;SAVE STACK PTR
MOV R0,R3
CLR =(R3)
MOV #1,=(SP) ;(SP) = 377
CMP #377,(R3) ;CHECK THAT ONLY EVEN BYTE WAS AFFECTED
BNE 15
CMP R3,SP ;CHECK AUTO-DEC
BEQ ,+4
15i HLT

015420 105226 INCB (SP)+
015422 005723 TST (R3)+ ;CHECK RESULT
015424 001002 BNE 25
015426 020006 CMP R0,SP ;CHECK AUTO-INC
015430 001401 BEQ ,+4
015432 104400 25i HLT

015434 005143 COM =(R3) ;(R3)=17777
015436 144613 R1CB =(SP),(R3)
015440 022713 177400 CMP #177400,(R3) ;CHECK RESULT
015444 001002 BNE 35
015446 020603 CMP SP,R3
015450 001401 BEQ ,+3
015452 104400 35i HLT

015454 132627 000377 BITB (SP)+,#377
015460 001002 BNE 45
015462 020600 CMP SP,R0
015464 001401 BEQ ,+3
015466 104400 45i 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 55
015510 122002 CMPB (SP)+,R2

```

```

015512 001002 BNE 55
015514 020006 CMP R0,SP
015516 001401 BEQ ,+4
015520 104400 55i 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

ICHECK THAT 'C' BIT SETS/CLEAR PROPERLY
CBIT: MOV #17776,(PC)+ ;LOAD CONSTANT
15i WORD 0
MOV PC,R0 ;GET CURRENT PC
SUB #4,R0 ;POINT R0 TO 15 ABOVE
25i ADC (R0)+ ;ADD 'C' BIT TO 15 ABOVE
ASL =(R0) ;SHIFT 15
BVC 25 ;UNTIL 'V' BIT SETS
CMP #07776,15 ;CHECK RESULT
BEQ ,+4
HLT ;ERROR! INCORRECT RESULT IN 15 ABOVE
;R0=ADDRESS OF DATA

ICHECK THAT CONDITION CODES ARE SET PROPERLY WHEN A NUMBER (CURRENT PC)
AND THAT NUMBER #1 ARE COMPARED, AND VICE VERSA.
CMPN: MOV PC,R0 ;GET CURRENT PC
MOV R0,R2 ;SAVE IN R2
INC R2 ;MAKE R2 = R2+1
SCC
*CLC/CLN ;CLEAR C & N BITS
CMP R0,R2 ;COMPARE #1 WITH #1
BCC 15 ;CARRY BIT SHOULD SET
BVS 15 ;V BIT SHOULD CLEAR
BEQ 15 ;Z BIT SHOULD CLEAR
BMI ,+4 ;N BIT SHOULD SET
15i HLT ;ERROR! COMPARE # WITH #1 FAILED TO
;SET CONDITION CODES IN PSW CORRECTLY

015572 010700 SCC ;SET CONDITION CODES IN PSW
015574 010002 CMPB R2,R0 ;COMPARE #1 WITH #
015576 005202 BCS 25 ;C BIT SHOULD CLEAR
015600 000277 BVS 25 ;V BIT SHOULD CLEAR
015602 000251 BEQ 25 ;Z BIT SHOULD CLEAR
015604 020002 BMI ,+4 ;N BIT SHOULD CLEAR
015606 103003 ;ERROR! COMPARE #1 WITH # FAILED TO SET
015610 102402 ;CONDITION CODES IN PSW CORRECTLY
015612 001401
015614 100401
015616 104400

015620 000277 SCC
015622 120200 CMPB R2,R0
015624 103403 BCS 25
015626 102402 BVS 25
015630 001401 BEQ 25
015632 100001 BMI ,+4
015634 104400 25i HLT

```

124 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


```

016276 105737 177570 TSTB ##SWR ;DELETE END OF PASS TYPE OUT IF SW7=0
016302 100020 BPL 15 ;BRANCH IF SW7 IS DOWN
016304 016702 162470 MOV ICNT,R2 ;GET PASS COUNT
016310 004767 163014 JSR PC,FORM0 ;GO TO FORMAT ROUTINE
016314 012702 001664 MOV #DIGITS+2,R2 ;GET ASCII VALUES
016320 012703 001702 MOV #PASSES,R3 ;AND MOVE THEM INTO MESSAGE
016324 012223 MOV (R2)+,(R3)
016326 012223 MOV (R2)+,(R3)
016330 012737 001672 010000 MOV #PASCNT,0#MSG ;PASS MESSAGE ADRS TO TELETYPE SERVICE
016336 002737 000100 177504 BIS #100,0#TPS ;SET IE BIT
016344 012737 000610 000024 1S1 MOV #DOWN,0#PFVEC ;ENABLE POWER FAIL TRAP
016392 012737 000340 000026 MOV #340,0#PFVEC+2 ;PRIORITY 7 ON POWER FAIL
016300 005267 162414 INC ICNT
016364 116700 MOVB OP,CP,R0 ;GET CP TYPE
016370 026067 019570 162402 CMP PASTAB(R0),ICNT ;CHECK IF END OF TEST
016376 001002 BNE 25 ;BRANCH IF NOT AT END
016400 000167 000000 JMP DONE
016404 016702 162370 2S1 MOV ICNT,R2 ;GET PASS COUNT
016410 006302 ASL R2
016412 046002 016500 BIC CPPASS(0),R2 ;LIMIT PASS COUNT TO 0=6
016416 005037 000016 CLR #10 ;CLEAR T BIT TRAP ADDRESS
016422 012737 000040 001122 MOV #40,0#SCOPE+2 ;SET ITERATION COUNT = 40
016430 016216 016554 MOV PSWTAB(2),(SP) ;PUSH NEXT PASS PSW ON STACK
016434 032716 000020 BIT #20,(SP) ;WILL IT' BIT BE SET ON NEXT PASS?
016440 001426 BEO 35 ;BRANCH IF NOT
016442 012737 000002 001122 MOV #2,0#SCOPE+2 ;SET ITERATION COUNT = 2 FOR IT' BIT
016450 016737 000006 000016 MOV RT1,#10 ;SET IT' BIT TRAP TO RETURN VIA 10
016456 012746 002230 3S1 MOV #START2,0(SP) ;RESTART PROGRAM AT START2
016462 000002 RT1 ;RESTART PROGRAM AT START2 WITH NEW PSW
; (FROM TABLE BELOW) NOTE: THE RT1 IS
; CHANGED TO AN RTT IF NOT AN 11/05,11/20

```

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

```

016464 032737 000100 177564 DONE1 BIT #100,0#TPS ;WAIT FOR TTY OUTPUT TO FINISH
016472 001374 BNE DO E
016474 105737 177564 TSTB ##TPS ;WAIT FOR LAST CHARACTER TO BE PRINTED
016500 100375 BPL ,#4
016502 005027 CLR (PC)+
016504 000000 1S1 ,WORD 0
016506 005267 177772 2S1 INC 15 ;DELAY WAITING FOR TELETYPE TO FINISH
016512 001375 BNE 25 ;TYPING CHARACTER BEFORE ISSUING RESET
016514 000005 RESET
016516 105737 177570 TSTB ##SWR
016522 100003 BPL 35
016524 004767 162502 JSR PC,PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
016530 016714 ENDMMSG
016532 013702 000042 3S1 MOV ##42,R2 ;CHECK DDPACT11 MONITOR HOOK
016536 001404 BEO DONE1
016540 004712 LOGICALJSR PC,(R2) ;GO TO DDPACT11 MONITOR VIA 42
016542 000240 NOP
016544 000240 NOP
016546 000240 NOP
016550 000137 002224 DONE11 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 IT' BIT
;IN ALL PASSES PATCH OUT THE IT' BIT IN THE TABLE,
PSWTAB: 000000 ;ALL 11 FAMILY CP'S
016554 000000
016556 000020

;THE BELOW TABLE IS THE 'BIT MASK' USED TO DETERMINE THE INDEX VALUE
;NEEDED TO SET THE 'NEW' PSW,
CPPASS: 177774 ;11/05
016560 177774 ;11/20
016562 177774

;THE BELOW TABLE REPRESENTS THOSE BITS IN THE CP WHICH CAN BE SET/CLEARED
PSWBIT: 000377 ;11/05
016564 000377 ;11/20
016566 000377

;THE BELOW TABLE CONTAINS THE # OF PASSES REQUIRED TO COMPLETE TEST
PASTAB: ,WORD 2 ;11/05
016570 000002 ;11/20
016572 000002

;MESSAGES
MSG11 ,ASCII <15><12>'LOW LIMIT?'
016574 005015 047514 020127
016602 044514 044515 037524
016610 000
016611 110 043511 020110
016616 044514 044515 037524
016624 000
016625 005 020112 044510
016632 020123 042524 051123
016640 044400 053116 041001
016646 042111 043040 051117
016654 030440 027461 030004
016662 030455 027461 032464
016670 050040 042514 051501
016676 020105 052522 020116
016704 041504 045521 005503
016712 000012
016714 005015 042040 050532 ENDMMSG ,ASCII <15><12>' DZQKCD DONE'
016722 041513 042040 047117
016730 000105
000001

;END

```

ADCB2 004564
ADCB 002524
ADCB 005672
ADD1A 010026
ADD6 011450
ASLB3 005364
ASLB 002646
ASLB 005642
ASRB2 004634
ASRB7 007024
ASR3 005102
BICB1 010220
BIC2 010522
BIN1 010400
BIN 007712
BITB1 010176
BIT1 007640
BIT2 010546
C = 000001
CC2 002370
CLR0 002442
CMPN 015572
CMP1A 007742
COMB1A 004200
COMB7 004274
COM4 004276
CPPASS 014560
DDATAB 011742
DECB5 005442
DEC1 003360
DEC6 005656
DISPLA= 177570
EMT1 013724
END 016244
ERRPC 001707
FORXOR 000740
GSYST 003160
INCB1 003704
INCB6A 006176
INC3 005170
IOTVEC= 000020
JMP1 013092
JMP6 013244
JSR1A 013372
JSR4A 013536
LKS = 177546
LPS = 177514
MOV0 007120
MS0 014000
NEGB1 003794
NEG0 002626
NEG6 005544
OAEERR 012616
PASSES 001702

ADCB5 005374
ADCB1 003400
ADCB 006604
ADD1B 010044
ADD7 012112
ASLB4 004670
ASLB 003554
ASLB7 004432
ASRB2A 004652
ASR0 002674
ASR6 005524
BICB1A 010242
BIC3 011120
BISB1 010206
BIS2 010460
BITB2 010736
BIT13 = 020000
BIT6 = 001000
CBIT 013536
CCS 002402
CMPB1 010192
CMP0 007170
CMP2 010450
COMB2 004546
COM0 002506
COM6 005510
DBINB7 012332
DECB1 003770
DECB6A 006214
DECB1A 003630
DEC7 006450
DONE 014464
EMTB 014004
ENDMSG 016714
ERRPC0 001721
FPEVEC= 002244
HLT = 104400
INCB2 004776
INCB7 006756
INC4 004326
IOT1 013624
JMP3 013122
JMP7 013302
JSR3 013456
JSR6A 013570
LKVEC = 002100
LSTMEM 002140
MOV0A 007150
MSG1 019574
NEGB4 004726
NEG1 003620
NEG7 005524
OPT_CP 014254
PASTAB 010570

ADCB6 006002
ADCB 001374
ADD0 007376
ADD2 010432
ASLB1 003742
ASLB6 006044
ASLB3 002116
ASRB1 004036
ASRB5 005324
ASR1 003442
ASR7 002466
BIC0 007310
BIC7 012570
BIS0 007266
BIS2A 012562
BITB3 011270
BIT4 = 040000
BIT8 = 000400
CC0 005340
CC4 004416
CMPB2 010722
CMPB0A 007432
CMP7 012046
COMB5 003302
COM1 003566
COM7 006556
DBIN7 011746
DECB1A 003120
DECB7 006772
DEC2 004444
DIGITS 001602
DONE1 010550
EMT1C 011010
END1 016246
ERRVEC= 000004
FRSTAD 001010
ICNT 001000
INCB3 007334
INC0 002606
INC6 005620
IOT1A 013632
JMP4 013100
JSRST 013306
JSR3A 013460
KPTR = 000000
LOGICA = 016540
HMVEC = 00 290
MOV1 010012
MSG2 019611
NEGB6 005132
NEG2 004390
NULL = 001000
OVPLW 014332
PC =X000007

ADCB7 006710
ADCB 005202
ADD1 007602
ADD3 011106
ASLB1A 004106
ASLB7 007000
ASLB 004466
ASRB1A 004092
ASRB6 006142
ASR2 004410
BELL 001747
BIC1 007724
BINB7 012336
BIS0A 007344
BIS7 012530
BITB6 011694
BIT15 = 100000
BPTVEC= 000014
CC1 002354
CHKSP 015342
CMPB3 011302
CMP1 007622
COMB1 004020
COMB6 006114
COM3 005100
CPCHK 014204
DDATA 011312
DECB2 004702
DEC0 002566
DECS 005134
DIGTAB 001652
EMTVEC= 000030
EMT10 014022
ERROR 001412
FACTOR 001004
FRSTME 001012
ILLTES 016622
INCB6 006026
INC1 003474
INC7 006616
IOT1B 013646
JMP5 013224
JSR1 013370
JSR4 013534
KW11 010070
LFB = 177516
MOVB1 010126
MOV7 012016
N = 000010
NEGB7 007040
NEG5 005146
NULLS 010064
PASCNT 001672
PDWN 006610

PFALL 000656
PRY4 = 000200
PSWB1 016564
REQD 001792
RELI 006302
REBVEC= 000010
ROLB3 005410
ROL0 002660
ROL4 004424
RORB1A 004074
RORB7 006742
ROR2 004312
RT11 014402
R11 =X000001
R15 =X000005
R5 =X000005
SBCB6 004002
SBC1A 003526
SBB1B7 012322
SCOPEB 001046
SCOPEF 001120
SP =X000006
START1 002100
STKPTR= 000500
SUB1B 010002
SUB3A 011094
SWAB1 004134
SWAB7 006540
TICKS 000776
TP8 = 177564
TRAP1C 014136
TSB2A 004746
TSB2 004266
UBREAK= 17770
UH7 006336
FILLS 001002
PRINT 001232

PFVEC = 000024
PRTY6 = 000300
PSWCHK 013124
RELOC 001132
REL2 012274
RETPC 001230
ROLB6 006146
ROL1 003414
ROL6 003474
RORB4 004602
ROR0 002544
ROR5 005070
R0 =X000000
R12 =X000002
R2 =X000002
SBCB1 003730
SBCB7 006670
SBC5 005230
SBB17 011744
SCOPEC 001102
SDATA 011310
SPCHK 015372
START2 002230
SUB0 007136
SUB2 010476
SUB6 011470
SWAB2 004340
SNR = 177570
TKB = 177562
TRVEC = 000064
TRTVEC= 000014
TSTB6 003794
TST0 006256
UB7 011754
V = 000002
SFORM0 001330
 = 016732

PIRO = 177772
PRY7 = 000340
PSWTAB 012554
RELR1 001006
REL3 014306
ROLB1 003716
ROLB6A 005242
ROL1A 004426
ROL7 006634
RORB5 005392
ROR1 003344
ROR6 005504
R1 =X000001
R13 =X000003
R3 =X000003
SBCB3 003430
SBC0 004710
SBC6 005600
SCOPE = 104000
SCOPE0 001114
SDATAB 011740
SR0 = 177572
START3 002224
SUB1 007600
SUB2A 010616
SUB7 012066
SWAB4 001700
T = 000020
TKS = 177560
TRAPVE= 000034
TSB1 004446
TST0 002444
TTVCHK 013734
UHM6 005450
Z = 000004
SRESTR 000240

PIRVEC= 000240
PSW = 177776
PUP 000620
RELB 002272
RESTRP 014674
ROLB2 004620
ROLB7 007056
ROL3 005216
RORB1 004004
RORB6 006100
ROR1A 003456
ROR7 006500
R10 =X000000
R14 =X000004
R4 =X000004
SBCB4 004714
SBC1 003510
SBC7 006412
SCOPEA 001016
SCOPEE 001126
SLR = 177774
START 000066
STATUS 001714
SUB1A 007706
SUB3 011032
SWAB0 002726
SWAB6 006226
SBITVE= 000014
TPB = 177566
TRAP1 014112
TSTB2 004736
TST1 003604
UBM6 006262
UHM7 006332
SCRLF 001744
SSAVR 000214

ERRORS DETECTED: 0/2

*DEQKCD,DEQKCB/SOL=DEQKCD
RUN=TIME: 12 21 0 SECONDS
CORE USED: 7K