

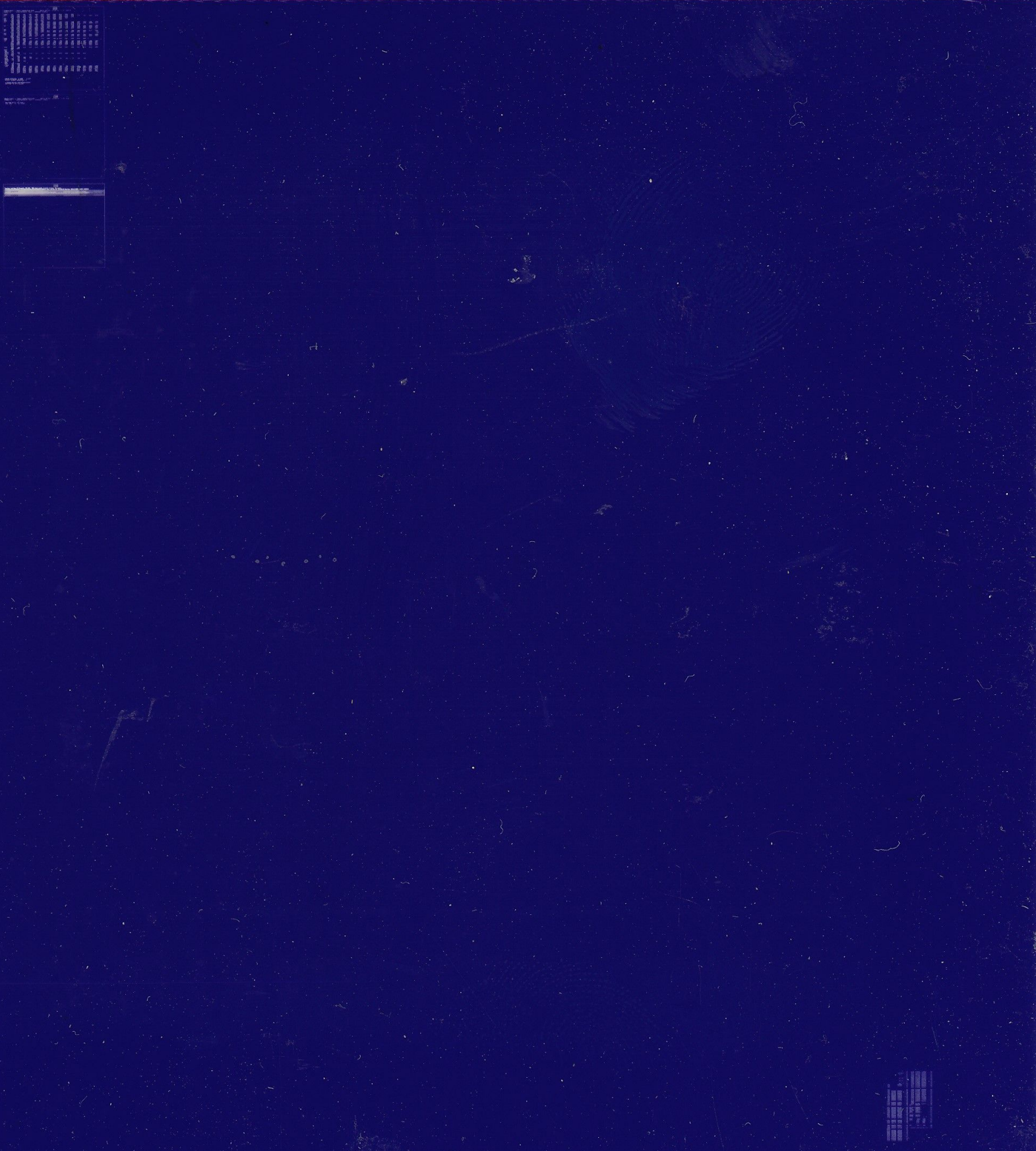
PDP11

INSTRUCTION EXERCISER
MD-11-DDQAA-A

EP-DDQAA-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

This section contains a grid of 100 small tables, arranged in 10 rows and 10 columns. Each table is a technical reference for a specific instruction or operation. The tables contain various data points, including instruction codes, register names, and bit patterns. The text is small and dense, typical of a technical manual's reference section. The tables are organized in a regular grid, with each cell containing a small table of data.



1.0

ABSTRACT

THIS DIAGNOSTIC PROGRAM IS DESIGNED TO BE A COMPREHENSIVE CHECK OF THE PDP11/05 AND PDP11/10 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 11/10 OR 11/05. THE PROGRAM RELOCATES THE TEST CODE THROUGHOUT MEMORY 0-28K.

THE PROGRAM DDQAAA HAS BEEN CREATED BY MODIFYING THE PROGRAM DZOKCE. FOLLOWING SECTION 2 THERE IS A ROUTINE --"CHECK THAT ALL RESERVED INSTRUCTIONS TRAP"-- WHICH 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.----- THIS ROUTINE, EVENTHOUGH QUITE HARMLESS FOR THE SEDM SYSTEM, COULD HAVE BEEN DELETED TO MAKE THE PROGRAM EXCLUSIVE FOR THE SEDM SYSTEM. PLEASE IGNORE THIS ROUTINE AND ANY DEFINITION ETC. WHICH IS RELATED TO IT, WHEN RUNNING ON THE SEDM SYSTEM.

2.0

REQUIREMENTS

2.1

EQUIPMENT

PDP11 FAMILY CENTRAL PROCESSOR
OPTIONAL - K111-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 VTOS FILLER CHARACTERS MAY BE REQUIRED. DEPOSIT INTO LOCATION 1002 (FILLS) A 4400.

1. LOAD ADDRESS = 200
 2. SET SW15 UP
 3. SET SW12 UP --- FOR SEDM SYSTEM ONLY
 4. SET SW06 UP --- OPTIONAL, FOR END OF PASS HALT
 5. PRESS START
 - E. SET OTHER OPERATING SWITCHES, AS DESIRED
- PASS COUNT IS PRINTED AFTER EACH PASS (SEE SEC 6.4).
"DDQAA DONE" IS PRINTED WHEN DONE (SEE SEC 7.1).
THERE WILL BE A 5 ON THE DISPLAY LIGHTS FOR A

FEW SECONDS AFTER EACH PASS AND THEN PROGRAM
WILL HALT IF SW06 WAS UP.
PRESS CONTINUE FOR ANOTHER PASS.

4.0

SWITCH SETTINGS

SW15 HALT ON ERROR... THIS SWITCH WHEN SET WILL HALT THE
PROCESSOR, AT LOCATION 1664, WHEN AN ERROR IS DETECTED. THE PC, THE
CURRENT STATUS, AND THE PASS COUNT AT THE TIME OF
THE ERROR, IS STORED IN CORE STARTING AT LOCATION
017400.

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 THIS SWITCH WHEN RESET (0) INHIBITS THE END OF PASS TYPEOUT (ICNT=XXXX) AND THE END OF PROGRAM TYPEOUT (DDQAAA DONE).

SW6 WHEN SET HALTS THE PROCESSOR ON END OF PASS. PRESS CONTINUE FOR ANOTHER PASS.

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 PROCESSOR STATUS AT THE TIME OF THE ERROR. ALSO (IF REQUIRED) THE ORIGINAL PC (WHERE THE PC WAS RELOCATED FROM). PROGRAM WILL HALT AT LOCATION 1664, IF SW15 WAS UP.

TO DETERMINE TYPE OF ERROR:
(IF THERE IS NO TTY WITH THE SYSTEM)

1. LOAD ADDRESS 017400.
2. PRESS EXAMINE --- CONTENT IS ERROR PC.
3. PRESS EXAMINE --- CONTENT IS PSW.
4. PRESS EXAMINE --- CONTENT IS PASS COUNT.

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

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).

5.2

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.

F01

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 5

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

H01

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY!! 27(732) 22-SEP-76 14:39 PAGE 7

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

;000000000000FIRST ADDRESS TO BE RELOCATED0000000000

CODE TO BE MOVED AND EXECUTED

;000000000000LAST 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.

J01

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 9

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

RESERVED INSTRUCTION, AND ODD ADDRESS TRAPS.

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

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

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

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

8.1

STACK POINTER

THE STACK POINTER IS SET AT 500.

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

8.2

POWER FAILURE

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

POWER FAILED

AND RESTART THE PROGRAM AT THE BEGINNING. (START)

9.0

USER DEFINED RELOCATION LIMITS

THE PROGRAM WILL REQUEST A LOWER AND UPPER LIMIT FOR RELOCATION. THE LIMITS MUST BE BETWEEN 20070 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.

359
360
361
362
363
354
365
366

.NLIST MD,MC
.LIST ME
.ABS
.TITLE FRONT END
;CONTAINS DEFINITIONS, REGISTER ASSIGNMENTS AND MACRO CALLS
;GENERAL REGISTER ASSIGNMENTS
RO=%0

000000

367	000001	R1=%1
368	000002	R2=%2
369	000003	R3=%3
370	000004	R4=%4
371	000005	R5=%5
372	000006	SP=%6
373	000007	PC=%7
374	000000	R10=%0
375	000001	R11=%1
376	000002	R12=%2
377	000003	R13=%3
378	000004	R14=%4
379	000005	R15=%5

;STATUS REGISTER (PSW) BIT ASSIGNMENTS

384	000001	C=1	;C BIT
385	000002	V=2	;V BIT
386	000004	Z=4	;Z BIT
387	000010	N=10	;N BIT
388	000020	T=20	; 'T' BIT
389	000340	PRTY7=340	;PRIORITY LEVEL 7
390	000300	PRTY6=300	;PRIORITY LEVEL 6
391	000200	PRTY4=200	;PRIORITY LEVEL 4

;VECTOR ADDRESSES

394	000004	ERRVEC=4	;ADDRESS OF ERROR VECTOR
395	000010	RESVEC=10	;ADDRESS OF RESERVED INST. TRAP VECTOR
396	000014	TBITVEC=14	;ADDRESS OF 'T' BIT TRAP VECTOR
397	000014	TRTVEC=14	;ADDRESS OF 'TRACE' TRAP VECTOR
398	000014	BPTVEC=14	;ADDRESS OF 'BREAKPOINT' TRAP VECTOR
399	000020	IOTVEC=20	;ADDRESS OF IOT TRAP VECTOR
400	000024	PFVEC=24	;ADDRESS OF POWER FAIL TRAP VECTOR
401	000030	EMTVEC=30	;ADDRESS OF EMT VECTOR
402	000034	TRAPVEC=34	;ADDRESS OF TRAP VECTOR
403	000064	TPVEC=64	;ADDRESS OF TTY PRINTER INTERRUPT VECTOR
404	000100	LKVEC=100	;ADDRESS KW11-L LINE CLOCK INT. VECTOR
405	000240	PIRVEC=240	;ADDRESS OF PIRQ VECTOR
406	000244	FFEVEC=244	;ADDRESS OF FLOATING POINT INT. VECTOR
407	000250	MMVEC=250	;ADDRESS OF MEM MGMT ERROR TRAP VECTOR

;REGISTER ADDRESSES

410	177776	PSW= 177776	;ADDRESS OF STATUS REGISTER
411	177774	SLR= 177774	;ADDRESS OF STACK LIMIT REGISTER
412	177772	PIRQ= 177772	;ADDRESS OF PROGRAM INTERRUPT REQUEST
413	177770	UBREAK= 177770	;ADDRESS OF MICRO BREAK REGISTER
414	177546	LKS= 177546	;ADDRESS OF KW11-L STATUS REG.
415	177560	TKS= 177560	;ADDRESS OF KEYBOARD CSR
416	177562	TKB= 177562	;ADDRESS OF KEYBOARD BUFFER
417	177564	TPS= 177564	;ADDRESS OF TELEPRINTER CSR
418	177566	TPB= 177566	;ADDRESS OF TELEPRINTER BUFFER
419	177572	SRO= 177572	;ADDRESS OF MEM MGMT REGISTER SRO
420	177570	SWR= 177570	;ADDRESS OF CONSOL SWITCH REGISTER
421	177570	DISPLAY=177570	;ADDRESS OF CONSOL DISPLAY REGISTER
422	177514	LPS= 177514	;ADDRESS OF LINE PRINTER STATUS REG

MO1

FRONT END
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 12

```

423          177516          LPB= 177516          ;ADDRESS OF LINE PRINTER DATA DUFFER
424
425          ;INITIAL STACK POINTER SETTING
426          000500          STKPTR= 500          ;PROGRAM STACK PTR
427          000600          KPTR=600          ;KERNEL STACK PTR (USED BY KERNEL WHEN
428                                     ;PROGRAM IS RUNNING IN OTHER THAN KERNEL
429                                     ;MODE (NOT APPLICABLE TO 11/05,11/20)
430
431          ;MISCELLANEOUS BIT ASSIGNMENTS
432          100000          BIT15=100000
433          040000          BIT14=40000
434          020000          BIT13=20000
435          000400          BIT8=400
436          000100          BIT6=100
437
438          ;SWITCHES
439
440          000100          SW06=100
441          ;INSTRUCTION EQUATES
442          104400          HLT=TRAP          ;HLT IS A TRAP INST TO THE ERROR ROUTINE
443          104000          SCOPE=EMT          ;SCOPE IS AN EMT TRAP
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460          000046          .=46
461          000046          016612          LOGICAL
462          000052          .=52
463          000052          040000          BIT14
464          000200          000200          .=200
465          000200          012707          002102          MOV      #START,PC          ;GO TO START OF TEST
466          000204          012707          002174          MOV      #START1,PC          ;GO GET LOWER/UPPER RELOF ION BOUNDARY
467          000210          012707          002240          MOV      #START3,PC          ;START WITH LAST TYPED F IDARY LIMITS
468
469          ;ROUTINE TO SAVE REGISTERS ON THE STACK
470          ;CALLED BY SAVE MACRO OR JSR PC,$SAVR
471          000214          012667          000016          $SAVR: MOV      (SP)+,1$          ;SAVE RETURN PC
472          000220          010546          MOV      %5,-(SP)
473          000222          010446          MOV      %4,-(SP)
474          000224          010346          MOV      %3,-(SP)
475          000226          010246          MOV      %2,-(SP)
476          000230          010146          MOV      %1,-(SP)
477          000232          010046          MOV      %0,-(SP)
478          000234          012707          MOV      (PC)+,PC          ;RETURN

```

NO1

FRONT END
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 13

```

479 000236 000000      1$:      0      ;CONTAINS RETURN ADDRESS
480      000250 000250      . =250
481 000250 000000      EOPHLT: HALT      ; THIS IS AN END OF PASS HALT;
482      ; NOT AN ERROR HALT. YOU GET HERE
483      ; ONLY IF SW06 IS UP. PRESS
484      ; CONTINUE TO CONTINUE.
485 000252 000207      RTS      PC
486
487      ; ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
488      ; CALLED BY RESTORE MACRO OR JSR PC, $RESTR
489 000254 012667 000016  $RESTR: MOV      (SP)+, 1$      ; SAVE RETURN PC
490 000260 012600      MOV      (SP)+, %0
491 000262 012601      MOV      (SP)+, %1
492 000264 012602      MOV      (SP)+, %2
493 000266 012603      MOV      (SP)+, %3
494 000270 012604      MOV      (SP)+, %4
495 000272 012605      MOV      (SP)+, %5
496 000274 012707      MOV      (PC)+, PC      ; RETURN
497 000276 000000      1$:      0      ;CONTAINS RETURN ADDRESS
498
499      000610      . =610
500      ; POWER FAIL SUBROUTINE
501 000610 012737 000620 000024 PDWN:  MOV      #PUP, 2$PFVEC
502 000616 000000      HALT
503
504      ; POWER UP SUBROUTINE
505 000620 012737 000610 000024 PUP:   MOV      #PDWN, 2$PFVEC      ; RESTORE POWER FAIL TRAP TO POWER
506      ; DOWN ROUTINE ABOVE
507      MOV      #KPTR, SP      ; SET STACK PTR
508 000632 005027      CLR      (PC)+
509 000634 000000      1$:      WORD      0      ; KILL TIME
510 000636 005267 177772      2$:      INC      1$
511 000642 001375      BNE      2$
512 000644 004767 000362      JSR      PC, .PRINT      ; PRINT MESSAGE BEGINING AT FOLLOWING ADRS
513 000650 000656      PFAIL
514 000652 000137 002102      JMP      2$START      ; RESTART TEST
515
516 000656 005015 047520 042527 PFAIL: .ASCIZ <15><12>'POWER FAILED'<15><12>
517 000664 020122 040506 046111
518 000672 042105 005015      000
519
520      000740      . =740
521      ; NOTE: THIS CODE USED ONLY BY THE XOR TESTER.
522      ; TO USE CODE PLACE 776 (BR .-2) IN SCOPE
523 000740 012737 000002 000006 FORXOR: MOV      #RTI, 2$ERRVEC+2 ; SET TIME OUT TRAP TO RETURN
524 000746 000261      SEC      ; SET C
525 000750 005737 177060      TST      2$177060      ; IF A TIME OUT OCCURS THEN WHEN NEXT
526      ; INSTRUCTION IS EXECUTED 'C' WILL BE SET
527      ; AND IF NO TIME OUT 'C' WILL BE CLEARED
528 000754 103401      BCS      1$      ; BRANCH IF 'C' SET (TIMED OUT)
529 000756 011601      MOV      (SP), R1      ; ADDRESS OF NEXT SUBTEST TO R1
530 000760 005037 000006 1$:      CLR      2$ERRVEC+2      ; RESTORE TIME OUT TRAP
531 000764 010116      MOV      R1, (SP)      ; GET RETURN ADDRESS BACK TO SUBTESTS
532 000766 000240      NOP
533 000770 000002      RTI
534      ; RETURN EITHER TO LAST OR NEXT SUBTEST

```



```

535      000776 000776
536 000776 000000 TICKS: .WORD 0 ;CONTAINS CLOCK TICK COUNT
537      001000 001000          .=1000
538 001000 000000 ICNT: 0 ;CONTAINS PASS COUNT
539 001002 000000 $FILLS: .WORD 0 ;CONTAINS FILLS COUNT IN ODD BYTE
540      001004 000000          ;AND FILLER CHARACTER IN EVEN BYTE
541 001004 000000 FACTOR: 0 ;CONTAINS RELOCATION FACTOR
542      001006 000000          ;SUBTRACT # IN FACTOR FROM PC TO GET PC OF ORIGINAL CODE
543 001006 000000 RELR1: 0 ;CONTAINS RELOCATED R1 (THE R1 OF THE
544      001010 000000          ;ORIGINAL CODE MOVED)
545 001010 000000 FRSTAD: .WORD 0 ;CONTAINS FIRST ADRS OF CODE TO BE MOVED
546 001012 000000 FRSTMEN: .WORD 0 ;CONTAINS LOWER RELOCATION BOUNDARY ADDRESS
547 001014 000751          BR FORXOR ;BRANCH TO XOR TESTER CODE
548      ;SCOPE (EMT) SERVICE ROUTINE
549      ;THIS ROUTINE ALLOWS THE SUBTEST TO BE CONTINUOUSLY LOOPED, ITERATED
550      ;(OR NOT ITERATED) BEFORE BEGINNING NEXT SUBTEST
551 001016 000240 SCOPEA: NOP
552 001020 032766 004000 000002 BIT #4000,2(SP) ;WAS REGISTER SET BIT SET ON TRAP
553 001026 001403 BEQ 2$ ;BRANCH IF NOT
554 001030 052737 004000 177776 BIS #4000,2#PSW ;RETAIN REGISTER SET
555 001036 032737 040000 177570 2$: BIT #40000,2#SWR ;CHECK BIT 14 (CONTINUOUS LOOP)
556 001044 001416 SCOPEC: BEQ SCOPEC
557 001046 010116 SCOPEB: MOV R1,(SP) ;LOAD RETURN ADDRESS
558 001050 010137 001006 MOV R1,2#RELR1
559 001054 163737 001004 001006 SUB 2#FACTOR,2#RELR1 ;RELR1 CONTAINS UNRELOCATED R1
560 001062 032737 000400 177570 BIT #400,2#SWR ;LOAD PDP11/45 MICRO BREAK REG?
561 001070 001403 BEQ 1$
562 001072 113737 177570 177770 MOVB 2#SWR,2#UBREAK ;LOAD MICRO BREAK REG WITH SRO-7
563 001100 000002 1$: RTI ;RETURN TO SUBTEST
564 001102 032737 004000 177570 SCOPEC: BIT #4000,2#SWR ;SUBTEST ITERATION DESIRED?
565 001110 001006 BNE SCOPEE ;BRANCH IF NO ITERATION DESIRED?
566 001112 005327 DEC (PC)+ ;DECREMENT SUBTEST ITERATION COUNT
567 001114 000040 SCOPED: 40 ;CONTAINS SUBTEST ITERATION COUNT
568 001116 001353 BNE SCOPEB
569 001120 012767 000040 177766 SCOPEF: MOV #40,SCOPED ;RESET ITERATION COUNT
570 001126 011601 SCOPEE: MOV (SP),R1 ;GET ADDRESS OF NEXT TEST
571 001130 000746 BR SCOPEB
572
573 ;ROUTINE TO RELOCATE PROGRAM CODE
574 001132 032737 010000 177570 RELOC: BIT #10000,2#SWR ;CHECK IF RELOCATION DESIRED (BIT12)
575 001140 001031 BNE 3$ ;BRANCH IF NO RELOCATION DESIRED
576 001142 013700 001010 MOV 2#FRSTAD,R0 ;GET FIRST ADDRESS OF CODE TO BE MOVED
577 001146 010005 MOV R0,R5 ;SAVE
578 001150 010204 MOV R2,R4 ;GET LAST ADDRESS OF CODE TO BE MOVED
579 001152 160504 SUB R5,R4 ;R4 CONTAINS # OF WORDS TO RELOCATE
580 001154 010203 MOV R2,R3 ;SAVE LAST ADDRESS OF CODE TO BE MOVED
581 001156 005737 001004 TST 2#FACTOR ;FIRST RELOCATION IS TO 20000
582 001162 001004 BNE 10$
583 001164 010237 001230 MOV R2,2#RETPC ;SAVE RETURN PC TO NEXT SECTION OF CODE
584 001170 013702 001012 MOV 2#FRSTMEN,R2 ;SET FIRST ADDRESS
585 001174 060204 10$: ADD R2,R4 ;R4 CONTAINS LAST MEMORY ADDRESS
586      ;TO BE USED
587 001176 020437 002154 CMP R4,2#LSTMEN ;CHECK IF SUFFICIENT MEMORY REMAINS
588 001202 101011 BHI 4$
589 001204 012022 1$: MOV (R0)+,(R2)+ ;RELOCATE PROGRAM CODE
590 001206 020003 CMP R0,R3 ;CHECK IF DONE
    
```

```

591 001210 001375          BNE      1$
592 001212 024042          2$:    CMP      -(R0),-(R2)      ;CHECK THAT CODE WAS RELOCATED
593 001214 001401          BEQ      .+4                ;PROPERLY
594 001216 104400          HLT
595 001220 020005          CMP      R0,R5            ;ERROR! CODE NOT RELOCATED PROPERLY
596 001222 001373          BNE      2$                ;CHECK IF FINISHED CHECKING
597 001224 010207          3$:    MOV      R2,PC
598 001226 011707          4$:    MOV      (PC),PC      ;GO EXECUTE RELOCATED CODE
599 001230 000000          RETPC: 0                  ;RETURN TO NEXT SECTION OF CODE
600
601                                ;ROUTINE TO PRINT ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
602 001232 010046          .PRINT: MOV     RO,-(SP)      ;SAVE RO ON THE STACK
603 001234 017600 000002    MOV     @2(SP),RO          ;GET MESSAGE ADDRESS
604 001240 062766 000002 000002  ADD     #2,2(SP)          ;ADJUST RETURN PC
605
606 001246 112046          1$:    MOV     (RO)+,-(SP)      ;PUSH CHAR ON THE STACK
607 001250 001003          BNE      2$                ;BRANCH IF NOT TERMINATOR
608 001252 005726          TST     (SP)+              ;POP TERMINATOR OFF THE STACK
609 001254 012600          MOV     (SP)+,RO          ;RESTORE RO
610 001256 000207          RTS     PC                 ;RETURN
611
612 001260 004767 000026    2$:    JSR     PC,5$          ;TYPE CHARACTER
613 001264 122726 000012    3$:    CMP     #12,(SP)+      ;CHECK IF CHAR WAS A LINE FEED
614 001270 001366          BNE      1$                ;BRANCH IF NOT LINE FEED
615
616 001272 016746 177504    MOV     $FILLS,-(SP)      ;GET # OF FILLERS REQUIRED AFTER
617                                ;LINE FEED AND FILLER CHARACTER
618 001276 105366 000001    4$:    DECB   1(SP)           ;DECREMENT FILLERS COUNT
619 001302 002770          BLT     3$                ;BRANCH IF NO MORE FILLERS NEEDED
620 001304 004767 000002    JSR     PC,5$            ;TYPE FILLER CHARACTER
621 001310 000772          BR      4$
622
623 001312 105737 177564    5$:    TST     @#TPS          ;WAIT FOR OUTPUT DEVICE
624 001316 100375          BPL     .-4                ;TO BECOME READY
625 001320 116637 000002 177566  MOV     2(SP),@#TPB      ;TYPE CHARACTER
626 001326 000207          RTS     PC
627
628                                NULL=0
629                                ;ROUTINE TO PLACE ASCII VALUE OF AN ADDRESS IN TO ADDRESS MESSAGE
630                                $FORMO:
631 001330 004767 176660    JSR     PC,$SAVR          ;GO SAVE REGISTERS ON THE STACK
632 001334 012704 001676    MOV     #DIGITS,R4        ;ADDRESS WHERE ASCII VALUES ARE STORED
633 001340 005003          CLR     R3                ;WORKING & INDEX REGISTER
634 001342 010201          MOV     R2,R1            ;SAVE
635 001344 006302          1$:    ASL     R2,R1            ;FIRST DIGIT TO R3
636 001346 006103          ROL     R3
637 001350 012700 000006    MOV     #6,RO            ;DIGIT COUNT
638 001354 000404          BR      3$                ;PRINT FIRST DIGIT
639 001356 006302          2$:    ASL     R2
640 001360 006103          ROL     R3
641 001362 005301          DEC     R1
642 001364 001374          BNE      2$
643 001366 012701 000007    3$:    MOV     #3,R1          ;DIGIT SHIFT COUNT
644 001372 116324 001666    MOV     DIGTAB(3),(4)+    ;LOAD DIGIT INTO MESSAGE
645 001376 005003          CLR     R3                ;CLEAR INDEX
646 001400 005300          DEC     RO                ;DEC DIGIT COUNT

```



```

703          :DIGIT TABLE
704 001666 030460          DIGTAB: *01
705 001670 031462          *23
706 001672 032464          *45
707 001674 033466          *67
708 001676 030060 030060 030060 DIGITS: .ASCIZ '000000 '
709 001704 000040
710 001706 005015          PASCNT: .ASCII <15><12>
711 001710 044440 047103 036524          .ASCII ' ICNT='
712 001716 030060 030060 000          PASSES: .ASCIZ '0000'
713 001723 040 041520 000075          ERRPC: .ASCIZ ' PC='
714 001730 051520 036527 000          STATUS: .ASCIZ 'PSW='
715 001735 120 020103 042522          ERRPCD: .ASCIZ 'PC RELOCATED FROM '
716 001742 047514 040503 042524
717 001750 020104 051106 046517
718 001756 000040
719 001760 005015 000          $CRLF: .ASCIZ <15><12>
720 001763 007 000          BELL: .ASCIZ <7>
721          001766          .EVEN
722
723          ;ROUTINE TO GET TYPED OCTAL ADDRESS AND CONVERT TO OCTAL. CALL:
724          ;
725          JSR RS,RECD
726 001766 010046          RECD: .WORD 0          ; CONVERTED DATA IS PLACED HERE
727 001770 005015          MOV RO,-(SP)          ; SAVE RO ON THE STACK
728 001772 105737 177560          CLR (R5)          ; CLEAR OLD DATA
729 001775 100375          IS: TSTB @TKS          ; WAIT FOR USER TO TYPE CHARACTER
730 002000 113700 177562          BPL IS
731 002004 042700 000200          MOVB @TKB,RO          ; GET CHARACTER
732 002010 122700 000177          BIC #200,RO          ; STRIP MSB
733 002014 001010          CMPB #177,RO          ; CHECK IF RUBOUT
734 002016 112737 000134 177566          BNE 2$          ; BRANCH IF NOT RUBOUT
735 002024 100241          MOVB #' \ ,@TPB          ; TYPE \
736 002026 006015          CLC          ; CLEAR CARRY
737 002030 006215          ROR (R5)          ; SHIFT LAST TYPED CHARACTER
738 002032 006215          ASR (R5)          ; OUT OF DATA WORD
739 002034 000756          ASR (R5)
740          BR 1$          ; GO WAIT FOR NEXT CHARACTER
741 002036 110037 177566          2$: MOVB RO,@TPB          ; ECHO CHARACTER TYPED
742 002042 122700 000015          CMPB #15,RO          ; CHECK IF CARRIAGE RETURN
743 002046 001005          BNE 3$          ; BRANCH IF NOT CARRIAGE RETURN
744 002050 004767 177156          JSR PC,.PRINT          ; PRINT MESSAGE BEGINING AT FOLLOWING ADRS
745 002054 001760          $CRLF
746 002056 005725          TST (R5)+          ; STEP RETURN ADDRESS
747 002060 000205          RTS R5          ; RETURN
748
749 002062 042700 177770          3$: BIC #177770,RO          ; STRIP NON-ESSENTIAL BITS
750 002066 006315          ASL (R5)          ; SHIFT LAST CHARACTER 3 PLACES
751 002070 006315          ASL (R5)          ; LEFT
752 002072 006315          ASL (R5)
753 002074 050015          BIS RO,(R5)          ; AND INSERT NEW CHARACTER
754 002076 000735          BR 1$          ; WAIT FOR NEXT CHARACTER
755
756
757 002100 000002          RTI          ; RETURN

```

```

TITLE DDQAA-A BASIC II FAMILY INSTRUCTION EXER.

758
759
760 002102 005037 177776      START: CLR      2#PSW      ;KERNEL MODE
761 002106 005000              CLR      R0          ;CLEAR R0-R5
762 002110 005001              CLR      R1
763 002112 005002              CLR      R2
764 002114 005003              CLR      R3
765 002116 005004              CLR      R4
766 002120 005005              CLR      R5
767 002122 012706 000600      MOV      #KPTR,SP    ;SET KERNEL STACK PTR
768
769 ;ROUTINE TO DETERMINE LAST MEMORY ADDRESS
770 002126 012737 002146 000004  MOV      #15,2#ERRVEC
771 002134 005037 000006      CLR      2#ERRVEC+2
772 002140 005000              CLR      R0
773 002142 005720              TST      (R0)+       ;WILL TIME OUTWHEN END OF MEMORY
774 002144 000776              BR       #-2
775 002146 162700 000002      1$: SUB      #2,R0
776 002152 010027              MOV      R0,(PC)+   ;SET VALUE INTO LSTMEM
777 002154 000000              LSTMEM: .WORD 0    ;CONTAINS VALUE OF LAST MEMORY ADDRESS
778 002156 162737 004000 002154  SUB      #4000,2#LSTMEM ;SET PROTECTION FOR LOADERS
779 002164 012737 020000 001012  MOV      #20000,2#FRSTMEM ;SET LOWER BOUNDARY AT 20000
780 002172 000422              BR       START3    ;GO TO START 3
781 002174
782 002174 004767 177032      START1: JSR      PC,.PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
783 002200 016646              MSG1
784 002202 004567 177560      JSR      R5,RECD    ;GET LOWER LIMIT
785 002206 000000              1$: .WORD 0        ;CONTAINS TYPED LOWER LIMIT
786 002210 016737 177772 001012  MOV      1$,2#FRSTMEM ;SET IN LOWER LIMIT
787 002216 004767 177010      JSR      PC,.PRINT ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
788 002222 016663              MSG2
789 002224 004567 177536      JSR      R5,RECD    ;GET UPPER LIMIT
790 002230 000000              2$: .WORD 0        ;CONTAINS UPPER LIMIT
791 002232 016737 177772 002154  MOV      2$,2#LSTMEM
792
793 002240 005037 001000      START3: CLR      2#ICNT    ;CLEAR PASS COUNT
794 002244 012737 000006 000004  START2: MOV      #ERRVEC+2,2#ERRVEC ;SET ERROR TRAP TO HALT AT 6
795 002252 012706 000500              MOV      #STKPTR,SP ;SET STACK PTR
796 002256 013737 001000 177570      MOV      2#ICNT,2#DISPLAY ;DISPLAY PASS COUNT
797 002264 012737 001016 000030      MOV      #SCOPEA,2#EMTVEC ;SET EMT(SCOPE) TRAP VECTOR
798 002272 012737 001412 000034      MOV      #ERROR,2#TRAPVEC ;SET TRAP (HLT) VECTOR
799 002300 012737 000200 000036      MOV      #200,2#TRAPVEC+2 ;PRIORITY LEVEL 4 ON TRAP
800
801 ;0000000000000000 FIRST ADDRESS TO BE RELOCATED 00000000
802 002306 010700      RELO: MOV      PC,R0    ;GET PC
803 002310 005740              TST      -(R0)      ;R0 CONTAINS THE ADDRESS OF RELO
804 002312 010037 001010      MOV      R0,2#FRSTAD ;SAVE
805 002316 010700              MOV      PC,R0      ;GET CURRENT PC
806 002320 162700 002320      SUB      #.,R0      ;SUBTRACT RELOCATION FACTOR
807 002324 010037 001004      MOV      R0,2#FACTOR ;SAVE RELOCATION FACTOR
808 002330 010701              MOV      PC,R1      ;SET NEW SCOPE PTR
809 ;CHECK BRANCH INSTRUCTIONS
810 002332 000257              CCC              ;CC'S=0000
811 002334 103407              BCS      CCO        ;SAME AS BLO
812 002336 102406              BVS      CCO
813 002340 001405              BEQ      CCO
    
```

814	002342	100404	BMI	CC0	
815	002344	002403	BLT	CC0	
816	002346	003402	BLE	CC0	
817	002350	101401	BLOS	CC0	
818	002352	101001	BHI	.+4	
819	002354	104400	HLT		;ONE OF THE ABOVE BRANCHES FAILED
820					
821					;CONTINUE
822	002356	000270	SEN		;CC'S=1000
823	002360	100003	BPL	CC1	
824	002362	002002	BGE	CC1	
825	002364	003001	BGT	CC1	
826	002366	002401	BLT	.+4	
827	002370	104400	HLT		;ONE OF THE ABOVE BRANCHES FAILED
828					
829					;CONTINUE
830	002372	000262	SEV		;CC'S=1010
831	002374	102003	BVC	CC2	
832	002376	002402	BLT	CC2	
833	002400	003401	BLE	CC2	
834	002402	002001	BGE	.+4	
835	002404	104400	HLT		;ERROR! ONE OF THE ABOVE BRANCHES FAILED
836					
837					;CONTINUE
838	002406	000261	SEC		;CC'S=1011
839	002410	103002	BCC	CC3	
840	002412	101001	BHI	CC3	
841	002414	003001	BGT	.+4	
842	002416	104400	HLT		;ERROR! ONE OF THE ABOVE BRANCHES FAILED
843					
844					;CONTINUE
845	002420	000264	SEZ		;CC'S=1111
846	002422	001003	BNE	CC4	
847	002424	003002	BGT	CC4	
848	002426	101001	BHI	CC4	
849	002430	003401	BLE	.+4	
850	002432	104400	HLT		;ERROR! ONE OF THE ABOVE BRANCHES FAILED
851	002434	104000	SCOPE		
852					
853					;TEST UNARY CONDITION CODES
854					;CLR
855	002436	000277	RO		
856	002440	000244	SCC		
857	002442	005000	CLZ		
858	002444	103404	CLR	RO	;RO=0,CC'S=0100
859	002446	102403	BCS	CLRO	
860	002450	001002	BVS	CLRO	
861	002452	106401	BNE	CLRO	
862	002454	003401	BMI	CLRO	
863	002456	104400	BLE	.+4	
864			HLT		;ERROR! INCORRECT CC'S AFTER CLR
865	002460	000277	SCC		
866	002462	000244	CLZ		
867	002464	005700	TST	RO	;RO=0,CC'S=0100
868	002466	103404	BCS	TSTO	
869	002470	102403	BVS	TSTO	

870	002472	001002	BNE	TSTO	
871	002474	100401	BMI	TSTO	
872	002476	101401	BLOS	.+4	
873	002500	104400	HLT		;ERROR! INCORRECT CC'S AFTER TST
874					
875	002502	000257	CCC		
876	002504	000266	+SEZ!SEV		
877	002506	005100	COM	RO	;RO=-1,CC'S=1001
878	002510	103004	BCC	COMO	
879	002512	102403	BVS	COMO	
880	002514	001402	BEQ	COMO	
881	002516	100001	BPL	COMO	
882	002520	002401	BLT	.+4	
883	002522	104400	HLT		;ERROR! INCORRECT CC'S AFTER COM
884					
885	002524	000261	SEC		
886	002526	005500	ADC	RO	;RO=000000,CC'S=0101
887	002530	103003	BCC	ADCO	
888	002532	102402	BVS	ADCO	
889	002534	001001	BNE	ADCO	
890	002536	002001	BGE	.+4	
891	002540	104400	HLT		;ERROR! INCORRECT CC'S AFTER ADC
892					
893	002542	000261	SEC		
894	002544	006000	ROR	RO	;RO=100000,CC'S=1010
895	002546	103404	BCS	RORO	
896	002550	102003	BVC	RORO	
897	002552	001402	BEQ	RORO	
898	002554	100001	BPL	RORO	
899	002556	003001	BGT	.+4	
900	002560	104400	HLT		;ERROR! INCORRECT CC'S AFTER ROR
901	002562	000277	SCC		
902	002564	000242	CLV		
903	002566	005300	DEC	RO	;RO=077777,CC'S=0011
904	002570	103004	BCC	DECO	
905	002572	102003	BVC	DECO	
906	002574	001402	BEQ	DECO	
907	002576	100401	BMI	DECO	
908	002600	003401	BLE	.+4	
909	002602	104400	HLT		;ERROR! INCORRECT CC'S AFTER DEC
910					
911	002604	000257	CCC		
912	002606	005200	INC	RO	;RO=100000,CC'S=1010
913	002610	103404	BCS	INCO	
914	002612	102003	BVC	INCO	
915	002614	001402	BEQ	INCO	
916	002616	100001	BPL	INCO	
917	002620	003001	BGT	.+4	
918	002622	104400	HLT		;ERROR! INCORRECT CC'S AFTER INC
919					
920	002624	000277	SCC		
921	002626	000242	CLV		
922	002630	005400	NEG	RO	;RO=100000,CC'S=1011
923	002632	103003	BCC	NEGO	
924	002634	102002	BVC	NEGO	
925	002636	001401	BEQ	NEGO	

926	002640	002001			
927	002642	104400	NEGO:	BGE HLT	.+4 ;ERROR! INCORRECT CC'S AFTER NEG
928					
929	002644	000261		SEC	
930	002646	006300		ASL	RO ;RO=000000,CC'S=0111
931	002650	103004		BCC	ASLO
932	002652	102003		BVC	ASLO
933	002654	001002		BNE	ASLO
934	002656	100401		BMI	ASLO
935	002660	101401		BLOS	.+4
936	002662	104400	ASLO:	HLT	;ERROR! INCORRECT CC'S AFTER ASL
937					
938	002664	006100		ROL	RO ;RO=000001,CC'S=0000
939	002666	103402		BCS	ROLO
940	002670	003401		BLE	ROLO
941	002672	002001		BGE	.+4
942	002674	104400	ROLO:	HLT	;ERROR! INCORRECT CC'S AFTER ROL
943					
944	002676	006200		ASR	RO ;RO=000000,CC'S=0111
945	002700	103003		BCC	ASRO
946	002702	102002		BVC	ASRO
947	002704	001001		BNE	ASRO
948	002706	002401		BLT	.+4
949	002710	104400	ASRO:	HLT	;ERROR! INCORRECT CC'S AFTER ASR
950					
951	002712	000277		SCC	
952	002714	005600		SBC	RO ;RO=-1,CC'S=1001
953	002716	103002		BCC	SBCO
954	002720	102401		BVS	SBCO
955	002722	003401		BLE	.+4
956	002724	104400	SBCO:	HLT	;ERROR! INCORRECT CC'S AFTER SBC
957					
958	002726	005400		NEG	RO ;RO=000001,CC'S=00001
959	002730	000300		SWAB	RO ;RO=000400,CC'S=0100
960	002732	103403		BCS	SWABO
961	002734	102402		BVS	SWABO
962	002736	001001		BNE	SWABO
963	002740	002001		BGE	.+4
964	002742	104400	SWABO:	HLT	;ERROR! INCORRECT CC'S AFTER SWAB
965	002744	104600		SCOPE	
966					
967			;CHECK REGISTER SELECTION		
968	002746	005000		CLR	RO
969	002750	000277		SCC	
970	002752	006100		ROL	RO ;RO=1
971	002754	010002		MOV	RO,R2
972	002756	006302		ASL	R2 ;R2=2
973	002760	010203		MOV	R2,R3
974	002762	006303		ASL	R3 ;R3=4
975	002764	010304		MOV	R3,R4
976	002766	006304		ASL	R4 ;R4=10
977	002770	010405		MOV	R4,R5
978	002772	006305		ASL	R5 ;R5=20
979	002774	010546		MOV	R5, -(SP) ;SET BITS SET IN REGISTERS
980	002776	050416		BIS	R4, (SP) ;INTO STACK ADDRESS
981	003000	050316		BIS	R3, (SP)


```

982 003002 050216          BIS      R2,(SP)
983 003004 050016          BIS      R0,(SP)
984 003006 022726 000037    CMP      #37,(SP)+
985 003012 001401          BEQ      .+4
986 003014 104400          HLT
987
988
989
990 003016 000257          ;CHECK THAT ALL BITS CAN BE SET & CLEARED IN ALL REGISTERS
991 003020 112700 000377    CCC
992 003024 006100          1$: MOVB   #377,R0      ;SET ALL BITS (MOVB EXTENDS SIGN)
993 003026 103776          ROL     R0           ;ROTATE A 0 THROUGH ALL BIT
994 003030 005200          BCS    1$           ;POSITIONS
995 003032 001401          INC     R0           ;FINAL RESULT IS -1
996 003034 104400          BEQ     .+4
997
998 003036 012700 000020    MOV     #16.,R0     ;SET SHIFT COUNT
999 003042 005002          CLR     R2
1000 003044 000261          2$: SEC
1001 003046 006002          ROR     R2           ;ROTATE 1 THROUGH ALL BIT POSITS
1002 003050 005300          DEC     R0           ;DECREMENT SHIFT COUNT
1003 003052 001374          BNE    2$
1004 003054 005102          COM     R2           ;R2 SHOULD CONTAIN -1
1005 003056 001401          BEQ     .+4
1006 003060 104400          HLT           ;ERROR! CHECK R2 SHOULD = 0
1007
1008 003062 012703 100000    MOV     #100000,R3
1009 003064 006203          3$: ASR     R3           ;EXTEND 1 BIT THROUGH ALL POSITIONS
1010 003070 103376          BCC    3$
1011 003072 005203          INC     R3
1012 003074 001401          BEQ     .+4
1013 003076 104400          HLT           ;ERROR!
1014
1015 003100 112704 177401    MOVB   #177401,R4
1016 003104 060404          4$: ADD     R4,R4     ;R4=1
1017 003106 103376          BCC    4$           ;HAS THE AFFECT OF SHIFTING A BIT
1018 003110 005704          TST    R4           ;THROUGH ALL POSITIONS
1019 003112 001401          BEQ     .+4         ;RESULT SHOULD BE 0
1020 003114 104400          HLT
1021
1022 003116 012705 000001    MOV     #1,R5
1023 003122 006305          5$: ASL     R5
1024 003124 102376          BVC    5$
1025 003126 006305          ASL     R5
1026 003130 102302          BCC    6$
1027 003132 005705          TST    R5
1028 003134 001401          BEQ     .+4
1029 003136 104400          6$: HLT
1030
1031          ;CHECK REGISTER VOLITILITY
1032 003140 005002          CLR     R2
1033 003142 005102          COM     R2           ;R2=-1
1034 003144 010203          MOV     R2,R3
1035 003146 000257          CCC
1036 003150 006002          ROR     R2           ;R2=LOOP COUNT
1037 003152 006202          ASR     R2

```

K02

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 23

1038	003154	010304			7\$:	MOV	R3,R4	
1039	003156	005302				DEC	R2	;DECREMENT LOOP COUNT
1040	003160	001375				BNE	7\$	
1041	003162	005203				INC	R3	;CHECK R3
1042	003164	001002				BNE	8\$	
1043	003166	005204				INC	R4	;CHECK R4
1044	003170	001401				BEG	.+4	
1045	003172	104400			8\$:	HLT		
1046								
1047								;CHECK TRANSFER OF REGISTER DATA BETWEEN THE GS AND GD REGISTERS (11/45)
1048	003174	032737	000020	177776	6\$TST:	BIT	#20,2#PSW	;CHECK IF 'T' BIT IS SET
1049	003202	001052				BNE	7\$;SKIP TEST IF 'T' BIT SET
1050	003204	010146				MOV	R1,-(SP)	;SAVE SCOPE PTR
1051	003206	010627				MOV	SP,(PC)+	;SAVE STACK PTR
1052	003210	000000			1\$:	.WORD	0	;CONTAINS SAVED STACK PTR
Z 1053	003212	010727				MOV	PC,(PC)+	;LOAD DATA. THE CURRENT PC IS USED AS
1054	003214	000000			2\$:	.WORD	0	;DATA. IF THIS TEST FAILS 2\$ CON-
1055								;TAINS THE DATA BEING USED.
1056	003216	005267	177772			INC	2\$;MAKE ODD TO CHECK BIT 0
1057	003222	016700	177766		3\$:	MOV	2\$,R0	;LOAD GD REGISTER 0
1058	003226	010001				MOV	R0,R1	;TRANSFER GS REG 0 TO GD REG 1
1059	003230	010102				MOV	R1,R2	;AND GS REG 1 TO GD REG 2
1060	003232	010203				MOV	R2,R3	;ETC...
1061	003234	010304				MOV	R3,R4	
1062	003236	010405				MOV	R4,R5	
1063	003240	152737	000340	177776		BISB	#340,2#PSW	;SET PRIORITY LEVEL 7
1064	003246	010506				MOV	R5,SP	;TRANSFER GS REG 5 TO GD STK PTR
1065	003250	010627				MOV	SP,(PC)+	;TRANSFER GS STK PTR TO MEMORY
1066	003252	000000			4\$:	.WORD	0	;CONTAINS GS STACK PTR
1067	003254	016706	177730			MOV	1\$,SP	;RESTORE STK PTR NEEDED FOR HLT/SCOPE
1068	003260	142737	000340	177776		BICB	#340,2#PSW	;SET PRIORITY LEVEL 0
1069	003266	026700	177760			CMP	4\$,R0	;COMPARE GS/GD STKPTR WITH GS REG 0
1070	003272	001004				BNE	5\$;BRANCH IF THEY WERE NOT =
1071	003274	006367	177714			ASL	2\$;SHIFT TEST DATA UNTIL = 000000
1072	003300	001350				BNE	3\$	
1073	003302	000411				BR	6\$	
1074	003304	010746			5\$:	MOV	R0,-(SP)	;GET GS REG 0
1075	003306	010146				MOV	R1,-(SP)	;ETC...
1076	003310	010246				MOV	R2,-(SP)	
1077	003312	010346				MOV	R3,-(SP)	
1078	003314	010446				MOV	R4,-(SP)	
1079	003316	010546				MOV	R5,-(SP)	
1080	003320	104400				HLT		;ERROR! DATA IN GS STK PTR NOT = GS REG 0
1081								;GS REG 0-GS REG 5 ARE ON THE STACK
1082	003322	016706	177662			MOV	1\$,SP	;RESTORE STACK PTR
1083	003326	012601			6\$:	MOV	(SP)+,R1	;RESTORE SCOPE PTR
1084	003330	104000			7\$:	SCOPE		
1085								
1086								;TEST UNARY WORD INSTRUCTIONS USING ADDRESS MODE 1
1087	003332	000401				BR	.+4	
1088	003334	000000				.WORD	0	;RESERVE ADDRESS FOR TESTS
1089	003336	010702				MOV	PC,R2	
1090	003340	162702	000004			SUB	#4,R2	;R2 POINTS TO RESERVED WORD
1091	003344	005012				CLR	(R2)	;PRESET (R2)
1092								
1093	003346	000261				SLC		

1094	003350	006012	ROR	(R2)	;(R2)=100000,CC=1010
1095	003352	101402	BLOS	ROR1	
1096	003354	100001	BPL	ROR1	
1097	003356	002001	BGE	.+4	
1098	003360	104400	ROR1: HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1099					
1100	003362	000257	CCC		
1101	003364	000261	SEC		
1102	003366	005312	DEC	(R2)	;(R2)=077777,CC=0011
1103	003370	103001	BCC	DEC1	
1104	003372	003401	BLE	.+4	
1105	003374	104400	DEC1: HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1106					
1107	003376	000257	CCC		
1108	003400	000261	SEC		
1109	003402	005512	ADC	(R2)	;(R2)=100000,CC=1010
1110	003404	103403	BCS	ADC1	
1111	003406	102002	BVC	ADC1	
1112	003410	100001	BPL	ADC1	
1113	003412	001001	BNE	.+4	
1114	003414	104400	FDC1: HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1115					
1116	003416	006112	ROL	(R2)	;(R2)=000000,CC=0111
1117	003420	103003	BCC	ROL1	
1118	003422	102002	BVC	ROL1	
1119	003424	001001	BNE	ROL1	
1120	003426	100001	BPL	.+4	
1121	003430	104400	ROL1: HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1122					
1123	003432	006112	ROL	(R2)	;(R2)=000001,CC=0000
1124	003434	101402	BLOS	ROL1A	;BRANCH IF C OR Z IS SET
1125	003436	102401	BVS	ROL1A	
1126	003440	100001	BPL	.+4	
1127	003442	104400	ROL1A: HLT		
1128					
1129	003444	006212	ASR	(R2)	;(R2)=000000,CC=0111
1130	003446	103003	BCC	ASR1	
1131	003450	102002	BVC	ASR1	
1132	003452	001001	BNE	ASR1	
1133	003454	100001	BPL	.+4	
1134	003456	104400	ASR1: HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1135					
1136	003460	006012	ROR	(R2)	;(R2)=100000,CC=1010
1137	003462	103403	BCS	ROR1A	
1138	003464	102002	BVC	ROR1A	
1139	003466	001401	BEQ	ROR1A	
1140	003470	100401	BMI	.+4	
1141	003472	104400	ROR1A: HLT		
1142					
1143	003474	000261	SEC		
1144	003476	005212	INC	(R2)	;(R2)=100001,CC=1001
1145	003500	103003	BCC	INC1	
1146	003502	102402	BVS	INC1	
1147	003504	001401	BEQ	INC1	
1148	003506	100401	BMI	.+4	
1149	003510	104400	INC1: HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE

1150					
1151	003512	005612	SBC	(R2)	;(R2)=100000,CC=1000
1152	003514	103403	BCS	SBC1	
1153	003516	102402	BVS	SBC1	
1154	003520	001401	BEQ	SBC1	
1155	003522	100401	BMI	.+4	
1156	003524	104400	SBC1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1157					
1158	003526	000261	SEC		
1159	003530	005612	SBC	(R2)	;(R2)=077777,CC=0010
1160	003532	103403	BCS	SBC1A	
1161	003534	102002	BVC	SBC1A	
1162	003536	001401	BEQ	SBC1A	
1163	003540	100001	BPL	.+4	
1164	003542	104400	SBC1A:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1165					
1166	003544	000261	SEC		
1167	003546	005512	ADC	(R2)	;(R2)=100000,CC=1010
1168	003550	100401	BMI	.+4	
1169	003552	104400	HLT		
1170					
1171	003554	000261	SEC		
1172	003556	006312	ASL	(R2)	;(R2)=000000,CC=0111
1173	003560	103003	BCC	ASL1	
1174	003562	102002	BVC	ASL1	
1175	003564	001001	BNE	ASL1	
1176	003566	100001	BPL	.+4	
1177	003570	104400	ASL1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1178					
1179	003572	005112	COM	(R2)	;(R2)=177777,CC=1001
1180	003574	103002	BCC	COM1	
1181	003576	102401	BVS	COM1	
1182	003600	100401	BMI	.+4	
1183	003602	104400	COM1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1184					
1185	003604	000250	CLN		
1186	003606	005712	TST	(R2)	;(R2)=177777,CC=1000
1187	003610	103403	BCS	TST1	
1188	003612	102402	BVS	TST1	
1189	003614	100001	BPL	TST1	
1190	003616	001001	BNE	.+4	
1191	003620	104400	TST1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1192					
1193	003622	000262	SEV		
1194	003624	005412	NEG	(R2)	;(R2)=000001,CC=0000
1195	003626	103002	BCC	NEG1	
1196	003630	102401	BVS	NEG1	
1197	003632	001001	BNE	.+4	
1198	003634	104400	NEG1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1199					
1200	003636	005312	DEC	(R2)	;(R2)=000000,CC=0101
1201	003640	103001	BCC	DEC1A	
1202	003642	001401	BEQ	.+4	
1203	003644	104400	DEC1A:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1204	003646	104000	SCOPE		
1205					

```

1206                                     ;CHECK UNARY BYTE INSTRUCTIONS USING ADDRESS MODE 1
1207 003650 000401                       BR      .+4           ;RESERVE A WORD
1208 003652 000000                       .WORD  0           ;ADDRESS RESERVED FOR TESTS
1209 003654 010703                       MOV     PC,R3
1210 003656 162703 000004               SUB     #4,R3       ;R3 POINTS TO EVEN BYTE OF WORD
1211 003662 010304                       MOV     R3,R4       ;R4 POINTS TO ODD BYTE OF WORD
1212 003664 005204                       INC     R4
1213 003666 005013                       CLR     (R3)        ;PRESET DATA
1214
1215 003670 000261                       1$:     SEC
1216 003672 105513                       ADCB   (R3)         ;ADD CARRY TO EVEN BYTE
1217 003674 100402                       BMI    2$          ;UNTIL EVEN BYTE BECOMES NEGATIVE
1218 003676 105214                       INCB   (R4)        ;INCREMENT ODD BYTE
1219 003700 000773                       BR     1$
1220 003702 102401                       2$:     BVS        .+4           ;(R3)=077600=[0774][200],CC=1010
1221 003704 104400                       HLT
1222 003706 070242                       CLV
1223 003710 105214                       INCB   (R4)        ;(R3)=100200=[1000][200],CC=1010
1224 003712 103402                       BCS   INCB1
1225 003714 102001                       BVC   INCB1
1226 003716 100401                       BMI    .+4
1227 003720 104400                       INCB1:  HLT        ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1228
1229 003722 106114                       ROLB   (R4)        ;(R3)=000200=[0000][200],CC=0111
1230 003724 103002                       BCC   ROLB1
1231 003726 102001                       BVC   ROLB1
1232 003730 001401                       BEQ   .+4
1233 003732 104400                       ROLB1:  HLT        ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1234
1235 003734 105614                       SBCB   (R4)        ;(R3)=177600=[1774][200],CC=1001
1236 003736 103002                       BCC   SBCB1
1237 003740 102401                       BVS   SBCB1
1238 003742 100401                       BMI    .+4
1239 003744 104400                       SBCB1:  HLT        ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1240
1241 003746 106313                       ASLB   (R3)        ;(R3)=177400,CC=0111
1242 003750 103002                       BCC   ASLB1
1243 003752 102001                       BVC   ASLB1
1244 003754 001401                       BEQ   .+4
1245 003756 104400                       ASLB1:  HLT        ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1246
1247 003760 105413                       NEGB   (R3)        ;(R3)=177400,CC=0100
1248 003762 103402                       BCS   NEGB1
1249 003764 102401                       BVS   NEGB1
1250 003766 001401                       BEQ   .+4
1251 003770 104400                       NEGB1:  HLT        ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1252
1253 003772 000277                       SCC
1254 003774 105313                       DECB   (R3)        ;(R3)=177777,CC=1001
1255 003776 103002                       BCC   DECB1
1256 004000 102401                       BVS   DECB1
1257 004002 001001                       BNE    .+4
1258 004004 104400                       DECB1:  HLT        ;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1259
1260 004006 000241                       CLC
1261 004010 106013                       RORB   (R3)        ;(R3)=177577,CC=0011

```


1262	004012	103002	BVC	RORB1	
1263	004014	102001	BVC	RORB1	
1264	004016	100001	BPL	.+4	
1265	004020	104400	RORB1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1266					
1267	004022	000241	CLC		
1268	004024	105114	COMB	(R4)	; (R3)=000177,CC=0101
1269	004026	103002	BCC	COMB1	
1270	004030	102401	BVS	COMB1	
1271	004032	001401	BEQ	.+4	
1272	004034	104400	COMB1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1273					
1274	004036	106213	1S:	ASRB (R3)	;SHIFT EVEN BYTE UNTIL V CLEARS
1275	004040	102002	BVC	2S	
1276	004042	105514	ADCB	(R4)	;AND ADD CARRY TO ODD BYTE
1277	004044	000774	BR	1S	
1278	004046	103401	2S:	BCS ASRB1	
1279	004050	001401	BEQ	.+4	
1280	004052	104400	ASRB1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1281					
1282	004054	106214	ASRB	(R4)	
1283	004056	106214	ASRB	(R4)	; (R3)=000400,CC=0011
1284	004060	103002	BCC	ASRB1A	
1285	004062	102001	BVC	ASRB1A	
1286	004064	001001	BNE	.+4	
1287	004066	104400	ASRB1A:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1288					
1289	004070	105314	DECB	(R4)	; (R3)=000000,CC=0100
1290	004072	001401	BEQ	.+4	
1291	004074	104400	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1292					
1293	004076	000261	SEC		
1294	004100	106014	RORB	(R4)	; (R3)=100000,CC=1010
1295	004102	103402	BCS	RORB1A	
1296	004104	102001	BVC	RORB1A	
1297	004106	100401	BMI	.+4	
1298	004110	104400	RORB1A:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1299					
1300	004112	000242	CLV		
1301	004114	105314	DECB	(R4)	; (R3)=077400,CC=0100
1302	004116	102401	BVS	.+4	
1303	004120	104400	HLT		
1304					
1305	004122	000261	SEC		
1306	004124	105313	DECB	(R3)	; (R3)=077777,CC=1001
1307	004126	103002	BCC	DEC81A	
1308	004130	102401	BVS	DEC81A	
1309	004132	100401	BMI	.+4	
1310	004134	104400	DEC81A:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1311					
1312	004136	000277	SCC		
1313	004140	000313	SWAB	(R3)	; (R3)=177577=[1774][177],CC=0000
1314	004142	103402	BCS	SWAB1	
1315	004144	102401	BVS	SWAB1	
1316	004146	100001	BPL	.+4	
1317	004150	104400	SWAB1:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE

1318						
1319	004152	105714	TSTB	(R4)		; (R3)=177577=[1774][177],CC=1000
1320	004154	103402	BCS	TSTB1		
1321	004156	102401	BVS	TSTB1		
1322	004160	100401	BMI	.+4		
1323	004162	104400	TSTB1:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1324						
1325	004164	105014	CLRB	(R4)		; (R3)=000177=[0000][177],CC=0100
1326	004166	001401	BEQ	.+4		
1327	004170	104400	HLT			
1328	004172	106313	ASLB	(R3)		; (R3)=000376 ,CC=1010
1329	004174	103402	BCS	ASLB1A		
1330	004176	102001	BVC	ASLB1A		
1331	004200	100401	BMI	.+4		
1332	004202	104400	ASLB1A:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1333						
1334	004204	105113	COMB	(R3)		; (R3)=000001,CC=0001
1335	004206	103002	BCC	COMB1A		
1336	004210	102401	BVS	COMB1A		
1337	004212	100001	BPL	.+4		
1338	004214	104400	COMB1A:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1339						
1340	004216	000313	SWAB	(R3)		; (R3)=000400, CC=0100
1341	004220	001401	BEQ	.+4		
1342	004222	104400	HLT			
1343						
1344	004224	105213	INCB	(R3)		
1345	004226	000751	SEC			
1346	004230	105613	SBCB	(R3)		; (R3)=000400,CC=0100
1347	004232	001401	BEQ	.+4		
1348	004234	104400	HLT			
1349	004236	022713	CMP	#400,(R3)		;CHECK REMAINING RESULT
1350	004242	001401	BEQ	.+4		
1351	004244	104400	HLT			
1352	004246	104000	SCOPE			
1353						
1354						;CHECK UNARY WORD OPS USING ADDRESS MODES 2 AND 4 (AUTO INC/DEC)
1355	004250	000401	BR	.+4		
1356	004252	000000	.WORD	0		;ADDRESS RESERVED FOR TESTS
1357	004254	010704	MOV	PC,R4		
1358	004256	162704	SUB	#4,R4		;R4 AND R5 POINT TO
1359	004262	010405	MOV	R4,R5		;RESERVED WORD
1360	004264	005015	CLR	(R5)		;PRESET DATA=0
1361						
1362	004266	000277	SCC			
1363	004270	000244	CLZ			
1364	004272	005725	TST	(R5)+		; (R5)=000000,CC=0100
1365	004274	103402	BCS	TST2		
1366	004276	102401	BVS	TST2		
1367	004300	001401	BEQ	.+4		
1368	004302	104400	TST2:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1369						
1370	004304	005145	COM	-(R5)		; (R5)=177777,CC=1001
1371	004306	103001	BCC	COM4		
1372	004310	100401	BMI	.+4		
1373	004312	104400	COM4:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE

1374					
1375	004314	000241	CLC		
1376	004316	006024	ROR	(R4)+	;(R4)=077777,CC=0011
1377	004320	103002	BCC	ROR2	
1378	004322	102001	BVC	ROR2	
1379	004324	100001	BPL	.+4	
1380	004326	104400	ROR2:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1381					
1382	004330	000257	CCC		
1383	004332	005244	INC	-(R4)	;(R4)=100000,CC=1010
1384	004334	102002	BVC	INC4	
1385	004336	001401	BEQ	INC4	
1386	004340	100401	BMI	.+4	
1387	004342	104400	INC4:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1388					
1389	004344	000261	SEC		
1390	004346	000324	SWAB	(R4)+	;(R4)=000200,CC=1000
1391	004350	103401	BCS	SWAB2	
1392	004352	100401	BMI	.+4	
1393	004354	104400	SWAB2:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1394					
1395	004356	005425	NEG	(R5)+	;(R5)=177600,CC=1001
1396	004360	103001	BCC	NEG2	
1397	004362	100401	BMI	.+4	
1398	004364	104400	NEG2:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1399					
1400	004366	005044	CLR	-(R4)	;(R4)=000000,CC=0100
1401	004370	001401	BEQ	.+4	
1402	004372	104400	HLT		
1403					
1404	004374	000261	SEC		
1405	004376	006045	ROR	-(R5)	;(R5)=100000,CC=1010
1406	004400	000261	SEC		
1407	004402	005525	ADC	(R5)+	;(R5)=100001,CC=1000
1408	004404	102401	BVS	ADC2	
1409	004406	100401	BMI	.+4	
1410	004410	104400	ADC2:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1411					
1412	004412	000262	SEV		
1413	004414	006224	ASR	(R4)+	;(R4)=140000,CC=1001
1414	004416	103002	BCC	ASR2	
1415	004420	102401	BVS	ASR2	
1416	004422	100401	BMI	.+4	
1417	004424	104400	ASR2:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1418					
1419	004426	000262	SEV		
1420	004430	006144	ROL	-(R4)	;(R4)=100001,CC=1001
1421	004432	103002	BCC	ROL4	
1422	004434	102401	BVS	ROL4	
1423	004436	100401	BMI	.+4	
1424	004440	104400	ROL4:	HLT	;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1425					
1426	004442	005645	SBC	-(R5)	;(R5)=100000,CC=1000
1427	004444	103001	BCC	.+4	
1428	004446	104400	HLT		;ERROR! 'C' BIT FAILED TO CLEAR
1429					

1430	004450	005325		DEC	(R5)+		; (R5)=077777,CC=0010
1431	004452	103402		BCS	DEC2		
1432	004454	102001		BVC	DEC2		
1433	004456	100001		BPL	.+4		
1434	004460	104400		DEC2:	HLT		; ERROR! INCORRECT CC'S AS SHOWN ABOVE
1435							
1436	004462	006324		ASL	(R4)+		; (R4)=177776,CC=1010
1437	004464	102401		BVS	.+4		
1438	004466	104400		HLT			
1439	004470	006344		ASL	-(R4)		; (R4)=177774,CC=1001
1440	004472	103003		BCC	ASL4		
1441	004474	102402		BVS	ASL4		
1442	004476	001401		BEQ	ASL4		
1443	004500	100401		BMI	.+4		
1444	004502	104400		ASL4:	HLT		; ERROR! INCORRECT CC'S AS SHOWN ABOVE
1445							
1446	004504	022724	177774	CMP	#177774,(R4)+		
1447	004510	001401		BEQ	.+4		
1448	004512	104400		HLT			
1449	004514	020405		CMP	R4,R5		
1450	004516	001401		BEQ	.+4		
1451	004520	104400		HLT			
1452	004522	104000		SCOPE			
1453							
1454							; CHECK UNARY BYTE OPS USING ADDRESS MODES 2 AND 4
1455	004524	000401		BR	.+4		; RESERVE A WORD
1456	004526	000000		.WORD	0		; RESERVED WORD
1457	004530	010705		MOV	PC,R5		
1458	004532	162705	000004	SUB	#4,R5		; R5 POINTS TO EVEN BYTE OF RESERVED WORD
1459	004536	010500		MOV	R5,R0		
1460	004540	010002		MOV	R0,R2		
1461	004542	005202		INC	R2		; R2 POINTS TO ODD BYTE OF RESERVED WORD
1462	004544	005010		CLR	(R0)		; PRESET
1463							
1464	004546	000277		SCC			
1465	004550	000241		CLC			
1466	004552	105125		COMB	(R5)+		; (R0)=000377,CC=1001
1467	004554	103002		BCC	COMB2		
1468	004556	102401		BVS	COMB2		
1469	004560	100401		BMI	.+4		
1470	004562	104400		COMB2:	HLT		; ERROR! INCORRECT CC'S AS SHOWN ABOVE
1471							
1472	004564	105542		ADCB	-(R2)		; (R0)=000000,CC=0101
1473	004566	001401		BEQ	.+4		
1474	004570	104400		HLT			; ERROR! INCORRECT RESULT AS SHOWN ABOVE
1475	004572	105525		ADCB	(R5)+		; (R0)=000400,CC=0000
1476	004574	103401		BCS	ADCB2		
1477	004576	001001		BNE	.+4		
1478	004600	104400		ADCB2:	HLT		; ERROR! INCORRECT CC'S AS SHOWN ABOVE
1479							
1480	004602	000263		+SEC!SEV			
1481	004604	106045		RORB	-(R5)		; (R0)=100000,CC=1001
1482	004606	103003		BCC	RORB4		
1483	004610	102402		BVS	RORB4		
1484	004612	001401		BEQ	RORB4		
1485	004614	100401		BMI	.+4		

1486	004616	104400	RORB4:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1487						
1488	004620	000277		SCC		
1489	004622	106122		ROLB	(R2)+	;(RO)=100001,CC=0000
1490	004624	103403		BCS	ROLB2	
1491	004626	102402		BVS	ROLB2	
1492	004630	001401		BEQ	ROLB2	
1493	004632	100001		BPL	.+4	
1494	004634	104400	ROLB2:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1495						
1496	004636	000257		CCC		
1497	004640	106225		ASRB	(R5)+	;(RO)=140001, CC=1010
1498	004642	103402		BCS	ASRB2	
1499	004644	102001		BVC	ASRB2	
1500	004646	100401		BMI	.+4	
1501	004650	104400	ASRB2:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1502						
1503	004652	105242		INCB	-(R2)	;(RO)=140002,CC=0000
1504	004654	000277		SCC		
1505	004656	106222		ASRB	(R2)+	;(RO)=140001,CC=0000
1506	004660	103402		BCS	ASRB2A	
1507	004662	102401		BVS	ASRB2A	
1508	004664	100001		BPL	.+4	
1509	004666	104400	ASRB2A:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1510						
1511	004670	000266		+SEZ!SEV		;SET Z,V
1512	004672	106345		ASLB	-(R5)	;(RO)=100001,CC=1001
1513	004674	103003		BCC	ASLB4	
1514	004676	102402		BVS	ASLB4	
1515	004700	001401		BEQ	ASLB4	
1516	004702	100401		BMI	.+4	
1517	004704	104400	ASLB4:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1518						
1519	004706	105322		DECB	(R2)+	;(RO)=077401=[0774][001] ,CC=0010
1520	004710	103002		BCC	DECB2	
1521	004712	102001		BVC	DECB2	
1522	004714	100001		BPL	.+4	
1523	004716	104400	DECB2:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1524						
1525	004720	105645		SBCB	-(R5)	;(RO)=077400, CC=0100
1526	004722	103402		BCS	SBCB4	
1527	004724	102401		BVS	SBCB4	
1528	004726	001401		BEQ	.+4	
1529	004730	104400	SBCB4:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1530						
1531	004732	105442		NEGB	-(R2)	;(RO)=10400,CC=1001
1532	004734	103002		BCC	NEGB4	
1533	004736	102401		BVS	NEGB4	
1534	004740	100401		BMI	.+4	
1535	004742	104400	NEGB4:	HLT		;ERROR! INCORRECT CC'S AS SHOWN ABOVE
1536						
1537	004744	105725		TSTB	(R5)+	;(RO)=100400,CC=0100
1538	004746	103401		BCS	TSTB2	
1539	004750	001401		BEQ	.+4	
1540	004752	104400	TSTB2:	HLT		
1541						

1542	004754	105722		TSTB	(R2)+	; (R0)=100400,CC=1000
1543	004756	001401		BEQ	TSTB2A	
1544	004760	100401		BMI	.+4	
1545	004762	104400		TSTB2A:	HLT	
1546						
1547	004764	000261		SEC		
1548	004766	000342		SWAB	-(R2)	; (R0)=000201,CC=1000
1549	004770	103401		BCS	SWAB4	
1550	004772	100401		BMI	.+4	
1551	004774	104400		SWAB4:	HLT	
1552						
1553	004776	000277		SCC		
1554	005000	105225		INCB	(R5)+	; (R0)=000601=[0004][201],CC=0000
1555	005002	103003		BCC	INCB2	
1556	005004	102402		BVS	INCB2	
1557	005006	001401		BEQ	INCB2	
1558	005010	100001		BPL	.+4	
1559	005012	104400		INCB2:	HLT	
1560						
1561	005014	022227	000601	CMP	(R2)+, #000601	; CHECK END RESULT
1562	005020	001401		BEQ	.+4	
1563	005022	104400		HLT		
1564	005024	020205		CMP	R2,R5	; CHECK REGISTERS
1565	005026	001401		BEQ	.+4	
1566	005030	104400		HLT		
1567	005032	104000		SCOPE		
1568						
1569						
1570	005034	000402				; CHECK UNARY WORD OPS USING ADDRESS MODES 3 AND 5
1571	005036	000000		BR	.+6	; RESERVE 2 WORDS
1572	005040	000000		.WORD	0	; 1 FOR THE ADDRESS
1573	005042	010703		.WORD	0	; AND 1 FOR DATA
1574	005044	162703	000004	MOV	PC,R3	
1575	005050	005013		SUB	#4,R3	
1576	005052	010300		CLR	(R3)	; PRESET DATA
1577	005054	005743		MOV	R3,R0	; R0 POINTS TO DATA WORD
1578	005056	010013		TST	-(R3)	
1579	005060	010304		MOV	R0,(R3)	
1580				MOV	R3,R4	
1581	005062	000257		CCC		
1582	005064	005733		TST	2(R3)+	; (R0)=000000,CC=0100
1583	005066	001401		BEQ	.+4	
1584	005070	104400		HLT		
1585						
1586	005072	000261		SEC		
1587	005074	006053		ROR	2-(R3)	; (R0)=100000,CC=1010
1588	005076	103402		BCS	RORS	
1589	005100	102001		BVC	RORS	
1590	005102	100401		BMI	.+4	
1591	005104	104400		RORS:	HLT	
1592						
1593	005106	000257		CCC		
1594	005110	006234		ASR	2(R4)+	; (R0)=140000,CC=1010
1595	005112	102001		BVC	ASR3	
1596	005114	100401		BMI	.+4	
1597	005116	104400		ASR3:	HLT	

1598					
1599	005120	000250	CLN		
1600	005122	006333	ASL	2(R3)+	;(R0)=100000,CC=1001
1601	005124	103002	BCC	ASL3	
1602	005126	102401	BVS	ASL3	
1603	005130	100401	BMI	.+4	
1604	005132	104400	HLT		
1605			ASL3:		
1606	005134	003277	SCC		
1607	005136	005354	DEC	2-(R4)	;(R0)=077777, CC=0010
1608	005140	103003	BCC	DEC5	
1609	005142	102002	BVC	DEC5	
1610	005144	001401	BEQ	DEC5	
1611	005146	100001	BPL	.+4	
1612	005150	104400	HLT		
1613			DEC5:		
1614	005152	005453	NEG	2-(R3)	;(R0)=100001, CC=1001
1615	005154	103002	BCC	NEG5	
1616	005156	102401	BVS	NEG5	
1617	005160	100401	BMI	.+4	
1618	005162	104400	HLT		
1619			NEG5:		
1620	005164	000262	SEV		
1621	005166	005134	COM	2(R4)+	;(R0)=077776, CC=0001
1622	005170	103001	BCC	COM3	
1623	005172	102001	BVC	.+4	
1624	005174	104400	HLT		
1625			COM3:		
1626	005176	005233	INC	2(R3)+	;(R0)=077777, CC=0001
1627	005200	103001	BCC	INC3	
1628	005202	100001	BPL	.+4	
1629	005204	104400	HLT		
1630			INC3:		
1631	005206	005554	ADC	2-(R4)	;(R0)=100000, CC=1010
1632	005210	103402	BVS	ADC5	
1633	005212	102001	BVC	ADC5	
1634	005214	100401	BMI	.+4	
1635	005216	104400	HLT		
1636			ADC5:		
1637	005220	000257	CCC		
1638	005222	006134	ROL	2(R4)+	;(R0)=000000,CC=0111
1639	005224	103002	BCC	ROL3	
1640	005226	102001	BVC	ROL3	
1641	005230	001401	BEQ	.+4	
1642	005232	104400	HLT		
1643			ROL3:		
1644	005234	005253	INC	2-(R3)	;(R0)=000001, CC=0001
1645	005236	005654	SBC	2-(R4)	;(R0)=000000, CC=0100
1646	005240	103401	BVS	SBC5	
1647	005242	001401	BEQ	.+4	
1648	005244	104400	HLT		
1649	005246	104000	SCOPE		
1650			SBC5:		
1651			:CHECK UNARY BYTE OPS USING ADDRESS MODES 3 AND 5		
1652	005250	000403	BR	.+10	;RESERVE 3 WORDS
1653	005252	000000	.WORD	0	;1 FOR EVEN BYTE ADDRESS

1654	005254	000000	.WORD	0	; 1 FOR ODD BYTE ADDRESS
1655	005256	000000	.WORD	0	; AND 1 FOR DATA
1656	005260	010702	MOV	PC,R2	
1657	005262	005742	TST	-(R2)	; BACK R2 UP TO
1658	005264	005742	TST	-(R2)	; DATA WORD
1659	005266	010200	MOV	R2,R0	; R0 POINTS TO THE DATA WORD
1660	005270	005010	CLR	(R0)	; PRESET DATA
1661	005272	005742	TST	-(R2)	; BACK R2 UP TO
1662	005274	005742	TST	-(R2)	; EVEN BYTE ADDRESS WORD
1663	005276	010022	MOV	R0,(R2)+	; LOAD ADDRESS
1664	005300	005200	INC	R0	; ODD BYTE ADDRESS
1665	005302	010022	MOV	R0,(R2)+	; LOAD ODD BYTE ADDRESS
1666	005304	010200	MOV	R2,R0	; RESET R0
1667	005306	010205	MOV	R2,R5	
1668					
1669	005310	105152	COMB	2-(R2)	; (R0)=177400, CC=1001
1670	005312	103001	BCC	COMB5	
1671	005314	100401	BMI	.+4	
1672	005316	104400	COMB5:	HLT	
1673					
1674	005320	105752	TSTB	2-(R2)	; (R0)=177400, CC=0100
1675	005322	001401	BEQ	.+4	
1676	005324	104400	HLT		
1677					
1678	005326	000262	SEV		
1679	005330	106255	ASRB	2-(R5)	; (R0)=177400, CC=1001
1680	005332	103002	BCC	ASRB5	
1681	005334	102401	BVS	ASRB5	
1682	005336	100401	BMI	.+4	
1683	005340	104400	ASRB5:	HLT	
1684					
1685	005342	105232	INCB	2(R2)+	; (R0)=177401, CC=000
1686	005344	103001	BCC	INCB3	
1687	005346	100001	BPL	.+4	
1688	005350	104400	INCB3:	HLT	
1689					
1690	005352	000241	CLC		
1691	005354	106055	RORB	2-(R5)	; (R0)=177400, CC=0111
1692	005356	103003	BCC	RORB5	
1693	005360	102002	BVC	RORB5	
1694	005362	001001	BNE	RORB5	
1695	005364	100001	BPL	.+4	
1696	005366	104400	RORB5:	HLT	
1697					
1698	005370	106332	ASLB	2(R2)+	; (R0)=177000, CC=1001
1699	005372	103002	BCC	ASLB3	
1700	005374	102401	BVS	ASLB3	
1701	005376	100401	BMI	.+4	
1702	005400	104400	ASLB3:	HLT	
1703					
1704	005402	105552	ADCB	2-(R2)	; (R0)=177400, CC=1000
1705	005404	103401	BCC	ADCB5	
1706	005406	100401	BMI	.+4	
1707	005410	104400	ADCB5:	HLT	
1708					
1709	005412	000277	SCC		

1710	005414	106135		ROLB	2(R5)+	;(R0)=177401, CC=0000
1711	005416	101402		BLOS	ROLB3	;BRANCH IF C OR Z IS SET
1712	005420	102401		BVS	ROLB3	
1713	005422	100001		BPL	.+4	
1714	005424	104400		ROLB3:	HLT	
1715						
1716	005426	000352		SWAB	2-(R2)	;(R0)=000777, CC=1000
1717	005430	100401		BMI	.+4	
1718	005432	104400		HLT		
1719						
1720	005434	000261		SEC		
1721	005436	105635		SBCB	2(R5)+	;(R0)=000377, CC=0100
1722	005440	103401		BCS	SBCB3	
1723	005442	001401		BEQ	.+4	
1724	005444	104400		SBCB3:	HLT	
1725						
1726	005446	105432		NEGB	2(R2)+	;(R0)=000001
1727	005450	105352		DECB	2-(R2)	;(R0)=000000, CC=0101
1728	005452	103001		BCC	DECB5	
1729	005454	001401		BEQ	.+4	
1730	005456	104400		DECBS:	HLT	
1731	005460	104000		SCOPE		
1732						
1733						;CHECK UNARY WORD OPS USING ADDRESS MODE 6 (PC)
1734	005462	005027		CLR	(PC)+	;PRESET DATA = 0
1735	005464	000000		UWM6:	.WORD 0	;RESERVED FOR DATA
1736	005466	010700		MOV	PC, R0	
1737	005470	024040		CMP	-(R0), -(R0)	;R0 POINTS TO DATA WORD
1738	005472	000277		SCC		
1739	005474	006167	177764	ROL	UWM6	;(R0)=000001, CC=0000
1740	005500	103403		BCS	ROL6	
1741	005502	102402		BVS	ROL6	
1742	005504	001401		BEQ	ROL6	
1743	005506	100001		BPL	.+4	
1744	005510	104400		ROL6:	HLT	
1745						
1746	005512	005167	177746	COM	UWM6	;(R0)=177776, CC=1001
1747	005516	103002		BCC	COM6	
1748	005520	102401		BVS	COM6	
1749	005522	100401		BMI	.+4	
1750	005524	104400		COM6:	HLT	

K03

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 36

1751

..

1752	005526	006267	177732	ASR	UWM6	;(RO)=177777, CC=1010
1753	005532	103402		BCS	ASR6	
1754	005534	102001		BVC	ASR6	
1755	005536	100401		BMI	.+4	
1756	005540	104400		HLT		
1757				ASR6:		
1758	005542	000277		SCC		
1759	005544	005467	177714	NEG	UWM6	;(RO)=000001, CC=0001
1760	005550	103003		BCC	NEG6	
1761	005552	102402		BVS	NEG6	
1762	005554	001401		BEQ	NEG6	
1763	005556	100001		BPL	.+4	
1764	005560	104400		HLT		
1765				NEG6:		
1766	005562	000277		SCC		
1767	005564	006067	177674	ROR	UWM6	;(RO)=100000, CC=1001
1768	005570	103003		BCC	ROR6	
1769	005572	102402		BVS	ROR6	
1770	005574	001401		BEQ	ROR6	
1771	005576	100401		BMI	.+4	
1772	005600	104400		HLT		
1773				ROR6:		
1774	005602	005667	177656	SBC	UWM6	;(RO)=077777, CC=0010
1775	005606	103402		BCS	SBC6	
1776	005610	102001		BVC	SBC6	
1777	005612	100001		BPL	.+4	
1778	005614	104400		HLT		
1779				SBC6:		
1780	005616	000242		CLV		
1781	005620	005267	177640	INC	UWM6	;(RO)=100000, CC=1011
1782	005624	103403		BCS	INC6	
1783	005626	102002		BVC	INC6	
1784	005630	001401		BEQ	INC6	
1785	005632	100401		BMI	.+4	
1786	005634	104400		HLT		
1787				INC6:		
1788	005636	006267	177622	ASR	UWM6	;(RO)=140000, CC=1010
1789	005642	000261		SEC		
1790	005644	006367	177614	ASL	UWM6	;(RO)=100000, CC=1001
1791	005650	103002		BCC	ASL6	
1792	005652	102401		BVS	ASL6	
1793	005654	100401		BMI	.+4	
1794	005656	104400		HLT		
1795				ASL6:		
1796	005660	005367	177600	DEC	UWM6	;(RO)=077777, CC=0011
1797	005664	103002		BCC	DEC6	
1798	005666	102001		BVC	DEC6	
1799	005670	100001		BPL	.+4	
1800	005672	104400		HLT		
1801				DEC6:		
1802	005674	005567	177564	ADC	UWM6	;(RO)=100000, CC=1010
1803	005700	103402		BCS	ADC6	

1804	005702	102001		BVC	ADC6	
1805	005704	100401		BMI	.+4	
1806	005706	104400		ADC6:	HLT	
1807	005710	000242		CLV		
1808	005712	000367	177546	SWAB	UWM6	
1809	005716	100401		BMI	.+4	
1810	005720	104400		HLT		
1811	005722	022710	000200	CMP	#200, (R0)	
1812	005726	001401		BEQ	.+4	
1813	005730	104400		HLT		
1814	005732	104000		SCOPE		
1815						
1816				;CHECK UNARY BYTE OPS (EVEN/ODD) USING ADDRESS MODE 6 (PC)		
1817	005734	012700	006276	MOV	#UBM6, R0	
1818	005740	063700	001004	ADD	2#FACTOR, R0	;R0 POINTS TO ADDRESS OF DATA
1819	005744	005067	000326	CLR	UBM6	;CLEAR DATA
1820	005750	000277		SCC		
1821	005752	000244		CLZ		
1822	005754	105767	000316	TSTB	UBM6	
1823	005760	103403		BCS	TSTB6	
1824	005762	102402		BVS	TSTB6	
1825	005764	001001		BNE	TSTB6	
1826	005766	100001		BPL	.+4	
1827	005770	104400		TSTB6:	HLT	
1828						
1829	005772	000257		CCC		
1830	005774	105767	000277	TSTB	UBM6+1	;TEST ODD BYTE
1831	006000	001401		BEQ	.+4	
1832	006002	104400		HLT		
1833						
1834	006004	105667	000266	SBCB	UBM6	; (R0)=000000, CC=0100
1835	006010	103402		BCS	SBCB6	
1836	006012	102401		BVS	SBCB6	
1837	006014	001401		BEQ	.+4	
1838	006016	104400		SBCB6:	HLT	
1839						
1840	006020	000261		1\$:	SEC	
1841	006022	105267	000250	INCB	UBM6	;LOOP UNTIL (R0)=077600, CC=1011
1842	006026	100403		BMI	2\$	
1843	006030	105567	000243	ADCB	UBM6+1	;INCB INST INCREMENTS EVEN BYTE
1844	006034	000771		BR	1\$;ADCB INCREMENTS ODD BYTE
1845	006036	103001		2\$:	BCC	INCB6
1846	006040	102401		BVS	.+4	
1847	006042	104400		INCB6:	HLT	
1848						
1849	006044	106367	000226	ASLB	UBM6	; (R0)=077400, CC=0111
1850	006050	103003		BCC	ASLB6	
1851	006052	102002		BVC	ASLB6	
1852	006054	001001		BNE	ASLB6	
1853	006056	100001		BPL	.+4	
1854	006060	104400		ASLB6:	HLT	
1855						
1856	006062	000242		CLV		
1857	006064	105567	000207	ADCB	UBM6+1	; (R0)=100000, CC=1010
1858	006070	103402		BCS	ADCB6	
1859	006072	102001		BVC	ADCB6	

1860	006074	100401		BMI	.+4	
1861	006076	104400		ADCB6:	HLT	
1862						
1863	006100	000261		SEC		
1864	006102	106067	000171	RORB	UBM6+1	;(RO)=140000, CC=1010
1865	006106	103402		BCS	RORB6	
1866	006110	102001		BVC	RORB6	
1867	006112	100401		BMI	.+4	
1868	006114	104400		RORB6:	HLT	
1869						
1870	006116	105167	000154	COMB	UBM6	;(RO)=140377 CC=1001
1871	006122	103002		BCC	COMB6	
1872	006124	102401		BVS	COMB6	
1873	006126	100401		BMI	.+4	
1874	006130	104400		COMB6:	HLT	
1875						
1876	006132	000262		SEV		
1877	006134	105467	000137	NEGB	UBM6+1	;(RO)=040377, CC=0001
1878	006140	103002		BCC	NEGB6	
1879	006142	102401		BVS	NEGB6	
1880	006144	100001		BPL	.+4	
1881	006146	104400		NEGB6:	HLT	
1882						
1883	006150	106167	000123	ROLB	UBM6+1	;(RO)=100777, CC=1010
1884	006154	103402		BCS	ROLB6	
1885	006156	102001		BVC	ROLB6	
1886	006160	100401		BMI	.+4	
1887	006162	104400		ROLB6:	HLT	
1888						
1889	006164	106267	000106	ASRB	UBM6	;(RO)=100777, CC=1001
1890	006170	103002		BCC	ASRB6	
1891	006172	102401		BVS	ASRB6	
1892	006174	100401		BMI	.+4	
1893	006176	104400		ASRB6:	HLT	
1894						
1895	006200	105267	000072	INCB	UBM6	;(RO)=100400, CC=0101
1896	006204	103002		BCC	INCB6A	
1897	006206	102401		BVS	INCB6A	
1898	006210	001401		BEQ	.+4	
1899	006212	104400		INCB6A:	HLT	
1900						
1901	006214	105367	000057	DECB	UBM6+1	;(RO)=100000, CC=1001
1902	006220	103003		BCC	DECB6A	
1903	006222	102402		BVS	DECB6A	
1904	006224	001401		BEQ	DECB6A	
1905	006226	100401		BMI	.+4	
1906	006230	104400		DECB6A:	HLT	
1907						
1908	006232	000367	000040	SWAB	UBM6	;(RO)=000200, CC=1000
1909	006236	103401		BCS	SWAB6	
1910	006240	100401		BMI	.+4	
1911	006242	104400		SWAB6:	HLT	
1912						
1913	006244	106167	000026	ROLB	UBM6	;(RO)=000000, CC=0111
1914	006250	103002		BCC	ROLB6A	
1915	006252	102001		BVC	ROLB6A	

```

1916 006254 001401
1917 006256 104400      ROLB6A: BEQ      .+4
1918
1919 006260 005767 000012      TST      UBM6      ;(RC)=000000, CC=0100
1920 006264 103402      BCS      TST6
1921 006266 102401      BVS      TST6
1922 006270 001401      BEQ      .+4
1923 006272 104400      TST6:  HLT
1924
1925 006274 000401      BR       .+4      ;RESERVE A WORD
1926 006276 000000      UBM6:  .WORD    0      ;WORD RESERVED FOR DATA
1927 006300 104000      SCOPE
1928 006302 010702      MOV      PC,R2
1929 006304 062702 000012      ADD      #12,R2
1930 006310 012707 001132      MOV      #RELOC,PC      ;GO RELOCATE PROGRAM CODE
1931 006314 000240      NOP      ;PROGRAM RETURNS HERE+2
1932      ;000000000000 LAST ADDRESS OF CODE TO BE RELOCATED 0000000000
1933

```

```

1934
1935
1936 ;11111111111111111111 FIRST ADDRESS TO BE RELOCATED 1111111111
1937 006316 010700 REL1: MOV PC,RO ;GET PC
1938 006320 005740 TST -(RO) ;RO CONTAINS THE ADDRESS OF REL1
1939 006322 010037 001010 MOV RO,#FRSTAD ;SAVE
1940 006326 010700 MOV PC,RO ;GET CURRENT PC
1941 006330 162700 006330 SUB #,RO ;SUBTRACT RELOCATION FACTOR
1942 006334 010037 001004 MOV RO,#FACTOR ;SAVE RELOCATION FACTOR
1943 006340 010701 MOV PC,R1 ;SET NEW SCOPE PTR
1944
1945 ;CHECK UNARY WORD OPS USING ADDRESS MODE 7
1946 006342 000403 BR UW7 ;RESERVE 3 WORDS FOR ADDRESSES & DATA
1947 006344 000000 .WORD 0 ;CONTAINS ADDRESS OF UW7
1948 006346 000000 UWM7: .WORD 0 ;CONTAINS DATA
1949 006350 000000 .WORD 0 ;CONTAINS ADDRESS OF UWM7
1950
1951 006352 010700 UW7: MOV PC,RO
1952 006354 005740 TST -(RO)
1953 006356 005740 TST -(RO)
1954 006360 005040 CLR -(RO) ;CLEAR TEST DATA
1955 006362 010002 MOV RO,R2
1956 006364 010240 MOV R2,-(RO) ;SET UP ADDRESS
1957 006366 005720 TST (RO)+ ;MOVE RO TO NEXT ADDRESS
1958 006370 005720 TST (RO)+
1959 006372 010210 MOV R2,(RO) ;SET NEXT ADDRESS
1960 006374 010200 MOV R2,RO ;SET RO POINTING TO DATA
1961 006376 000277 SCC
1962 006400 000244 CLZ
1963 006402 005772 000002 TST @2(2) ;(RO)=000000, CC=0100
1964 006406 001401 BEQ .+4
1965 006410 104400 HLT
1966
1967 006412 000277 SCC
1968 006414 005672 177776 SBC @2-2(2) ;(RO)=177777, CC=1001
1969 006420 103002 BCC SBC7
1970 006422 102401 BVS SBC7
1971 006424 100401 BMI .+4
1972 006426 104400 SBC7: HLT
1973
1974 006430 000277 SCC
1975 006432 000241 CLC
1976 006434 006372 000002 ASL @2(2) ;(RO)=177776, CC=1001
1977 006440 103002 BCC ASL7
1978 006442 102401 BVS ASL7
1979 006444 100401 BMI .+4
1980 006446 104400 ASL7: HLT
1981
1982 006450 000257 CCC
1983 006452 005372 000002 DEC @2(2) ;(RO)=177775, CC=1000
1984 006456 103402 BCS DEC7
1985 006460 102401 BVS DEC7
1986 006462 100401 BMI .+4
1987 006464 104400 DEC7: HLT
1988
1989 006466 000262 SEV

```

1990	006470	006272	177776	ASR	2-2(2)	;(RO)=177776, CC=1001
1991	006474	103002		BCC	ASR7	
1992	006476	102401		BVS	ASR7	
1993	006500	100401		BMI	.+4	
1994	006502	104400		HLT		
1995				ASR7:		
1996	006504	000241		CLC		
1997	006506	000262		SEV		
1998	006510	006072	177776	ROR	2-2(2)	;(RO)=077777, CC=0000
1999	006514	101402		ROR7		;BRANCH IF C OR Z IS SET
2000	006516	102401		BVS	ROR7	
2001	006520	100001		BPL	.+4	
2002	006522	104400		HLT		
2003				ROR7:		
2004	006524	000262		SEV		
2005	006526	005472	000002	NEG	2(2)	;(RO)=100001, CC=1001
2006	006532	103002		BCC	NEG7	
2007	006534	102401		BVS	NEG7	
2008	006536	100401		BMI	.+4	
2009	006540	104400		HLT		
2010				NEG7:		
2011	006542	000250		CLN		
2012	006544	000372	177776	SWAB	2-2(2)	;(RO)=000600, CC=1000
2013	006550	103401		BCS	SWAB7	
2014	006552	100401		BMI	.+4	
2015	006554	104400		HLT		
2016				SWAB7:		
2017	006556	000262		SEV		
2018	006560	005172	000002	COM	2(2)	;(RO)=177177, CC=1001
2019	006564	103002		BCC	COM7	
2020	006566	102401		BVS	COM7	
2021	006570	100401		BMI	.+4	
2022	006572	104400		HLT		
2023				COM7:		
2024	006574	000372	000002	SWAB	2(2)	;(RO)=077776, CC=1000
2025	006600	100401		BMI	.+4	
2026	006602	104400		HLT		
2027						
2028	006604	000277		SCC		
2029	006606	005572	177776	ADC	2-2(2)	;(RO)=077777, CC=0000
2030	006612	103402		BCS	ADC7	
2031	006614	102401		BVS	ADC7	
2032	006616	100001		BPL	.+4	
2033	006620	104400		HLT		
2034				ADC7:		
2035	006622	005272	000002	INC	2(2)	;(RO)=100000, CC=1010
2036	006626	102001		BVC	INC7	
2037	006630	100401		BMI	.+4	
2038	006632	104400		HLT		
2039				INC7:		
2040	006634	000257		CCC		
2041	006636	006172	177776	ROL	2-2(2)	;(RO)=000000, CC=0111
2042	006642	103002		BCC	ROL7	
2043	006644	102001		BVC	ROL7	
2044	006646	001401		BEQ	.+4	
2045	006650	104400		HLT		
				ROL7:		

```

2046 006652 104000          SCOPE
2047
2048          :CHECK UNARY BYTE OPS USING ADDRESS MODE 7
2049 006654 005720          TST      (R0)+
2050 006656 005210          INC      (R0)          ;WORD FOLLOWING UWM7 CONTAINS ADDRESS
2051 006660 005740          TST     -(R0)          ;OF ODD BYTE, R0 POINTS TO DATA WORD
2052 006662 005010          CLR     (R0)          ;PRESET DATA
2053 006664 010701          MOV     PC,R1        ;SET SCOPE PTR
2054          :NOTE: 2(2) REFERENCES THE ODD BYTE, AND 2-2(2) REFERENCES THE EVEN BYTE.
2055
2056 006666 000263          +SEC!SEV          ;SET C AND V
2057 006670 105672 000002  SBCB     2(2)          ;(R0)=177400, CC=1001
2058 006674 103003          BCC     SBCB7
2059 006676 102402          BVS     SBCB7
2060 006700 001401          BEQ     SBCB7
2061 006702 100401          BMI     .+4
2062 006704 104400          SBCB7:  HLT
2063
2064 006706 000277          SCC          ;SET CONDITION CODES
2065 006710 105572 177776  ADCB     2-2(2)          ;(R0)=177401, CC=0000
2066 006714 103403          BCS     ADCB7
2067 006716 102402          BVS     ADCB7
2068 006720 001401          BEQ     ADCB7
2069 006722 100001          BPL     .+4
2070 006724 104400          ADCB7:  HLT
2071
2072 006726 105172 177776  COMB     2-2(2)          ;(R0)=177776, CC=1001
2073 006732 103002          BCC     COMB7
2074 006734 102401          BVS     COMB7
2075 006736 100401          BMI     .+4
2076 006740 104400          COMB7:  HLT
2077
2078 006742 000241          CLC          ;CLEAR CARRY
2079 006744 106072 000002  RORB     2(2)          ;(R0)=077776, CC=0011
2080 006750 103002          BCC     RORB7
2081 006752 102001          BVC     RORB7
2082 006754 100001          BPL     .+4
2083 006756 104400          RORB7:  HLT
2084
2085 006760 105272 000002  INCB     2(2)          ;(R0)=100376, CC=1011
2086 006764 103002          BCC     INCB7
2087 006766 102001          BVC     INCB7
2088 006770 100401          BMI     .+4
2089 006772 104400          INCB7:  HLT
2090
2091 006774 105372 177776  DECB     2-2(2)          ;(R0)=100375, CC=1001
2092 007000 103002          BCC     DECB7
2093 007002 102401          BVS     DECB7
2094 007004 100401          BMI     .+4
2095 007006 104400          DECB7:  HLT
2096
2097 007010 106372 000002  ASLB     2(2)          ;(R0)=000375, CC=0111
2098 007014 103002          BCC     ASLB7
2099 007016 102001          BVC     ASLB7
2100 007020 001401          BEQ     .+4
2101 007022 104400          ASLB7:  HLT

```


2102						
2103	007024	000241		CLC		;CLEAR CARRY
2104	007026	106272	177776	ASRB	2-2(2)	;(RO)=000376, CC=1001
2105	007032	103002		BCC	ASRB7	
2106	007034	102401		BVS	ASRB7	
2107	007036	100401		BMI	.+4	
2108	007040	104400		ASRB7:	HLT	
2109						
2110	007042	105472	000002	NEGB	2(2)	;(RO)=000376, CC=0100
2111	007046	103402		BCS	NEGB7	
2112	007050	102401		BVS	NEGB7	
2113	007052	001401		BEQ	.+4	
2114	007054	104400		NEGB7:	HLT	
2115						
2116	007056	000262		SEV		
2117	007060	106172	177776	ROLB	2-2(2)	;(RO)=00374, CC=1001
2118	007064	103002		BCC	ROLB7	
2119	007066	102401		BVS	ROLB7	
2120	007070	100401		BMI	.+4	
2121	007072	104400		ROLB7:	HLT	
2122						
2123	007074	105272	177776	INCB	2-2(2)	;(RO)=000375, CC=1001
2124	007100	105272	177776	INCB	2-2(2)	;(RO)=000376, CC=1001
2125	007104	105572	177776	ADCB	2-2(2)	;(RO)=000377, CC=1000
2126	007110	105172	177776	COMB	2-2(2)	;(RO)=000000, CC=0100
2127	007114	001401		BEQ	.+4	
2128	007116	104400		HLT		
2129	007120	104000		SCOPE		
2130						
2131						;CHECK BINARY OPS USING ADDRESS MODE 0
2132	007122	000277		SCC		;SET CONDITION CODES
2133	007124	010700		MOV	PC, R0	;R0=PC, CC=X/001
2134	007126	103002		BCC	MOV0	
2135	007130	102401		BVS	MOV0	
2136	007132	001001		BNE	.+4	
2137	007134	104400		MOV0:	HLT	
2138						
2139	007136	010002		MOV	R0, R2	;R2=R0
2140	007140	000262		SEV		;SET V
2141	007142	160002		SUB	R0, R2	;R2=000000, CC=0100
2142	007144	103402		BCS	SUB0	
2143	007146	102401		BVS	SUB0	
2144	007150	001401		BEQ	.+4	
2145	007152	104400		SUB0:	HLT	
2146						
2147	007154	000274		CLZ		
2148	007156	010203		MOV	R2, R3	;R2=R3=000000, CC=0100
2149	007160	103401		BCS	MOV0A	
2150	007162	001401		BEQ	.+4	
2151	007164	104400		MOV0A:	HLT	
2152						
2153	007166	000257		CCC		
2154	007170	000272		+SEV!SEN		;SET V & N
2155	007172	020203		CMP	R2, R3	;R2=R3=000000, CC=0100
2156	007174	103403		BCS	CMPO	
2157	007176	102402		BVS	CMPO	

2158	007200	001001		BNE	CMPO	
2159	007202	100001		BPL	.+4	
2160	007204	104400	CMPO:	HLT		
2161						
2162	007206	010002		MOV	R0,R2	;R0=R2
2163	007210	010203		MOV	R2,R3	;R0=R2=R3
2164	007212	060203		ADD	R2,R3	;R3=2*R0
2165	007214	006302		ASL	R2	;R2=2*R0
2166	007216	020203		CMP	R2,R3	;R2=R3=2*R0
2167	007220	001401		BEQ	.+4	
2168	007222	104400		HLT		;ERROR! CHECK ADD INSTRUCTION
2169						
2170						
2171						
2172	007224	005002		CLR	R2	
2173	007226	005202		INC	R2	
2174	007230	000402		BR	2S	
2175	007232	006302	1S:	ASL	R2	
2176	007234	100407		BMI	4S	
2177	007236	010205	2S:	MOV	R2,R5	
2178	007240	000277		SCC		
2179	007242	030205		BIT	R2,R5	;R2=R5
2180	007244	103002		BCC	3S	
2181	007246	102401		BVS	3S	
2182	007250	001370		BNE	1S	
2183	007252	104400	3S:	HLT		
2184	007254	010205	4S:	MOV	R2,R5	
2185	007256	000257		CCC		
2186	007260	030205		BIT	R2,R5	
2187	007262	100401		BMI	.+4	
2188	007264	104400		HLT		
2189						
2190	007266	005002		CLR	R2	
2191	007270	000277		SCC		
2192	007272	050002		BIS	R0,R2	
2193	007274	103 32		BCC	BISO	
2194	007276	1024J1		BVS	BISO	
2195	007300	001001		BNE	.+4	
2196	007302	104400	BISO:	HLT		
2197						
2198	007304	010003		MOV	R0,R3	
2199	007306	000277		SCC		
2200	007310	000244		CLZ		
2201	007312	040003		BIC	R0,R3	
2202	007314	103003		BCC	BICO	
2203	007316	102402		BVS	BICO	
2204	007320	001001		BNE	BICO	
2205	007322	100001		BPL	.+4	
2206	007324	104400	BICO:	HLT		
2207						
2208	007326	010004		MOV	R0,R4	
2209	007330	005104		COM	R4	
2210	007332	040004		BIC	R0,R4	
2211	007334	005104		COM	R4	
2212	007336	020004		CMP	R0,R4	
2213	007340	001401		BEQ	.+4	

```

2214 007342 104400          HLT
2215
2216 007344 010004          MOV      R0,R4
2217 007346 005104          COM      R4
2218 007350 010403          MOV      R4,R3
2219 007352 050003          BIS      R0,R3
2220 007354 103001          BCC      BISOA
2221 007356 100401          BMI      .+4
2222 007360 104400          BISOA:  HLT
2223 007362 005203          INC      R3
2224 007364 001401          BEQ      .+4
2225 007366 104400          HLT
2226 007370 010304          MOV      R3,R4          ;R3=R4=0
2227 007372 005103          COM      R3          ;R3=177777
2228 007374 000261          SEC
2229 007376 006004          ROR      R4          ;SET C
2230 007400 060304          ADD      R3,R4          ;R4=100000
2231 007402 103003          BCC      ADD0          ;R3=177777,R4=077777, CC=0011
2232 007404 102002          BVC      ADD0
2233 007406 001401          BEQ      ADD0
2234 007410 100001          BPL      .+4
2235 007412 104400          ADDC:  HLT
2236 007414 010700          MOV      PC,R0
2237 007416 022020          CMP      (R0)+,(R0)+
2238 007420 020007          CMP      R0,PC
2239 007422 001401          BEQ      .+4
2240 007424 104400          HLT
2241
2242 007426 010700          MOV      PC,R0
2243 007430 062700 000010          ADD      #10,R0
2244 007434 010002          MOV      R0,R2
2245 007436 020700          CMP      PC,R0
2246 007440 001002          BNE      CMPOA
2247 007442 020200          CMP      R2,R0
2248 007444 001401          BEQ      .+4
2249 007446 104400          CMPOA: HLT
2250 007450 104000          SCOPE
2251
2252          ;CHECK BINARY BYTE OPS USING ADDRESS MODE 0.
2253 007452 012703 125252          MOV      #125252,R3
2254 007456 010304          MOV      R3,R4          ;R3=R4=125252
2255 007460 140304          BICB     R3,R4          ;R3=125252, R4=125000
2256 007462 022704 125000          CMP      #125000,R4
2257 007466 001401          BEQ      .+4
2258 007470 104400          HLT          ;ERROR! BICB FAILED
2259
2260 007472 005004          CLR      R4          ;R3=125252, R4=0
2261 007474 150304          BISB     R3,R4          ;R3=125252, R4=000252
2262 007476 022704 000252          CMP      #252,R4
2263 007502 001401          BEQ      .+4
2264 007504 104400          HLT          ;ERROR! BISB FAILED
2265
2266 007506 110404          MOV      R4,R4          ;R4=177652
2267 007510 022704 177652          CMP      #177652,R4          ;MOV B EXTENDS THE SIGN
2268 007514 001401          BEQ      .+4
2269 007516 104400          HLT          ;ERROR! MOV B FAILED

```

```

2270
2271 007520 132704 177525 BITB #177525,R4
2272 007524 001401 BEQ .+4
2273 007526 104400 HLT ;ERROR! BITE FAILED
2274
2275 007530 105104 COMB R4 ;R4=177525
2276 007532 110404 MOVB R4,R4 ;R4=000125
2277 007534 022704 000125 CMP #125,R4
2278 007540 001401 BEQ .+4
2279 007542 104400 HLT
2280
2281 007544 150304 BISB R3,R4 ;R3=125252, R4=000377
2282 007546 105204 INCB R4
2283 007550 005704 TST R4
2284 007552 001401 BEQ .+4
2285 007554 104400 HLT
2286 007556 104000 SCOPE
2287
2288
2289 ;CHECK BINARY OPS USING ADDRESS MODE 1
2290 007560 000402 BR .+6 ;RESERVE TWO WORDS
2291 007562 000000 .WORD 0 ;RESERVED FOR SOURCE DATA
2292 007564 000000 .WORD 0 ;RESERVED FOR DESTINATION DATA
2293 007566 010704 MOV PC,R4
2294 007570 005744 TST -(R4)
2295 007572 005044 CLR -(R4) ;R4 POINT TO DESTINATION DATA
2296 007574 010403 MOV R4,R3
2297 007576 005043 CLR -(R3) ;R3 POINTS TO SOURCE DATA
2298
2299 007600 005113 COM (R3) ;(R3)=177777
2300 007602 005214 INC (R4) ;(R4)=000001
2301 007604 000262 SEV ;SET V
2302 007606 061314 ADD (R3),(R4) ;(R3)=177777,(R4)=000000, CC=0101
2303 007610 103002 BCC ADD1
2304 007612 102401 BVS ADD1
2305 007614 001401 BEQ .+4
2306 007616 104400 HLT ADD1:
2307
2308 007620 000277 SCC
2309 007622 000250 CLN
2310 007624 021314 CMP (R3),(R4) ;(R3)=177777,(R4)=000000, CC=1000
2311 007626 103403 BCS CMP1
2312 007630 102402 BVS CMP1
2313 007632 001401 BEQ CMP1
2314 007634 100401 BMI .+4
2315 007636 104400 HLT CMP1:
2316
2317 007640 000277 SCC
2318 007642 000244 CLZ
2319 007644 031314 BIT (R3),(R4) ;(R3)=177777,(R4)=000000, CC=0101
2320 007646 103002 BCC BIT1
2321 007650 102401 BVS BIT1
2322 007652 001401 BEQ .+4
2323 007654 104400 HLT BIT1:
2324
2325 007656 000277 SCC

```

2326	007660	000245	+CLC!CLZ		
2327	007662	005114	COM	(R4)	;(R4)=177777
2328	007664	161314	SUB	(R3),(R4)	;(R3)=177777,(R4)=000000, CC=0100
2329	007666	103402	BCS	SUB1	
2330	007670	102401	BVS	SUB1	
2331	007672	001401	BEQ	+.4	
2332	007674	104400	HLT		
2333					
2334	007676	105013	CLRB	(R3)	;(R3)=177400
2335	007700	000313	SWAB	(R3)	;(R3)=000377
2336	007702	000270	SEN		
2337	007704	011314	MOV	(R3),(R4)	;(R3)=(R4)=000377
2338	007706	100001	BPL	+.4	
2339	007710	104400	HLT		
2340	007712	000314	SWAB	(R4)	;(R3)=000377,(R4)=177400
2341	007714	000263	+SEC!SEV		;SET C & V
2342	007716	051314	BIS	(R3),(R4)	;(R3)=000377,(R4)=177777, CC=1001
2343	007720	103002	BCC	BIS1	
2344	007722	102401	BVS	BIS1	
2345	007724	100401	BMI	+.4	
2346	007726	104400	HLT		
2347					
2348	007730	041314	BIC	(R3),(R4)	;(R3)=000377,(R4)=177400, CC=1001
2349	007732	103002	BCC	BIC1	
2350	007734	102401	BVS	BIC1	
2351	007736	100401	BMI	+.4	
2352	007740	104400	HLT		
2353					
2354	007742	000262	SEV		;SET V
2355	007744	021314	CMP	(R3),(R4)	;(R3)=000377,(R4)=177400, CC=0001
2356	007746	103003	BCC	CMP1A	
2357	007750	102402	BVS	CMP1A	
2358	007752	001401	BEQ	CMP1A	
2359	007754	100001	BPL	+.4	
2360	007756	104400	HLT		
2361					
2362	007760	005013	CLR	(R3)	;(R3)=000000
2363	007762	000261	SEC		
2364	007764	006013	ROR	(R3)	;(R3)=100000
2365	007766	011314	MOV	(R3),(R4)	;(R3)=(R4)=100000
2366	007770	005114	COM	(R4)	;(R4)=077777
2367	007772	161314	SUB	(R3),(R4)	;(R3)=100000,(R4)=177777, CC=1011
2368	007774	103002	BCC	SUB1A	
2369	007776	102001	BVC	SUB1A	
2370	010000	100401	BMI	+.4	
2371	010002	104400	HLT		
2372					
2373	010004	000277	SCC		
2374	010006	161314	SUB	(R3),(R4)	;(R3)=100000,(R4)=077777, CC=0000
2375	010010	101402	BLOS	SUB1B	;BRANCH IF C OR Z IS SET
2376	010012	102401	BVS	SUB1B	
2377	010014	100001	BPL	+.4	
2378	010016	104400	HLT		
2379					
2380	010020	011314	MOV	(R3),(R4)	;(R3)=100000,(R4)=100000 CC=1000
2381	010022	001401	BEQ	MOV1	

```

2382 010024 100401
2383 010026 104400      MCV1: BMI      .+4
2384
2385 010030 061314      ADD      (R3) (R4)      ;(R3)=100000,(R4)=000000, CC=0111
2386 010032 103003      BCC     ADD1A
2387 010034 102002      BVC     ADD1A
2388 010036 001001      BNE     ADD1A
2389 010040 100001      BPL     .+4
2390 010042 104400      ADD1A: HLT
2391
2392 010044 005113      COM     (R3)      ;(R3)=077777
2393 010046 011314      MOV     (R3) (R4)      ;(R4)=077777
2394 010050 061314      ADD     (R3) (R4)      ;(R3)=077777,(R4)=177776, CC=1010
2395 010052 103402      BCS     ADD1B
2396 010054 102001      BVC     ADD1B
2397 010056 100401      BMI     .+4
2398 010060 104400      ADD1B: HLT
2399
2400 010062 062714 000002      ADD     #2, (R4)
2401 010066 005714      TST     (R4)      ;CHECK FINAL RESULT
2402 010070 001401      BEQ     .+4
2403 010072 104400      HLT
2404 010074 104003      SCOPE
2405
2406      ;CHECK BINARY BYTE OPS USING ADDRESS MODE 1
2407 010076 000402      BR      .+6
2408 010100 000000      .WORD  0
2409 010102 000000      .WORD  0
2410 010104 010705      MOV     PC,R5
2411 010106 005745      TST     -(R5)
2412 010110 005045      CLR     -(R5)      ;(R5)=000000
2413 010112 010502      MOV     R5,R2
2414 010114 005042      CLR     -(R2)      ;(R2)=000000
2415 010116 005202      INC     R2      ;R2 POINTS TO ODD BYTE
2416 010120 105112      COMB    (R2)      ;(R2)=177400
2417
2418 010122 000277      SCC
2419 010124 111215      MOVB    (R2) (R5)      ;(R2)=177400,(R5)=000377,CC=1001
2420 010126 103005      BCC     MOVB1
2421 010130 102404      BVS     MOVB1
2422 010132 001403      BEQ     MOVB1
2423 010134 100002      BPL     MOVB1
2424 010136 105215      INCB    (R5)      ;CHECK RESULT
2425 010140 001401      BEQ     .+4
2426 010142 104400      MOVB1: HLT
2427
2428 010144 106312      ASLB    (R2)      ;SHIFT (R2) UNTIL
2429 010146 102376      BVC     .-2      ;(R2)=000000
2430 010150 106012      RORB    (R2)      ;(R2)=100000
2431 010152 105315      DECB    (R5)      ;(R5)=00377
2432 010154 106015      RORB    (R5)      ;(R5)=000177
2433 010156 000257      CCC
2434 010160 121512      CMPB    (R5) (R2)      ;(R5)=000177,(R2)=100000, CC=1010
2435 010162 102001      BVC     CMPB1
2436 010164 100401      BMI     .+4
2437 010166 104400      CMPB1: HLT

```

2438					
2439	010170	005003	CLR	R3	
2440	010172	000261	SEC		
2441	010174	006003	ROR	R3	;R3=100000
2442	010176	050315	BIS	R3,(R5)	;(R5)=100177
2443	010200	000273	+SEC!SEV!SEN		;SET C,V,&N
2444	010202	131215	BITB	(R2),(R5)	;(R2)=100000,(R5)=100177,CC=0101
2445	010204	103002	BCC	BITB1	
2446	010206	102401	BVS	BITB1	
2447	010210	001401	BEQ	.+4	
2448	010212	104400	BITB1:	HLT	
2449					
2450	010214	151215	BISB	(R2),(R5)	;(R2)=100000,(R5)=100377,CC=1001
2451	010216	103001	BCC	BISB1	
2452	010220	100401	BMI	.+4	
2453	010222	104400	BISB1:	HLT	
2454					
2455	010224	141215	BICB	(R2),(R5)	;(R2)=100000,(R5)=100177,CC=0001
2456	010226	103002	BCC	BICB1	
2457	010230	001401	BEQ	BICB1	
2458	010232	100001	BPL	.+4	
2459	010234	104400	BICB1:	HLT	
2460					
2461	010236	105112	COMB	(R2)	;(R2)=077400,(R5)=100177
2462	010240	121215	CMPB	(R2),(R5)	
2463	010242	001401	BEQ	.+4	
2464	010244	104400	HLT		
2465					
2466	010246	141512	BICB	(R5),(R2)	;(R5)=100177,(R2)=000000,CC=0100
2467	010250	001002	BNE	BICB1A	
2468	010252	105712	TSTB	(R2)	
2469	010254	001401	BEQ	.+4	
2470	010256	104400	BICB1A:	HLT	
2471					
2472	010260	000402	BR	.+6	;RESERVE TWO WORDS FOR DATA
2473	010262	000000	.WORD	0	;SOURCE DATA
2474	010264	000000	.WORD	0	;DEST DATA
2475	010266	010705	MOV	PC,R5	
2476	010270	005745	TST	-(R5)	
2477	010272	105045	CLRB	-(R5)	;R5 POINTS TO DEST ODD BYTE
2478	010274	010504	MOV	R5,R4	
2479	010276	105044	CLRB	-(R4)	;R4 POINTS TO DEST EVEN BYTE
2480	010300	010403	MOV	R4,R3	
2481	010302	105043	CLRB	-(R3)	;R3 POINTS TO SOURCE ODD BYTE
2482	010304	010302	MOV	R3,R2	
2483	010306	105042	CLRB	-(R2)	;R2 POINTS TO SOURCE EVEN BYTE
2484					
2485					;COMMENTS ARE LEAST SIGNIFICANT 4 BITS OF BYTES POINTED TO BY R2,R3
2486					;R4, AND R5 RESPECTIVELY AND THE REMAINING BITS ARE 0'S.
2487	010310	000261	SEC		;SET CARRY
2488					;(R2),(R3),(R4),(R5)
2489	010312	106112	ROLB	(R2)	;0001,0000,0000,0000
2490	010314	111214	MOVB	(R2),(R4)	;0001,0000,0001,0000
2491	010316	106112	ROLB	(R2)	;0010,0000,0001,0000
2492	010320	111213	MOVB	(R2),(R3)	;0010,0010,0001,0000
2493	010322	106112	ROLB	(R2)	;0100,0010,0001,0000

2494	010324	111315		MOVB	(R3), (R5)	;0100,0010,0001,0010
2495	010326	106112		ROLB	(R2)	;1000,0010,0001,0010
2496	010330	106113		ROLB	(R3)	;1000,0100,0001,0010
2497	010332	151215		BISB	(R2), (R5)	;1000,0100,0001,1010
2498	010334	131512		BITB	(R5), (R2)	;1000,0100,0001,1010
2499	010336	001426		BEQ	BIN1	
2500	010340	151314		BISB	(R3), (R4)	;1000,0100,0101,1010
2501	010342	131413		BITB	(R4), (R3)	;1000,0100,0101,1010
2502	010344	001423		BEQ	BIN1	
2503	010346	105213		INCB	(R3)	;1000,0101,0101,1010
2504	010350	121314		CMPB	(R3), (R4)	;1000,0101,0101,1010
2505	010352	001020		BNE	BIN1	
2506	010354	106113		ROLB	(R3)	;1000,1010,0101,1010
2507	010356	121315		CMPB	(R3), (R5)	;1000,1010,0101,1010
2508	010360	001015		BNE	BIN1	
2509	010362	106212		ASRB	(R2)	;0100,1010,0101,1010
2510	010364	131214		BITB	(R2), (R4)	;0100,1010,0101,1010
2511	010366	001412		BEQ	BIN1	
2512	010370	106015		RORB	(R5)	;0100,1010,0101,0101
2513	010372	121415		CMPB	(R4), (R5)	;0100,1010,0101,0101
2514	010374	001007		BNE	BIN1	
2515	010376	105314		DECB	(R4)	;0100,1010,0100,0101
2516	010400	141214		BICB	(R2), (R4)	;0100,1010,0000,0101
2517	010402	001004		BNE	BIN1	
2518	010404	111314		MOVB	(R3), (R4)	;0100,1010,1010,0101
2519	010406	106213		ASRB	(R3)	;0100,0101,1010,0101
2520	010410	141315		BICB	(R3), (R5)	;0100,0101,1010,0101
2521	010412	001401		BEQ	.+4	
2522	010414	104400		HLT		
2523	010416	104000		SCOPE		
2524						
2525						
2526	010420	010405				
2527	010422	012715	000001	MOV	R4, R5	;SET DESTINATION REGISTER
2528	010426	012712	177777	MOV	#1, (R5)	
2529	010432	000257		MOV	#-1, (R2)	
2530	010434	000262		CCC		
2531	010436	062225		SEV		
2532	010440	103002		ADD	(R2)+, (R5)+	; (R2)=177777, (R5)=000000, CC=0101
2533	010442	102401		BCC	ADD2	
2534	010444	001401		BVS	ADD2	
2535	010446	104400		BEQ	.+4	
2536				ADD2:	HLT	
2537	010450	000262		SEV		;SET V
2538	010452	024527	000001	CMP	-(R5), #1	; (R5)=000000, CC=1001
2539	010456	103002		BCC	CMP2	
2540	010460	102401		BVS	CMP2	
2541	010462	100401		BMI	.+4	
2542	010464	104400		CMP2:	HLT	
2543						
2544	010466	054225		BIS	-(R2), (R5)+	; (R2)=177777, (R5)=177777, CC=1001
2545	010470	103001		BCC	BIS2	
2546	010472	100401		BMI	.+4	
2547	010474	104400		BIS2:	HLT	
2548	010476	000277		SCC		
2549	010500	000244		CLZ		

2550	010502	162245		SUB	(R2)+, -(R5)	; (R2)=177777, (R5)=000000, CC=0100
2551	010504	103402		BCS	SUB2	
2552	010506	102401		BVS	SUB2	
2553	010510	001401		BEQ	.+4	
2554	010512	104400		HLT		
2555				SUB2:		
2556	010514	005442		NEG	-(R2)	; (R2)=000001
2557	010516	005115		COM	(R5)	; (R5)=177777
2558	010520	000277		SCC		
2559	010522	000250		CLN		
2560	010524	042225		BIC	(R2)+, (R5)+	; (R2)=000001, (R5)=177776, CC=1001
2561	010526	103003		BCC	BIC2	
2562	010530	102402		BVS	BIC2	
2563	010532	001401		BEQ	BIC2	
2564	010534	100401		BMI	.+4	
2565	010536	104400		HLT		
2566				BIC2:		
2567	010540	012742	125252	MOV	#125252, -(R2)	
2568	010544	012245		MOV	(R2)+, -(R5)	
2569	010546	005125		COM	(R5)+	; (R5)=052525
2570	010550	000262		SEV		
2571	010552	034245		BIT	-(R2), -(R5)	; (R2)=125252, (R5)=052525, CC=0101
2572	010554	103002		BCC	BIT2	
2573	010556	102401		BVS	BIT2	
2574	010560	001401		BEQ	.+4	
2575	010562	104400		HLT		
2576				BIT2:		
2577	010564	000262		SEV		
2578	010566	052225		BIS	(R2)+, (R5)+	; (R2)=125252, (R5)=177777, CC=1001
2579	010570	103002		BCC	BIS2A	
2580	010572	102401		BVS	BIS2A	
2581	010574	100401		BMI	.+4	
2582	010576	104400		HLT		
2583				BIS2A:		
2584	010600	042745	125252	BIC	#125252, -(R5)	; (R5)=052525
2585	010604	005125		COM	(R5)+	; (R5)=125252
2586	010606	024245		CMP	-(R2), -(R5)	
2587	010610	001401		BEQ	.+4	
2588	010612	104400		HLT		
2589						
2590	010614	005012		CLR	(R2)	
2591	010616	005122		COM	(R2)+	; (R2)=177777
2592	010620	162742	000001	SUB	#1, -(R2)	; (R2)=177776, CC=1000
2593	010624	103402		BCS	SUB2A	
2594	010626	102401		BVS	SUB2A	
2595	010630	100401		BMI	.+4	
2596	010632	104400		HLT		
2597	010634	104000		SCOPE		
2598				SUB2A:		
2599	010636	010702		MOV	PC, R2	; GET CURRENT PC
2600	010640	010205		MOV	R2, R5	; MOVE TO R5
2601	010642	124245		CMPB	-(R2), -(R5)	; COMPARE ALL PREVIOUS MEMORY ADDRESSES
2602	010644	001401		BEQ	.+4	
2603	010646	104400		HLT		; ERROR!
2604	010650	020237	001010	CMP	R2, #FRSTAD	; CHECK FOR LOW LIMIT
2605	010654	001372		BNE	1\$	

2606	010656	104000	SCOPE	
2607				
2608			:CHECK BINARY BYTE OPS USING ADDRESS MODES 2 & 4.	
2609	010660	000402	BR	.+6 ;RESERVE TWO WORDS
2610	010662	000000	.WORD	0 ;SOURCE DATA
2611	010664	000000	.WORD	0 ;DESTINATION DATA
2612	010666	010703	MOV	PC,R3
2613	010670	005743	TST	-(R3)
2614	010672	112743	MOV	#200,-(R3)
2615	010676	112743	MOV	#377,-(R3) ;(R3)=100377
2616	010702	010304	MOV	R3,R4
2617	010704	112744	MOV	#177,-(R4)
2618	010710	112744	MOV	#0,-(R4) ;(R4)=077400
2619	010714	001401	BEQ	.+4
2620	010716	104400	HLT	
2621				
2622	010720	152324	BISB	(R3)+,(R4)+ ;(R3)=100377,(R4)=077777
2623	010722	100401	BMI	.+4
2624	010724	104400	HLT	
2625				
2626	010726	122324	CMPB	(R3)+,(R4)+
2627	010730	103402	BCS	CMPB2
2628	010732	102001	BVC	CMPB2
2629	010734	!00001	BPL	.+4
2630	010736	104400	CMPB2: HLT	
2631				
2632	010740	000261	SEC	
2633	010742	134344	BITB	-(R3),-(R4)
2634	010744	103002	BCC	BITB2
2635	010746	102401	BVS	BITB2
2636	010750	001401	BEQ	.+4
2637	010752	104400	BITB2: HLT	
2638				
2639	010754	000244	CLZ	
2640	010756	144344	BICB	-(R3),-(R4) ;(R3)=100377,(R4)=077400
2641	010760	001401	BEQ	.+4
2642	010762	104400	HLT	
2643	010764	104000	SCOPE	
2644				
2645			:CHECK BINARY WORD OPS USING ADDRESS MODES 3 & 5.	
2646	010766	000404	BR	2\$;RESERVE SPACE FOR DATA AND ADDRESSES
2647	010770	000000	.WORD	0 ;CONTAINS ADDRESS OF SOURCE DATA
2648	010772	000000	.WORD	0 ;CONTAINS ADDRESS OF DEST DATA
2649	010774	000000	.WORD	0 ;CONTAINS SOURCE DATA
2650	010776	000000	.WORD	0 ;CONTAINS DEST DATA
2651	011000	010701	2\$: MOV	PC,R1
2652	011002	010100	MOV	R1,R0 ;SET SCOPE PTR
2653	011004	024040	CMP	-(R0),-(R0) ;ADJUST R0
2654	011006	010005	MOV	R0,R5 ;R5 POINTS TO DEST DATA
2655	011010	024545	CMP	-(R5),-(R5) ;SUB 4 FROM R5
2656	011012	010015	MOV	R0,(R5) ;R5 POINTS TO ADDRESS OF DEST DATA
2657	011014	010502	MOV	R5,R2
2658	011016	010004	MOV	R0,R4 ;R4 POINTS TO DEST DATA
2659	011020	005740	TST	-(R0)
2660	011022	010003	MOV	R0,R3 ;R3 POINTS TO SOURCE DATA
2661	011024	010042	MOV	R0,-(R2) ;R2 POINTS TO ADDRESS OF SOURCE DATA

2662	011026	005013		CLR	(R3)		; PRESET SOURCE DATA
2663	011030	005014		CLR	(R4)		; PRESET DEST DATA
2664							
2665	011032	000277		SCC			
2666	011034	000244		CLZ			
2667	011036	163235		SUB	2(R2)+, 2(R5)+		; (R3)=000000, (R4)=000000, CC=0100
2668	011040	103402		BCS	SUB3		
2669	011042	102401		BVS	SUB3		
2670	011044	001401		BEQ	.+4		
2671	011046	104400		SUB3:	HLT		
2672							
2673	011050	052752	100000	BIS	#100000, 2-(R2)		; (R3)=100000
2674	011054	062755	000001	ADD	#1, 2-(R5)		; (R4)=000001
2675	011060	163235		SUB	2(R2)+, 2(R5)+		; (R3)=100000, (R4)=100001, CC=1011
2676	011062	103002		BCC	SUB3A		
2677	011064	102001		BVC	SUB3A		
2678	011066	100401		BMI	.+4		
2679	011070	104400		SUB3A:	HLT		
2680							
2681	011072	005414		NEG	(R4)		; (R4)=077777
2682	011074	035255		BIT	2-(R2), 2-(R5)		; (R3)=100000, (R4)=077777
2683	011076	001401		BEQ	.+4		
2684	011100	104400		HLT			
2685	011102	023235		CMP	2(R2)+, 2(R5)+		
2686	011104	102401		BVS	.+4		
2687	011106	104400		HLT			
2688	011110	005152		COM	2-(R2)		
2689	011112	000257		CCC			
2690	011114	063255		ADD	2(R2)+, 2-(R5)		
2691	011116	102001		BVC	ADD3		
2692	011120	100401		BMI	.+4		
2693	011122	104400		ADD3:	HLT		
2694	011124	000261		SEC			
2695	011126	045235		BIC	2-(R2), 2(R5)+		; (R3)=077777, (R4)=100000
2696	011130	103001		BCC	BIC3		
2697	011132	100401		BMI	.+4		
2698	011134	104400		BIC3:	HLT		
2699							
2700	011136	005155		COM	2-(R5)		; (R4)=077777
2701	011140	023235		CMP	2(R2)+, 2(R5)+		; (R3)=077777, (R4)=077777
2702	011142	001401		BEQ	.+4		
2703	011144	104400		HLT			
2704	011146	104000		SCOPE			
2705							
2706							
2707	011150	000406		BR	15		; RESERVE SPACE FOR ADDRESSES & DATA
2708	011152	000000		.WORD	0		; CONTAINS ADDRESS OF SOURCE DATA (EVEN BYTE)
2709	011154	000000		.WORD	0		; CONTAINS ADDRESS OF SOURCE DATA (ODD BYTE)
2710	011156	000000		.WORD	0		; CONTAINS ADDRESS OF DEST DATA (EVEN BYTE)
2711	011160	000000		.WORD	0		; CONTAINS ADDRESS OF DEST DATA (ODD BYTE)
2712	011162	000000		.WORD	0		; CONTAINS SOURCE DATA
2713	011164	000000		.WORD	0		; CONTAINS DEST DATA
2714							
2715	011166	010700		15:	MOV	PC, R0	
2716	011170	024040			CMP	-(R0), -(R0)	; R0=ADDRESS OF DEST DATA
2717	011172	010003			MOV	R0, R3	; R3

2718	011174	010305		MOV	R3,R5	;R5
2719	011176	005743		TST	-(R3)	;SUB 2 FROM R3
2720	011200	010043		MOV	R0, -(R3)	;R3 POINTS TO ADDRESS OF DEST DATA
2721	011202	005213		INC	(R3)	;ODD BYTE
2722	011204	010043		MOV	R0, -(R3)	;EVEN BYTE
2723	011206	010304		MOV	R3,R4	
2724	011210	005740		TST	-(R0)	;R0=ADDRESS OF SOURCE DATA
2725	011212	010044		MOV	R0, -(R4)	;R4 POINTS TO ADDRESS OF SOURCE DATA
2726	011214	005214		INC	(R4)	;ODD BYTE
2727	011216	010044		MOV	R0, -(R4)	;EVEN BYTE
2728						
2729	011220	000261		SEC		;SET CARRY
2730	011222	012734	177001	MOV	#177001, 2(R4)+	
2731	011226	112734	000200	MOV	#200, 2(R4)+	;SOURCE DATA=100001
2732	011232	115433		MOV	2-(R4), 2(R3)+	
2733	011234	115433		MOV	2-(R4), 2(R3)+	;DEST DATA=000600
2734	011236	103401		BCS	.+4	
2735	011240	104400		HLT		;ERROR! MOV DOES AFFECT C BIT IN PSW
2736	011242	022715	000600	CMP	#600, (R5)	;CHECK DEST DATA
2737	011246	001401		BEQ	.+4	
2738	011250	104400		HLT		;ERROR! INCORRECT RESULT
2739	011252	024343		CMP	-(R3), -(R3)	;POINT R4 BACK TO EVEN BYTE
2740	011254	153433		BISB	2(R4)+, 2(R3)+	
2741	011256	153433		BISB	2(R4)+, 2(R3)+	;DEST DATA=100601
2742	011260	022715	100601	CMP	#100601, (R5)	;CHECK RESULT
2743	011264	001401		BEQ	.+4	
2744	011266	104400		HLT		;ERROR! INCORRECT DEST DATA AFTER BISB
2745	011270	145453		BICB	2-(R4), 2-(R3)	
2746	011272	145453		BICB	2-(R4), 2-(R3)	
2747	011274	133433		BITB	2(R4)+, 2(R3)+	
2748	011276	001002		BNE	BITB3	
2749	011300	135433		BITB	2-(R4), 2(R3)+	
2750	011302	001001		BNE	.+4	
2751	011304	104400		HLT		
2752				BITB3:		
2753	011306	123453		CMPB	2(R4)+, 2-(R3)	
2754	011310	001002		BNE	CMPB3	
2755	011312	123453		CMPB	2(R4)+, 2-(R3)	
2756	011314	001401		BEQ	.+4	
2757	011316	104400		HLT		
2758	011320	104000		SCOPE		
2759						
2760				;CHECK	BINARY OPS USING ADDRESS MODE 6	
2761	011322	000402		BR	.+6	;RESERVE TWO LOCATIONS
2762	011324	000000		SDATA:	.WORD 0	;RESERVED FOR SOURCE DATA
2763	011326	000000		DDATA:	.WORD 0	;RESERVED FOR DESTINATION DATA
2764						
2765	011330	013702	001004	MOV	2#FACTOR, R2	;GET RELOCATION FACTOR AND USE AS AN
2766	011334	010205		MOV	R2,R5	;INDEX VALUE TO POINT TO DATA
2767	011336	005065	011326	CLR	DDATA(5)	;PRESET DESTINATION DATA
2768	011342	012762	000001	MOV	#1, SDATA(2)	;THIS ROUTINE PUT A 1 BIT INTO EVERY
2769	011350	056265	011324	BIS	SDATA(2), DDATA(5)	;OTHER BIT POSITION IN THE DEST-
2770	011356	006362	011324	ASL	SDATA(2)	;INATION ADDRESS (52525)
2771	011362	006362	011324	ASL	SDATA(2)	
2772	011366	103370		BCC	1\$	
2773	011370	022765	052525	CMP	#52525, DDATA(5)	;CHECK RESULT

E05

DDQAA-A BASIC 1: FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 56

2774	011376	001401				BEQ	.+4	
2775	011400	104400				HLT		;ERROR! INCORRECT RESULT
2776	011402	012762	177777	011324		MOV	#-1,SDATA(2)	
2777	011410	046562	011326	011324		BIC	DDATA(5),SDATA(2)	;SOURCE DATA=125252
2778	011416	036265	011324	011326		BIT	SDATA(2),DDATA(5)	
2779	011424	001401				BEQ	.+4	
2780	011426	104400				HLT		;ERROR! BIT INST FAILED
2781	011430	006365	011326			ASL	DDATA(5)	;DDATA=125252
2782	011434	026265	011324	011326		CMP	SDATA(2),DDATA(5)	
2783	011442	001401				BEQ	.+4	
2784								
2785	011444	104400				HLT		;ERROR! CMP INST FAILED
2786	011446	000257				CCC		
2787	011450	066265	011324	011326		ADD	SDATA(2),DDATA(5)	
2788	011456	103002				BCC	ADD6	
2789	011460	102001				BVC	ADD6	
2790	011462	100001				BPL	.+4	
2791	011464	104400			ADD6:	HLT		
2792								
2793	011466	006362	011324			ASL	SDATA(2)	;SDATA=52524
2794	011472	166265	011324	011326		SUB	SDATA(2),DDATA(5)	
2795	011500	103401				BCS	SUB6	
2796	011502	001401				BEQ	.+4	
2797	011504	104400			SUB6:	HLT		
2798								
2799	011506	112700	000377			MOV#B	#377,R0	;R0=177777 (MOV#B %R EXTENDS SIGN)
2800	011512	010062	011324			MOV	R0,SDATA(2)	
2801	011516	012765	177777	011326		MOV	#-1,DDATA(5)	
2802	011524	166500	011326			SUB	DDATA(5),R0	
2803	011530	001401				BEQ	.+4	
2804	011532	104400				HLT		
2805	011534	066265	011324	011326	15:	ADD	SDATA(2),DDATA(5)	
2806	011542	006362	011324			ASL	SDATA(2)	
2807	011546	005162	011324			COM	SDATA(2)	
2808	011552	036265	011324	011326		BIT	SDATA(2),DDATA(5)	
2809	011560	001401				BEQ	.+4	
2810	011562	104400				HLT		
2811	011564	005162	011324			COM	SDATA(2)	
2812	011570	026265	011324	011326		CMP	SDATA(2),DDATA(5)	
2813	011576	001401				BEQ	.+4	
2814	011600	104400				HLT		
2815	011602	026200	011324			CMP	SDATA(2),R0	
2816	011606	001352				BNE	15	
2817	011610	104000				SCOPE		
2818								
2819								
2820								
2821								
2822								
2823	011612	013702	001004			MOV	#FACTOR,R2	;GET INDEX VALUE
2824	011616	010204				MOV	R2,R4	;R2 FOR SOURCE EVEN BYTE INDEX, R4 FOR
2825	011620	010403				MOV	R4,R3	;DEST ODD BYTE, R3 FOR SOURCE EVEN
2826	011622	005203				INC	R3	;AND R5 FOR DEST ODD BYTE
2827	011624	010305				MOV	R3,R5	
2828	011626	000261				SEC		;SET CARRY
2829	011630	012762	125252	011754		MOV	#125252,SDATAB(2)	

;CHECK 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

F05

DD000-A BASIC 11 FAMILY INSTRUCTION EXER.
DD000A.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 57

2830	011636	112763	177125	011754	MOV B	#177125,SDATAB(3)	;SOURCE DATA = 052652
2831	011644	016264	011754	011756	MOV	SDATAB(2),DDATAB(4)	
2832	011652	052764	125125	011756	BIS	#125125,DDATAB(4)	;DEST DATA = 177777
2833	011660	136263	011754	011754	BIT B	SDATAB(2),SDATAB(3)	
2834	011666	001401			BEQ	+.4	
2835	011670	104400			BITB6:	HLT	
2836							
2837	011672	146264	011754	011756	BIC B	SDATAB(2),DDATAB(4)	
2838	011700	103401			BCS	+.4	
2839	011702	104400			HLT		;ERROR MOV,BIS,BIT;BIC DO NOT AFFECT 'C'
2840	011704	126364	011754	011756	CMP B	SDATAB(3),DDATAB(4)	
2841	011712	001401			BEQ	+.4	
2842	011714	104400			HLT		
2843							
2844	011716	146365	011754	011756	BIC B	SDATAB(3),DDATAB(5)	
2845	011724	126265	011754	011756	CMP B	SDATAB(2),DDATAB(5)	
2846	011732	001401			BEQ	+.4	
2847	011734	104400			HLT		
2848							
2849	011736	136564	011756	011756	BIT B	DDATAB(5),DDATAB(4)	
2850	011744	001401			BEQ	+.4	
2851	011746	104400			HLT		
2852	011750	104000			SCOPE		
2853							
2854	011752	000406			BR	UB7	;RESERVE TWO WORDS
2855	011754	000000			SDATAB:	.WORD 0	;RESERVED FOR SOURCE DATA
2856	011756	000000			DDATAB:	.WORD 0	;RESERVED FOR DEST DATA
2857							
2858							
2859							
2860	011760	000000			SBIN7:	.WORD 0	;CONTAINS ADDRESS OF SOURCE DATA
2861	011762	000000			DBIN7:	.WORD 0	;CONTAINS ADDRESS OF DEST DATA
2862	011764	000000				.WORD 0	;CONTAINS SOURCE DATA
2863	011766	000000				.WORD 0	;CONTAINS DEST DATA
2864							
2865	011770	010700			UB7:	MOV PC,R0	
2866	011772	024040				CMP -(R0),-(R0)	
2867	011774	010002				MOV R0,R2	
2868	011776	024242				CMP -(R2),-(R2)	
2869	012000	010012				MOV R0,(R2)	
2870	012002	010203				MOV R2,R3	
2871	012004	024043				CMP -(R0),-(R3)	
2872	012006	010013				MOV R0,(R3)	
2873							
2874	012010	000261			SEC		
2875	012012	012777	100000	177740	MOV	#100000,SBIN7	;SOURCE DATA = 100000
2876	012020	017777	177734	177734	MOV	SBIN7,DBIN7	;DEST DATA = 100000
2877	012026	103001			BCC	MOV7	
2878	012030	100401			BMI	+.4	
2879	012032	104400			MOV7:	HLT	
2880	012034	006377	17772P		ASL	DBIN7	;DEST DATA = 000000
2881	012040	102001			BVC	+.4	
2882	012042	001401			BEQ	+.4	
2883	012044	104400			HLT		
2884							
2885	012046	027777	177706	177706	CMP	SBIN7,DBIN7	;(R2)=100000,(R3)=000000

G05

DD000A-A BASIC 11 FAMILY INSTRUCTION EXER.
DD000AA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 58

2886	012054	103402			BUS	CMP7		
2887	012056	102401			BVS	CMP7		
2888	012060	100401			BMI	.+4		
2889	012062	104400			CMP7:	HLT		
2890								
2891	012064	167777	177670	177670	SUB	25BIN7,20BIN7	; (R2)=100000, (R3)=100000	
2892	012072	103003			BCC	SUB7		
2893	012074	102002			BVC	SUB7		
2894	012076	001401			BEQ	SUB7		
2895	012100	100401			BMI	.+4		
2896	012102	104400			SUB7:	HLT		
2897								
2898	012104	006277	177650		ASR	25BIN7	; (R2)=140000	
2899	012110	067777	177644	177644	ADD	25BIN7,20BIN7	; (R2)=140000, (R3)=040000	
2900	012116	103003			BCC	ADD7		
2901	012120	102002			BVC	ADD7		
2902	012122	001401			BEQ	ADD7		
2903	012124	100001			BPL	.+4		
2904	012126	104400			ADD7:	HLT		
2905								
2906	012130	047777	177624	177624	BIC	25BIN7,20BIN7	; (R2)=140000, (R3)=000000	
2907	012136	001401			BEQ	.+4		
2908	012140	104400			HLT			
2909								
2910	012142	057777	177612	177612	BIS	25BIN7,20BIN7	; (R2)=140000, (R3)=140000	
2911	012150	100401			BMI	.+4		
2912	012152	104400			HLT			
2913								
2914	012154	027777	177600	177600	CMP	25BIN7,20BIN7		
2915	012162	001401			BEQ	.+4		
2916	012164	104400			HLT			
2917	012166	104000			SCOPE			
2918								
2919								
2920								
2921	012170	005000			CLR	RO		
2922	012172	005067	000072		CLR	IS		
2923	012176	010707			MOV	PC,PC		
2924	012200	120707			CMPB	PC,PC		
2925	012202	030707			BIT	PC,PC		
2926	012204	060007			ADD	RO,PC		
2927	012206	105707			TSTB	PC		
2928	012210	005507			ADC	PC		
2929	012212	021007			CMP	(RO),PC		
2930	012214	131007			BITB	(RO),PC		
2931	012216	062707	000000		ADD	RO,PC		
2932	012222	023707	001004		CMP	2#FACTOR,PC		
2933	012226	133707	001004		BITB	2#FACTOR,PC		
2934	012232	000240			NOP			
2935								
2936								
2937	012234	163707	001004		SUB	2#FACTOR,PC	; JUMPS TO UNRELOCATED CODE	
2938	012240	063707	001004		ADD	2#FACTOR,PC	; RETURNS	
2939	012244	000240			NOP			
2940	012246	024607			CMP	-(SP),PC		

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

; 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)

H05

DD000-A BASIC 11 FAMILY INSTRUCTION EXER.
DD000A.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 59

2941	012250	132607		BITB	(SP)+,PC	
2942	012252	026707	000012	CMP	15,PC	
2943	012256	166707	000006	SUB	15,PC	
2944	012262	046707	000002	BIC	15,PC	
2945	012266	000401		BR	.+4	;BRANCH OVER 15
2946	012270	000000		0		
2947	012272	104000		SCOPE		
2948						
2949	012274	010702		MOV	PC,R2	
2950	012276	062702	000012	ADD	#12,R2	
2951	012302	012707	001132	MOV	#RELOC,PC	;GO RELOCATE PROGRAM CODE
2952	012306	000240		NOP		;PROGRAM RETURNS HERE+2
2953						
2954						;1111111111111111 LAST ADDRESS OF CODE TO BE RELOCATED 111111111111
2955						
2956						;2222222222222222 FIRST ADDRESS TO BE RELOCATED 2222222222
2957	012310	010700		REL2: MOV	PC,R0	;GET PC
2958	012312	005740		TST	-(R0)	;R0 CONTAINS THE ADDRESS OF REL2
2959	012314	010037	001010	MOV	R0,#FRSTAD	;SAVE
2960	012320	010700		MOV	PC,R0	;GET CURRENT PC
2961	012322	162700	012322	SUB	#,R0	;SUBTRACT RELOCATION FACTOR
2962	012326	010037	001004	MOV	R0,#FACTOR	;SAVE RELOCATION FACTOR
2963	012332	010701		MOV	PC,R1	;SET NEW SCOPE PTR
2964						;CHECK BINARY BYTE OPS USING ADDRESS MODE 7
2965	012334	000406		BR	BIN87	;RESERVE SPACE FOR ADDRESSES & DATA
2966	012336	000000		SBIN87: .WORD	0	;CONTAINS ADDRESS OF SOURCE EVEN BYTE
2967	012340	000000		.WORD	0	;CONTAINS ADDRESS OF SOURCE ODD BYTE
2968	012342	000000		.WORD	0	;CONTAINS ADDRESS OF DEST EVEN BYTE
2969	012344	000000		.WORD	0	;CONTAINS ADDRESS OF DEST ODD BYTE
2970	012346	000000		DBIN87: .WORD	0	;CONTAINS SOURCE DATA
2971	012350	000000		.WORD	0	;CONTAINS DEST DATA
2972						
2973	012352	010700		BIN87: MOV	PC,R0	
2974	012354	024040		CMP	-(R0),-(R0)	;R0 = ADDRESS OF DEST DATA
2975	012356	010060	177772	MOV	R0,-6(R0)	;LOAD ADDRESS OF DEST EVEN BYTE DATA
2976	012362	010060	177774	MOV	R0,-4(R0)	
2977	012366	005260	177774	INC	-4(R0)	;LOAD ADDRESS OF DEST ODD BYTE DATA

2978	012372	005740		TST	-(RO)	;RO=ADDRESS OF SOURCE DATA
2979	012374	010060	177770	MOV	RO,-10(RO)	;LOAD ADDRESS OF SOURCE EVEN BYTE DATA
2980	012400	010060	177772	MOV	RO,-6(RO)	
2981	012404	005260	177772	INC	-6(RO)	;LOAD ADDRESS OF SOURCE ODD BYTE DATA
2982						
2983	012410	005002		CLR	R2	;SET INDEX REGISTERS
2984	012412	012703	000002	MOV	#2,R3	;DSBINB7(2);DSBINB7(3) REFERENCE EVEN &
2985	012416	012704	177774	MOV	#-4,R4	;ODD BYTE SOURCE DATA; DSBINB7(4);DSBINB7(5)
2986	012422	012705	177776	MOV	#-2,R5	;REFERENCE DEST EVEN& ODD BYTE DATA
2987						
2988						
2989	012426	005020		CLR	(RO)+	;PRESET SOURCE DATA
2990	012430	005010		CLR	(RO)	;PRESET DEST DATA
2991	012432	013746	001004	MOV	#FACTOR,-(SP)	;GET RELOCATION FACTOR
2992	012436	061602		ADD	(SP),R2	;AND ADD TO INDEX VALUES
2993	012440	061603		ADD	(SP),R3	
2994	012442	061604		ADD	(SP),R4	
2995	012444	062605		ADD	(SP)+,R5	
2996						
2997	012446	112773	177777	MOVB	#-1,DSBINB7(3)	;SRC DATA = 177400
2998	012454	132772	000377	BITB	#377,DSBINB7(2)	;CHECK THAT EVEN BYTE WAS NOT AFFECTED
2999	012462	001401		BEQ	+.4	;BY MOVB INSTRUCTION
3000	012464	104400		HLT		
3001						
3002	012466	157374	012336	BISB	DSBINB7(3),DSBINB7(4)	
3003	012474	105274	012346	INCB	DSBINB7(4)	;CHECK THAT BIS SET ALL BITS
3004	012500	001401		BEQ	+.4	
3005	012502	104400		HLT		
3006						
3007	012504	105375	012346	DECB	DSBINB7(5)	;DEST DATA = 177400
3008	012510	005274	012346	INC	DSBINB7(4)	;DEST DATA = 177401
3009	012514	127375	012336	CMPB	DSBINB7(3),DSBINB7(5)	
3010	012522	001401		BEQ	+.4	
3011	012524	104400		HLT		
3012						
3013	012526	147375	012336	BICB	DSBINB7(3),DSBINB7(5)	
3014	012534	001401		BEQ	+.4	
3015	012536	104400		HLT		
3016						
3017	012540	105073	012336	CLRB	DSBINB7(3)	;SRC DATA = 000000
3018						
3019						
3020	012544	157473	012346	BIS7:	BISB DSBINB7(4),DSBINB7(3)	;THIS ROUTINE SETS ALL BITS IN THE SOURCE ODD BYTE BY BISING A BIT FROM
3021	012552	106174	012346	ROLB	DSBINB7(4)	;THE DEST EVEN BYTE INTO THE SOURCE ODD BYTE
3022	012556	103372		BCC	BIS7	
3023	012560	022772	177400	CMP	#177400,DSBINB7(2)	;CHECK RESULT
3024	012566	001401		BEQ	+.4	
3025	012570	104400		HLT		
3026						
3027	012572	000372	012336	SWAB	DSBINB7(2)	;SRC DATA = 000377
3028	012576	112775	000200	MOVB	#200,DSBINB7(5)	;DEST DATA = 100000
3029						
3030	012604	147572	012346	BIC7:	BICB DSBINB7(5),DSBINB7(2)	
3031	012612	106075	012346	RORB	DSBINB7(5)	
3032	012616	103372		BCC	BIC7	
3033	012620	005772	012336	TST	DSBINB7(2)	

3034	012624	001401				BEQ	.+4	
3035	012626	104400				HLT		
3036	012630	104000				SCOPE		
3037								
3038	012632	012702	0C0001		OAERR:	MOV	#1,R2	;LOAD R2 WITH ODD #
3039	012636	010703				MOV	PC,R3	
3040	012640	000401				BR	.+4	;RESERVE SPACE FOR A WORD
3041	012642	000000				.WORD	0	;WILL CONTAIN AN ODD ADDRESS
3042	012644	005723				TST	(R3)+	;STEP R3 TO POINT TO WORD ABOVE
3043	012646	010313				MOV	R3,(R3)	
3044	012650	005213				INC	(R3)	;AND MAKE ODD
3045	012652	012737	013000	000004		MOV	#1\$,2#ERRVEC	;SET ODD ADDRESS & RESERVED INSTRUCTION
3046	012660	063737	001004	000004		ADD	2#FACTOR,2#ERRVEC	
3047	012666	013737	000004	000010		MOV	2#ERRVEC,2#RESVEC	:TO TRAP TO 1\$ BELOW
3048								
3049	012674	000277				SCC		;SET ALL CC'S
3050	012676	160212				SUB	R2,(R2)	
3051	012700	104400				HLT		
Z 3052	012702	060222				ADD	R2,(R2)+	
3053	012704	104400				HLT		
3054	012706	006342				ASL	-(R2)	
3055	012710	104400				HLT		
3056	012712	106512				MFPD	(R2)	;NOTE: MAY BE RESERVED
3057	012714	104400				HLT		
3058	012716	170412				CLRF	(R2)	
3059	012720	104400				HLT		
3060	012722	042202				BIC	(R2)+,R2	
3061	012724	104400				HLT		
3062	012726	164202				SUB	-(R2),R2	
3063	012730	104400				HLT		
3064	012732	155202				BISB	2-(R2),R2	
3065	012734	104400				HLT		
3066	012736	105532				ADCB	2(R2)+	
3067	012740	104400				HLT		
3068	012742	163302				SUB	2(R3)+,R2	
3069	012744	104400				HLT		
3070	012746	005733				TST	2(R3)+	
3071	012750	104400				HLT		
3072	012752	106533				MFPD	2(R3)+	
3073	012754	104400				HLT		
3074	012756	170453				CLRD	2-(R3)	
3075	012760	104400				HLT		
3076	012762	137702	177775			BITB	2.+1,R2	
3077	012766	104400				HLT		
3078	012770	105477	177773			NEGB	2.-1	
3079	012774	104400				HLT		
3080	012776	000406				BR	2\$	
3081								
3082	013000	062716	000002		1\$:	ADD	2,(SP)	;ADJUST RETURN PC
3083	013004	052766	000017	000002		BIS	#17,2(SP)	;SET CONDITION CODES ON RETURN
3084	013012	000002				RTI		
3085								
3086	013014	012706	000500		2\$:	MOV	#STKPTR,SP	;RESET STACK PTR
3087	013020	012737	000006	000004		MOV	#ERRVEC+2,2#ERRVEC	
3088	013026	012737	000012	000010		MOV	#RESVEC+2,2#RESVEC	
3089	013034	104000				SCOPE		

```

3090
3091          ;CHECK JMP INSTRUCTIONS
3092
3093 013036 010700          MOV    PC,R0
3094 013040 062700 000012  ADD    #12,R0          ;SET ADDRESS FOR JMP INST
3095 013044 000277          SCC          ;SET CC'S
3096 013046 000110          JMP    (R0)
3097 013050 000402          BR     .+6
3098 013052 000250          CLN          ;JMP INST JUMPS HERE
3099 013054 000775          BR     .-4
3100
3101 013056 103003          BCC    JMP1
3102 013060 102002          BVC    JMP1
3103 013062 001001          BNE    JMP1
3104 013064 100001          BPL    .+4
3105 013066 104400          JMP1:  HLT          ;ERROR! INCORRECT CC'S AFTER JMP
3106
3107 013070 005002          CLR    R2          ;SET INDICATOR
3108 013072 010703          MOV    PC,R3
3109 013074 000401          BR     .+4          ;RESERVE WORD FOR JMP ADDRESS
3110 013076 000000          .WORD 0          ;CONTAINS ADDRESS FOR JMP INST
3111 013100 005723          TST    (R3)+
3112 013102 010313          MOV    R3,(R3)
3113 013104 010300          MOV    R3,R0
3114 013106 062713 000022  ADD    #22,(R3)          ;(R3) IS JMP ADDRESS
3115 013112 010300          MOV    R3,R0
3116 013114 000133          JMP    @R3+          ;JUMP TO ADDRESS CONTAINED IN R3
3117 013116 000402          BR     .+6
3118 013120 005102          COM    R2          ;COMPLEMENT INDICATOR
3119 013122 000775          BR     -4
3120 013124 005202          INC    R2          ;CHECK INDICATOR
3121 013126 001003          BNE    JMP3
3122 013130 005720          TST    (R0)+
3123 013132 020003          CMP    R0,R3          ;CHECK AUTO-INC R3
3124 013134 001401          BEQ    .+4
3125 013136 104400          JMP3:  HLT
3126
3127 013140 005002          CLR    R2          ;SET INDICATOR
3128 013142 010704          MOV    PC,R4          ;SET UP JMP REGISTER
3129 013144 010400          MOV    R4,R0          ;SET UP CHECK REGISTER
3130 013146 000402          BR     1$
3131 013150 005102          COM    R2          ;COMPLEMENT INDICATOR
3132 013152 000403          BR     2$
3133 013154 022424          1$:  CMP    (R4)+,(R4)+
3134 013156 005724          TST    (R4)+          ;R4=JMP ADDRESS
3135 013160 000144          JMP    -(R4)          ;USE R4 AS ADDRESS
3136 013162 005202          2$:  INC    R2          ;CHECK INDICATOR
3137 013164 001003          BNE    JMP4
3138 013166 022020          CMP    (R0)+,(R0)+
3139 013170 020004          CMP    R0,R4          ;CHECK AUTO-DEC R4
3140 013172 001401          BEQ    .+4
3141 013174 104400          JMP4:  HLT
3142
3143 013176 010703          MOV    PC,R3
3144 013200 000401          BR     .+4          ;RESERVE WORD FOR JMP ADDRESS
3145 013202 000000          1$:  .WORD 0          ;CONTAINS JUMP ADDRESS

```

```

3146 013204 005723          TST      (R3)+
3147 013206 010313          MOV      R3,(R3)
3148 013210 062723 000016  ADD      #16,(R3)+
3149 013214 010300          MOV      R3,R0          ;LOAD CHECK REGISTER
3150 013216 000402          BR       3$
3151 013220 005102          2$:     COM      R2
3152 013222 000401          BR       4$
3153 013224 000153          3$:     JMP      2-(R3)      ;JUMP TO 2$ VIA 1$ ABOVE
3154 013226 005202          4$:     INC      R2          ;CHECK INDICATOR
3155 013230 001003          BNE     JMP5
3156 013232 005740          TST     -(R0)
3157 013234 020003          CMP     R0,R3          ;CHECK AUTO-DEC R3
3158 013236 001401          BEQ     .+4
3159 013240 104400          JMP5:   HLT
3160
3161 013242 000402          BR       2$
3162 013244 005102          1$:     COM      R2          ;COMPLEMENT INDICATOR
3163 013246 000402          BR       3$
3164 013250 000167 177770  2$:     JMP      1$
3165 013254 005202          3$:     INC      R2
3166 013256 001401          BEQ     .+4
3167 013260 104400          JMP6:   HLT
3168
3169 013262 012767 013300 000020  MOV     #1$,7$          ;SET UP JMP ADDRESS
3170 013270 063767 001004 000012  ADD     @#FACTOR,7$    ;ADD RELOCATION FACTOR
3171 013276 000402          BR       2$          ;GO TO JMP 27$ INST
3172 013300 005102          1$:     COM      R2          ;COMPLEMENT INDICATOR
3173 013302 000403          BR       3$          ;GO TO CHECK ROUTINE
3174 013304 000177 000000  2$:     JMP      @7$      ;JMP TO 1$ ABOVE VIA 7$
3175 013310 000000          7$:     .WORD   0          ;CONTAINS JMP ADDRESS
3176 013312 005202          3$:     INC      R2          ;CHECK INDICATOR
3177 013314 001401          BEQ     .+4
3178 013316 104400          JMP7:   HLT
3179 013320 104000          SCOPE
3180
3181          ;CHECK JSR INSTRUCTIONS
3182 013322 013705 001004  JSRST:  MOV     @#FACTOR,R5      ;GET RELOCATION FACTOR
3183 013326 012702 013360  MOV     #3$,R2          ;FORM DEST ADRS
3184 013332 060502          ADD     R5,R2          ;ADD RELOCATION FACTOR
3185 013334 000277          SCC
3186 013336 000242          CLV
3187 013340 004512          JSR     R5,(R2)        ;GO TO 3$ VIA R2
3188 013342 005702          1$:     TST     R2          ;CHECK INDICATOR
3189 013344 001017          BNE     JSR1          ;R2 SHOULD=0
3190 013346 023705 001004  CMP     @#FACTOR,R5    ;CHECK THAT RTS R5 RESTORED R5
3191 013352 001014          BNE     JSR1
3192 013354 000414          BR      JSR1A
3193 013356 000205          2$:     RTS     R5          ;EXIT TO SCOPE
3194 013360 103011          3$:     BCC     JSR1      ;RETURN FROM SUBROUTINE
3195 013362 102410          BVS     JSR1          ;CHECK THAT JSR DID NOT
3196 013364 001007          BNE     JSR1          ;AFFECT CC'S
3197 013366 100006          BPL     JSR1
3198 013370 005002          CLR     R2          ;CLEAR INDICATOR
3199 013372 012704 013342  MOV     #1$,R4          ;GET UNRELOCATED RETURN ADDRESS
3200 013376 061604          ADD     (SP),R4        ;ADD RELOCATION FACTOR (OLD R5)
3201 013400 020405          CMP     R4,R5          ;CHECK THAT OLD R5 WAS PLACED ON THE

```

3202	013402	001765			BEQ	2\$;STACK, & THAT NEW R5 CONTAINS RETURN PC
3203	013404	104400		JSR1:	HLT			;ERROR! ABOVE
3204								
3205	013406	013704	001004	JSR1A:	MOV	2#FACTOR,R4		;GET RELOCATION FACTOR
3206	013412	005000			CLR	R0		;SET INDICATOR
3207	013414	012705	013434		MOV	#1\$,R5		
3208	013420	060405			ADD	R4,R5		;SET UP JSR DEFERRED ADRS
3209	013422	010502			MOV	R5,R2		
3210	013424	012715	013452		MOV	#5\$, (R5)		
3211	013430	060415			ADD	R4, (R5)		; (R5)=DEST ADRS
3212	013432	000401			BR	2\$;RESERVE WORD FOR ADDRESS
3213	013434	000000		1\$:	.WORD	0		;CONTAINS DEST ADRS FOR JSR
3214	013436	004435		2\$:	JSR	R4,2(R5)+		;JSR TO 5\$ VIA 1\$ ABOVE
3215	013440	005200		3\$:	INC	R0		;CHECK INDICATOR
3216	013442	001013			BNE	JSR3		
3217	013444	000413			BR	JSR3A		
3218	013446	005100		4\$:	COM	R0		;COMPLEMENT INDICATOR
3219	013450	000204			RTS	4		;RETURN FROM SUBROUTINE
3220	013452	012703	013440	5\$:	MOV	#3\$,R3		;GET UNRELOCATED RETURN ADDRESS
3221	013456	061603			ADD	(SP),R3		;ADD RELOCATION FACTOR (OLD R4)
3222	013460	020403			CMP	R4,R3		
3223	013462	001003			BNE	JSR3		
3224	013464	005722			TST	(R2)+		
3225	013466	020205			CMP	R2,R5		;CHECK AUTO-INC R5
3226	013470	001766			BEQ	4\$;GO TO RTS
3227	013472	104400		JSR3:	HLT			;ERROR ABOVE
3228								
3229	013474	013704	001004	JSR3A:	MOV	2#FACTOR,R4		
3230	013500	010405			MOV	R4,R5		
3231	013502	010703			MOV	PC,R3		
3232	013504	000401			BR	2\$		
3233	013506	000405		1\$:	BR	4\$		
3234	013510	022323		2\$:	CMP	(R3)+, (R3)+		
3235	013512	000277			SCC			
3236	013514	004443			JSR	R4, -(R3)		;GO TO 2\$
3237	013516	104400		3\$:	HLT			
3238	013520	000414			BR	JSR4A		
3239	013522	103012		4\$:	BCC	JSR4		
3240	013524	102011			BVC	JSR4		
3241	013526	001010			BNE	JSR4		
3242	013530	100007			BPL	JSR4		
3243	013532	012702	013516		MOV	#3\$,R2		;GET UNRELOCATED RETURN ADDRESS
3244	013536	061602			ADD	(SP),R2		;ADD RELOCATION FACTOR (OLD R4)
3245	013540	020204			CMP	R2,R4		;CHECK THAT CALCULATED RETURN
3246	013542	001002			BNE	JSR4		;PC = NEW R4
3247	013544	005724			TST	(R4)+		
3248	013546	000204			RTS	R4		
3249	013550	104400		JSR4:	HLT			
3250								
3251								
3252	013552	000401		JSR4A:	BR	2\$		
3253	013554	000405		1\$:	BR	3\$		
3254	013556	010700		2\$:	MOV	PC,R0		
3255	013560	004767	177770		JSR	PC,1\$		
3256	013564	100407			BMI	JSR6A		
3257	013566	104400			HLT			


```

3258 013570 022020          3$:  CMP      (RO)+,(RO)+
3259 013572 020016          CMP      RO,(SP)      ;CHECK THAT RETURN ADDRESS IS ON THE
3260 013574 001401          BEQ      .+4          ;STACK
3261 013576 104400          HLT
3262 013600 000270          SEN
3263 013602 000207          RTS      PC          ;SET N
3264 013604 104000          JSR6A: SCOPE
3265
3266          ;CHECK IOT TRAP (AND ROLB/ASLB)
3267 013606 012737 013640 000020  MOV      #IOT1,#IOTVEC
3268 013614 063737 001004 000020  ADD      @#FACTOR,@#IOTVEC      ;ADD RELOCATION FACTOR
3269 013622 000261          SEC          ;SET CARRY
3270 013624 013737 177776 000022  MOV      @#PSW,@#IOTVEC+2      ;RETAIN CURRENT PSW ON TRAP
3271 013632 005000          CLR      RO          ;PRESET RO
3272 013634 000004          IOT
3273 013636 000403          BR      IOT1A
3274 013640 106100          IOT1:  ROLB      RO          ;ROTATE RO
3275 013642 102376          BVC      .-2          ;UNTIL V SETS (RO=200)
3276 013644 000002          RTI
3277 013646 106300          IOT1A: ASLB      RO          ;SHIFT SHOULD SET CARRY
3278 013650 103004          BCC      IOT1B
3279 013652 102003          BVC      IOT1B
3280 013654 001002          BNE      IOT1B
3281 013656 005700          TST      RO          ;RO SHOULD =0
3282 013660 001401          BEQ      .+4
3283 013662 104400          IOT1B: HLT
3284 013664 012737 000022 000020  MOV      #IOTVEC+2,@#IOTVEC      ;ERROR! ROL/ASL FAILED TO SET CC'S PROPERLY
3285 013672 005037 000022          CLR      @#IOTVEC+2      ;RESTORE IOT TRAP
3286 013676 104000          SCOPE          ;VECTOR
3287
3288          ;CHECK EMT TRAP SEQUENCE
3289 013700 013746 000030          MOV      @#EMTVEC,-(SP)      ;SAVE SCOPE PTR
3290 013704 012737 013740 000030  MOV      #EMT1,@#EMTVEC      ;SET EMT TRAP VECTOR
3291 013712 063737 001004 000030  ADD      @#FACTOR,@#EMTVEC      ;ADD RELOCATION FACTOR
3292 013720 000262          SEV          ;SET V
3293 013722 013737 177776 000032  MOV      @#PSW,@#EMTVEC+2      ;RETAIN CURRENT PSW ON TRAP
3294 013730 000265          +SEZ!SEC
3295 013732 104000          EMT
3296 013734 001433          BEQ      EMT1C          ;TRAP TO EMT1
3297 013736 104400          HLT          ;GO TO EMT1C
3298 013740 102027          EMT1:  BVC      EMT1B          ;ERROR! INCORRECT CC'S WERE SET ON RETURN
3299 013742 105100          COMB      RO          ;'V' SHOULD'VE SET ON EMT1 TRAP
3300 013744 105500          ADCB      RO          ;RO=000377,CC'S=1001
3301 013746 106000          RORB      RO          ;RO=000000,CC'S=0101
3302 013750 102023          BVC      EMT1B          ;RO=000200,CC'S=1010
3303 013752 100022          BPL      EMT1B
3304 013754 000257          CCC
3305 013756 105400          NEGB      RO          ;RO=000200,CC'S=1010
3306 013760 102017          BVC      EMT1B
3307 013762 100016          BPL      EMT1B
3308 013764 000242          CLV          ;CLEAR 'V'
3309 013766 000261          SEC          ;AND SET 'C'
3310 013770 105300          DECB      RO          ;RO=000177,CC'S=0011
3311 013772 102012          BVC      EMT1B
3312 013774 100411          BMI      EMT1B
3313 013776 000242          CLV          ;CLEAR 'V'

```

```

3314 014000 105200      INCB      RO          ;RO=00020C,CC'S=1011
3315 014002 103006      BCC      EMT1B
3316 014004 102005      BVC      EMT1B
3317 014006 100004      BPL      EMT1B
3318 014010 000242      CLV
3319 014012 106200      ASRB     RO          ;CLEAR 'V'
3320 014014 102776      BVS     .-2         ;SHIFT RO UNTIL 'V' CLEARS
3321 014016 000401      BR      .+4
3322 014020 104400      EMT1B:  HLT
3323 014022 000002      RTI     ;ERROR!
3324 014024 105500      EMT1C:  AOCB     RO          ;EXIT WITH RO=000377
3325 014026 103003      BCC     EMT1D     ;RO=000000
3326 014030 001002      BNE     EMT1D
3327 014032 005700      TST     RO
3328 014034 001401      BEQ     .+4
3329 014036 104400      EMT1D:  HLT
3330 014040 012637 000030      MOV     (SP)+,2*EMTVEC ;RESTORE SCOPE PTR
3331 014044 005037 000032      CLR     2*EMTVEC+2
3332 014050 104000      SCOPE
3333
3334      ;CHECK TRAP INSTRUCTION TRAP SEQUENCE
3335      HLT=IOT      ;REDEFINE HLT
3336 014052 013737 000034 000020      MOV     2*TRAPVEC,2*IOTVEC ;SET IOT (HLT) TRAP VECTOR
3337 014060 012737 014126 000034      MOV     2*TRAP1,2*TRAPVEC ;SET TRAP VECTOR
3338 014066 063737 001004 000034      ADD     2*FACTOR,2*TRAPVEC ;ADD RELOCATION FACTOR
3339 014074 000270      SEN
3340 014076 013737 177776 000036      MOV     2*PSW,2*TRAPVEC+2 ;SET N
3341 014104 000261      SEC     ;RETAIN CURRENT PSW ON TRAP
3342 014106 110700      MOVVB  PC,RO      ;SET CARRY
3343 014110 000264      SEZ
3344 014112 104400      TRAP   ;SET Z BIT
3345 014114 103401      IS:    BCS     .+4      ;TRAP TO TRAP1
3346 014116 000004      HLT
3347 014120 001401      BEQ     .+4
3348 014122 000004      HLT
3349 014124 003412      BR      TRAP1C
3350 014126 100401      TRAP1: BMI     .+4      ;N BIT GOT SET ON TRAP
3351 014130 000004      HLT
3352 014132 062700 000004      ADD     2,RO
3353 014136 120016      CMPB   RO,(SP)   ;CHECK LOW BYTE OF RETURN PC ON
3354 014140 001401      BEQ     .+4      ;STACK
3355 014142 000004      HLT
3356 014144 124646      CMPB   -(SP),-(SP)
3357 014146 032626      BIT    (SP)+,(SP)+
3358 014150 000002      RTI     ;RETURN TO INST FOLLOWING TRAP (IS)
3359
3360 014152 013737 000020 000034      TRAP1C: MOV     2*IOTVEC,2*TRAPVEC ;RESTORE TRAP (HLT) TRAP VECTOR
3361 014160 012737 000200 000036      MOV     2*PTY4,2*TRAPVEC+2
3362 014166 012737 000022 000020      MOV     2*IOTVEC+2,2*IOTVEC
3363 014174 005037 000022      CLR     2*IOTVEC+2
3364 014200 104000      SCOPE
3365 014202 104400      HLT=TRAP ;RESTORE HLT TO A TRAP INST
3366
3367 014202 010702      MOV     PC,R2
3368 014204 062702 000012      ADD     2,R2
3369 014210 012707 001132      MOV     2*RELOC,PC ;GO RELOCATE PROGRAM CODE

```

```

3370 014214 000240          NOP          ;PROGRAM RETURNS HERE+2
3371          ;222222222222 LAST ADDRESS OF CODE TO BE RELOCATED 2222222222
3372
3373 014216 010701          MOV      PC,R1          ;SET SCOPE PTR
3374
3375          ;THE BELOW ROUTINE ASCERTAINS WHICH CP & CP OPTIONS THE PROGRAM IS RUN-
3376          ;NING ON AND SETS AN INDICATOR IN OPT.CP ACCORDINGLY.
3377 014220 005767 164554    CPCHK:  TST      ICNT          ;CHECK IF PASS 0
3378 014224 001036          BNE     REL3          ;DO NOT EXECUTE ROUTINE IF NOT PASS 0
3379 014226 012737 000002 00C006    MOV     #RTI,2#ERRVEC+2 ;SET UP ERROR TRAP TO RETURN
3380 014234 012700 000003          MOV     #3,R0
3381 014240 000261          SEC
3382 014242 005737 177772    TST     2#PIRQ          ;R0=3 IF 11/45
3383 014246 005600          SBC     R0             ;R0=2 IF 11/40
3384 014250 000261          SEC
3385 014252 105737 177777    TSTB   2#PSW+1        ;R0=1 IF 11/20
3386 014256 005600          SBC     R0
3387 014260 005037 177700    CLR     2#177700       ;R0=0 IF 11/05
3388 014264 006300          ASL     R0             ;SHIFT INDICATOR
3389 014266 010027          MOV     R0,(PC)+      ;SET CP INDICATOR
3390 014270 000000          OPT.CP: .WORD 0       ;CONTAINS OPTION & CP INDICATORS
3391          ;EVEN BYTE: 0=11/05, 2=11/20, 4=11/40, 6=11/45
3392          ;ODD BYTE: 200=MEM MGMT, 100=EIS, 40=11/45 FLOATING POINT
3393 014272 005037 000006    3$:    CLR     2#ERRVEC+2   ;RESTORE ERROR TRAP TO HALT ON TRAP
3394 014276 005037 000012          CLR     2#RESVEC+2
3395
3396 014302 126727 177762 000004    CMPB   OPT.CP,#4      ;BRANCH IF 11/05 OR 11/20
3397 014310 002404          BLT     REL3
3398 014312 004767 164714    JSR     PC,.PRINT     ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
3399 014316 016677          ILLTEST
3400 014320 000000          HALT
3401
3402
3403
3404          ;3333333333333333 FIRST ADDRESS TO BE RELOCATED 3333333333
3405 014322 010700          REL3:  MOV     PC,R0      ;GET PC
3406 014324 005740          TST     -(R0)         ;R0 CONTAINS THE ADDRESS OF REL3
3407 014326 010037 001010    MOV     R0,2#FRSTAD   ;SAVE
3408 014332 010700          MOV     PC,R0         ;GET CURRENT PC
3409 014334 162700 014334    SUB     #,R0          ;SUBTRACT RELOCATION FACTOR
3410 014340 010037 001004    MOV     R0,2#FACTOR   ;SAVE RELOCATION FACTOR
3411 014344 010701          MOV     PC,R1         ;SET NEW SCOPE PTR
3412
3413          ;CHECK STACK OVERFLOW
3414 014346 013767 177776 000306    OVFLW: MOV     2#PSW,7$    ;SAVE STATUS IN 7$ BELOW
3415 014354 005037 177776          CLR     2#PSW         ;SET KERNEL MODE
3416 014360 010746          MOV     PC,-(SP)      ;PUSH CURRENT PC ONTO STACK
3417 014362 062716 000136          ADD     #2$,-(SP)     ;FORM ADDRESS OF 2$ BELOW
3418 014366 011637 000004          MOV     (SP),2#ERRVEC ;SET ERROR VECTOR
3419 014372 012737 000340 000006    MOV     #340,2#ERRVEC+2 ;SET PRIORITY LEVEL 7 ON TRAP
3420 014400 062716 000074          ADD     #41$-2$, (SP) ;FORM ADDRESS OF 41$ BELOW
3421 014404 012637 000020          MOV     (SP)+,2#IOTVEC ;SET IOT TRAP VECTOR TO 41$
3422 014410 012746 000340          MOV     #340,-(SP)   ;
3423 014414 011637 000022          MOV     (SP),2#IOTVEC+2 ;SET PRIORITY LEVEL 7 ON IOT TRAP
3424 014420 010746          MOV     PC,-(SP)     ;PUSH CURRENT PC ONTO THE STAK
3425 014422 062716 000006          ADD     #6,(SP)      ;ADD OFFSET TO INST FOLLOWING RTI

```

```

3426 014426 000002          RTI          ;SET PRIORITY LEVEL 7,CLEAR 'T' BIT
3427                                ;AND EXECUTE FOLLOWING INST NEXT
3428 014430 012703 000376    MOV      #376,R3
3429 014434 010313          MOV      R3,(R3)      ;LOAD 376 INTO ADDRESS 376
3430 014436 010306          MOV      R3,SP        ;SET STACK PTR AT BOUNDARY
3431
3432                                ;THE BELOW INSTRUCTIONS SHOULD NOT CAUSE AN OVERFLOW TRAP
3433 014440 005716          TST      (SP)         ;BECAUSE TST IS A NON MODIFYING INST
3434 014442 021666 177776    CMP      (SP),-2(SP)  ;SO IS COMPARE
3435 014446 122737 000002 014270  CMPB     #2,2#OPT.CP  ;CHECK IF 11/20 OR 11/05
3436 014454 002411          BLT      12$         ;BRANCH IF 11/40 OR 11/45
3437 014456 001404          BEQ      11$         ;BRANCH IF 11/20
3438 014460 012767 000014 000144  MOV      #14,51$     ;CHANGE CHECK WORD IN 51$ IF 11/05
3439 014466 000407          BR       10$
3440 014470 012767 000034 000134 11$:    MOV      #34,51$     ;CHANGE CHECK WORD IN 51$ IF 11/20
3441 014476 000403          BR       10$
3442 014500 012656          12$:    MOV      (SP)+,2-(SP) ;BECAUSE OF ADDRESS MODE 5
3443 014502 054676 000000          BIS      -(SP),2(SP) ;BECAUSE OF ADDRESS MODE 7
3444 014506 005066 000004 10$:    CLR      4(SP)       ;BECAUSE DEST ADDRESS IS > 376
3445 014512 057636 000000          BIS      2(SP),2(SP)+ ;BECAUSE OF ADDRESS MODE 3
3446 014516 000406          BR       3$         ;BRANCH OVER NON KERNEL MODE TESTS
3447
3448                                ;ERROR SERVICE ROUTINE
3449 014520 012600          2$:    MOV      (SP)+,R0   ;SAVE PC OF INSTRUCTION THAT TRAPPED
3450 014522 012602          MOV      (SP)+,R2   ;SAVE PSW
3451 014524 012706 000500          MOV      #STKPTR,SP ;SET STACK PTR
3452 014530 104400          HLT
3453                                ;ERROR! AN INSTRUCTION THAT WAS NOT
3454                                ;SUPPOSED TO TRAP TRAPPED
3455                                ;R0 CONTAINS PC, R2 CONTAINS PSW
3456 014532 000450          BR       E$         ;EXIT TEST
3457
3458                                ;THE BELOW INSTRUCTIONS WILL CAUSE A STACK OVERFLOW
3459                                ;STACK PTR IS AT 376
3460 014534 062737 000066 000004 3$:    ADD      #45-2$,2#ERRVEC ;SET ERROR VECTOR TO 4$
3461 014542 010306          MOV      R3,SP      ;SET STACK PTR AT 376
3462 014544 012702 000001          MOV      #1,R2
3463 014550 005000          CLR      R0
3464 014552 005016          CLR      (SP)       ;SETS BIT 0 IN R0
3465 014554 006302          ASL      R2         ;SHIFT INDICATOR BIT
3466 014556 105226          INCB     (SP)+      ;SETS BIT 1 IN R0
3467 014560 006302          ASL      R2
3468 014562 060746          ADD      PC,-(SP)   ;SETS BIT 2 IN R0
3469 014564 006302          ASL      R2
3470 014566 000004          IOT
3471 014570 006302          ASL      R2         ;SETS BIT 3 IN R0
3472 014572 004767 000014          JSR      PC,40$     ;SETS BIT 4 IN R0
3473 014576 006302          ASL      R2         ;NOTE: 11/05 WITHOUT ECO # KD11A-00005
3474 014600 050666 177776          BIS      SP,-2(SP)  ;DOES NOT SET BIT 4.
3475 014604 000407          BR       5$         ;SETS BIT 5 IN R0
3476
3477                                ;PROGRAM WILL TRAP HERE ON OVERFLOW TRAP
3478 014606 050200          4$:    BIS      R2,R0   ;SET APPROPRIATE BIT IN R0
3479 014610 000002          RTI          ;RETURN FROM TRAP
3480
3481 014612 000207          40$:   RTS      PC

```

E06

```

3500 014614 012737 000022 000020 41$: MOV #IOTVEC+2,2#IOTVEC
3501 014622 000002 RTI
3502 014624 012706 000500 :CHECK THAT ABOVE INSTRUCTIONS DID TRAP
3503 014630 022700 5$: MOV #STKPTR,SP ;SET STACK PTR
3504 014632 000000 50$: CMP (PC)+,R0 ;EACH INSTRUCTION SET A BIT IN R0
3505 014634 001407 51$: .WORD 0 ;CONTAINS CHECK WORD
3506 014636 105737 014270 BEQ 6$ ;R0= 77 IF 40 OR 45,14 IF 05,34 IF 20
3507 014642 001003 TSTB 2#OPT.CP ;CHECK IF 11/05
3508 014644 022700 000034 BNE 52$ ;BRANCH IF NOT AN 11/05
3509 014650 001401 CMP #34,R0 ;USE ECO KD11A-00005 CHECK WORD
3510 014652 104400 52$: BEQ 6$
3511 014654 012706 000600 :EXIT ROUTINE
3512 014660 012746 6$: MOV #KPTR,SP ;SET KERNEL STACK PTR
3513 014662 000000 7$: MOV (PC)+,-(SP) ;PUSH OLD PSW ONTO STACK
3514 014664 010746 .WORD 0 ;CONTAINS SAVED PSW
3515 014666 062716 000006 MOV PC,-(SP) ;PUSH CURRENT PC ONTO STACK
3516 014672 000002 ADD #6,(SP) ;ADD OFFSET
3517 014674 012706 000500 MOV #STKPTR,SP ;SET STACK PTR
3518 014700 012737 000006 000004 MOV #ERRVEC+2,2#ERRVEC
3519 014706 104000 SCOPE
3520 014710 012737 000002 001114 :CHECK THAT ALL RESERVED INSTRUCTIONS TRAP (TO LOCATION 10)
3521 014716 010701 RESTRP: MOV #2,2#SCOPED ;LIMIT TO TWO ITERATIONS
3522 014720 012702 015040 MOV PC,R1 ;SET SCOPE POINTER
3523 014724 063702 001004 MOV #5$,R2 ;GET ADDRESS OR RESERVED INSTRUCTION TABLE
3524 014730 122737 000004 014270 ADD 2#FACTOR,R2
3525 014736 003402 CMPB #4,2#OPT.CP ;ADJUST TABLE ADDRESS IF 11/20, 11/05
3526 014740 062702 000036 BLE 11$ ;5$=11/45, 11/40 TABLE, 6$=11/05
3527 014744 132737 000040 014271 11$: ADD #6$-5$,R2 ;11/20 TABLE
3528 014752 001402 BITB #40,2#OPT.CP+1 ;CHECK IF 11/45 FLOATING POINT IS AVAIL.
3529 014754 005067 000110 BEQ .+6 ;BRANCH IF NOT AVAILABLE
3530 014760 012737 015016 000010 CLR 50$ ;SET TABLE TERMINATOR AT GROUP 7
3531 014766 063737 001004 000010 MOV #4$,2#RESVEC ;SET RESERVED INSTRUCTION TRAP
3532 014774 012203 1$: MOV (R2)+,R3 ;GET FIRST RESERVED INSTRUCTION
3533 014776 001454 BEQ 7$ ;0 TERMINATES THE TABLE
3534 015000 012204 MOV (R2)+,R4 ;GET LAST RESERVED INSTRUCTION IN GROUP
3535 015002 010317 2$: MOV R3,(PC) ;EXECUTE RESERVED INSTRUCTION
3536 015004 000000 3$: .WORD 0 ;CONTAINS RESERVED INSTRUCTION
3537 015006 104400 HLT ;ERROR! INSTRUCTION IN R3
3538 015010 104400 HLT ;(2$) ABOVE FAILED TO CAUSE A
3539 015012 104400 HLT ;RESERVED INSTRUCTION TRAP
3540 015014 000405 BR 41$
3541 015016 012716 015030 4$: MOV #41$(,SP) ;ADJUST RETURN PC
3542 015022 063716 001004 ADD 2#FACTOR,(SP) ;TO RETURN TO 41$
3543 015026 000002 RTI ;RETURN TO 41$
3544 015030 020304 41$: CMP R3,R4 ;HAS GROUP OF RESERVED INSTRUCTIONS
3545 015032 001760 BEQ 1$ ;BEEN EXECUTED
3546 015034 005203 INC R3 ;INCREMENT THIS RESERVED INSTRUCTION
3547 015036 000761 BR 2$ ;TO NEXT ONE AND EXECUTE
;TABLE OF 11/40,11/45 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)

```

```

3538 015040 000007          55:      7          ;GROUP 1
3539 015042 000077          77          ;
3540 015044 000210          210         ;GROUP 2
3541 015046 000227          227         ;
3542 015050 007000          7000        ;GROUP 3
3543 015052 007777          7777        ;
3544 015054 075040          75040       ;GROUP 4
3545 015056 076777          76777       ;
3546 015060 106400          106400      ;GROUP 5
3547 015062 106477          106477      ;
3548 015064 106700          106700      ;GROUP 6
3549 015066 107777          107777      ;
3550 015070 170000          505:      170000     ;GROUP 7      FLOATING POINT
3551 015072 177777          177777     ;              INSTRUCTIONS
3552 015074 000000          0           ;0 TERMINATES THE TABLE
3553
3554 ;TABLE OF 11/05, 11/20 RESERVED INSTRUCTIONS (0 TERMINATES THE TABLE)
3555 015076 000006          65:      6          ;GROUP 1
3556 015100 000077          77          ;
3557 015102 000210          210         ;GROUP 2
3558 015104 000237          237         ;
3559 015106 006400          6400        ;GROUP 3
3560 015110 007777          7777        ;GROUP 3
3561 015112 070000          70000       ;GROUP 4
3562 015114 077777          77777       ;
3563 015116 106400          106400      ;GROUP 5
3564 015120 107777          107777      ;
3565 015122 170000          170000      ;GROUP 6
3566 015124 177777          177777      ;
3567 015126 000000          0           ;0 TERMINATES THE TABLE
3568 015130 012737 000012 000010 75:  MOV      #RESVEC+2, #RESVEC ;RESTORE RESERVED TRAP TO HALT AT 12
3569 015136 104000          SCOPE
3570
3571 ;CHECK THAT ALL BITS IN THE PROCESSOR STATUS WORD (PSW) CAN BE SET AND
3572 ;CLEARED.
3573 015140 013767 177776 000152 PSWCHK: MOV      #PSW, 3$ ;SAVE STATUS
3574 015146 005037 177776          CLR      #PSW ;CLEAR MODE BITS IN PSW
3575 015152 005046          CLR      -(SP) ;ROUTINE TO CLEAR
3576 015154 010746          MOV      PC, -(SP) ;STATUS WORD (PSW)
3577 015156 062716 000006          ADD      #6, (SP)
3578 015162 000002          RTI ;CLEAR PSW & EXECUTE FOLLOWING INST
3579
3580 015164 013746 000016          MOV      #TBITVEC+2, -(SP)
3581 015170 012704 177776          MOV      #PSW, R4 ;LOAD ADDRESS OF PSW INTO R4
3582 015174 000250          CLN
3583 015176 005714          TST      (R4) ;CHECK THAT PSW WAS CLEARED
3584 015200 001401          BEQ      .+4
3585 015202 104400          HLT ;ERROR! PSW FAILED TO CLEAR
3586 015204 113700 014270          MOVB    #OPT.CP, R0 ;GET CP TYPE
3587 015210 016000 016636          MOV     PSWBIT(0), R0 ;GET BIT MASK FOR TEST R0=THOSE BITS IN
3588 ;THE PSW WHICH CAN BE SET/CLEARED.
3589 015214 005737 014270          TST     #OPT.CP ;CHECK IF MEM MGMT IS AVAILABLE
3590 015220 100002          BPL     10$ ;BRANCH IF NOT AVAILABLE
3591 015222 052700 170000          BIS     #170000, R0 ;SET BITS 15-12 IF MEM MGMT
3592 015226 012702 000001          10$:  MOV     #1, R2 ;R2 = TEST BIT
3593 015232 030200          15:  BIT     R2, R0 ;CHECK IF BIT CAN BE SET/CLEARED

```

3594	015234	001423		BEQ	2\$		
3595	015236	005037	000016	CLR	2#TBITVEC+2		
3596	015242	030227	000020	BIT	R2, #20		;CHECK IF TEST WILL SET 'T' BIT
3597	015246	001403		BEQ	20\$		
3598	015250	012737	000002 000016	MOV	2#RTI, 2#TBITVEC+2		;SET RTI INTO RETURN
3599	015256	005014		CLR	(R4)		;CLEAR PSW
3600	015260	050214	20\$:	BIS	R2, (R4)		;SET R2 INTO PSW
3601	015262	011403		MOV	(R4), R3		;GET BIT
3602	015264	020203		CMP	R2, R3		;CHECK THAT BIT WAS SET IN PSW
3603	015266	001401		BEQ	.+4		
3604	015270	104400		HLT			;ERROR! BIT IN R2 FAILED TO SET IN PSW
3605	015272	000244		CLZ			;CLEAR Z BIT
3606	015274	040214		BIC	R2, (R4)		;CLEAR BIT IN PSW
3607	015276	011403		MOV	(R4), R3		;GET PSW RESULT
3608	015300	001401		BEQ	2\$;BRANCH IF BIC ABOVE CLEARED BIT IN PSW
3609	015302	104400		HLT			;ERROR! BIT IN R2 FAILED TO CLEAR IN PSW
3610	015304	006302	2\$:	ASL	R2		;SHIFT TEST BIT
3611	015306	103351		BCC	1\$;BRANCH IF ALL BITS NOT TESTED
3612	015310	005014		CLR	(R4)		;CLEAR STATUS
3613	015312	012637	000016	MOV	(SP)+, 2#TBITVEC+2		;RESTORE T BIT RETURN
3614	015316	012746		MOV	(PC)+, -(SP)		;PUSH ORIGINAL STATUS ON STACK
3615	015320	000000	3\$:	.WORD	0		;CONTAINS ORIGINAL PSW
3616	015322	010746		MOV	PC, -(SP)		;SET RETURN PC
3617	015324	062716	000006	ADD	#6, (SP)		
3618	015330	000002		RTI			;RETURN
3619	015332	104000	4\$:	SCOPE			
3620							
3621	015334	013704	177776	MOV	2#PSW, R4		;SAVE PSW IN R4
3622	015340	010446		MOV	R4, -(SP)		;PUSH R4 ONTO STACK
3623	015342	112716	000300	MOV	#300, (SP)		;SET PRIORITY LEVEL 6 AND
3624	015346	010746		MOV	PC, -(SP)		;CLEAR 'T' BIT AND EXECUTE
3625	015350	062716	000006	ADD	#6, (SP)		;INSTRUCTION FOLLOWING RTI
3626	015354	000002		RTI			
3627							
3628							;CHECK THAT ALL BITS IN THE CURRENT STACK PTR CAN BE SET/CLEARED
3629	015356	010603	CHKSP:	MOV	SP, R3		;SAVE STACK PTR
3630	015360	000257		CCC			
3631	015362	112706	000377	MOV	#377, SP		;SET STACK PTR = -1
3632	015366	006006	1\$:	ROR	SP		;ROTATE 0 BIT THROUGH ALL BIT
3633	015370	103776		BCS	1\$;BIT POSITIONS
3634	015372	005206		INC	SP		;SHOULD INCREMENT SP TO 0
3635	015374	001403		BEQ	2\$		
3636	015376	010602		MOV	SP, R2		;SAVE ERROR STACK PTR
3637	015400	010306		MOV	R3, SP		;SET STACK PTR FOR TRAP
3638	015402	104400		HLT			;ERROR!
3639							
3640	015404	010306	2\$:	MOV	R3, SP		;RESTORE ORIGINAL STACK PTR
3641							
3642							;CHECK BYTE OPERATIONS USING THE STACK
3643	015406	010600	SPCHK:	MOV	SP, R0		;SAVE STACK PTR
3644	015410	010003		MOV	R0, R3		
3645	015412	005043		CLR	-(R3)		
3646	015414	112746	177777	MOV	#-1, -(SP)		; (SP) = 377
3647	015420	022713	000377	CMP	#377, (R3)		;CHECK THAT ONLY EVEN BYTE WAS AFFECTED
3648	015424	001002		BNE	1\$		
3649	015426	020306		CMP	R3, SP		;CHECK AUTO-DEC

```

3650 015430 001401          BEQ      .+4
3651 015432 104400          1$:    HLT
3652
3653 015434 105226          INCB    (SP)+
3654 015436 005723          TST     (R3)+      ;CHECK RESULT
3655 015440 001002          BNE     2$
3656 015442 020006          CMP     RO,SP      ;CHECK AUTO-INC
3657 015444 001401          BEQ     .+4
3658 015446 104400          2$:    HLT
3659
3660 015450 005143          COM     -(R3)      ;(R3)=177777
3661 015452 144613          BICB    -(SP), (R3)
3662 015454 022713 177400          CMP     #177400, (R3) ;CHECK RESULT
3663 015460 001002          BNE     3$
3664 015462 020603          CMP     SP, R3
3665 015464 001401          BEQ     .+4
3666 015466 104400          3$:    HLT
3667
3668 015470 132627 000377          BITB    (SP)+, #377
3669 015474 001002          BNE     4$
3670 015476 020600          CMP     SP, RO
3671 015500 001401          BEQ     .+4
3672 015502 104400          4$:    HLT
3673
3674 015504 012746 000001          MOV     #1, -(SP)
3675 015510 062706 000002          ADD     #2, SP
3676 015514 012702 177401          MOV     #177401, R2
3677 015520 120246          CMPB   R2, -(SP)
3678 015522 001004          BNE     5$
3679 015524 122602          CMPB   (SP)+, R2
3680 015526 001002          BNE     5$
3681 015530 020006          CMP     RO, SP
3682 015532 001401          BEQ     .+4
3683 015534 104400          5$:    HLT
3684 015536 010446          MOV     R4, -(SP)  ;RESTORE ORIGINAL PSW TO STACK
3685 015540 010746          MOV     PC, -(SP)
3686 015542 062716 000006          ADD     #6, (SP)
3687 015546 000002          RTI
3688 015550 104000          SCOPE
3689
3690          ;CHECK THAT 'C' BIT SETS/CLEARs PROPERLY
3691 015552 012727 177776          CBIT:  MOV     #177776, (PC)+ ;LOAD CONSTANT
3692 015556 000000          1$:    .WORD 0
3693 015560 010700          MOV     PC, RO      ;GET CURRENT PC
3694 015562 162700 000004          SUB     #4, RO      ;POINT RO TO 1$ ABOVE
3695 015566 005520          2$:    ADC     (RO)+    ;ADD 'C' BIT TO 1$ ABOVE
3696 015570 006340          ASL     -(RO)      ;SHIFT 1$
3697 015572 102375          BVC     2$         ;UNTIL 'V' BIT SETS
3698 015574 022767 077776 177754          CMP     #077776, 1$ ;CHECK RESULT
3699 015602 001401          BEQ     .+4
3700 015604 104400          HLT          ;ERROR! INCORRECT RESULT IN 1$ ABOVE
3701          ;RO=ADDRESS OF DATA
3702
3703          ;CHECK THAT CONDITION CODES ARE SET PROPERLY WHEN A NUMBER (CURRENT PC)
3704          ;AND THAT NUMBER +1 ARE COMPARED, AND VICE VERSA.
3705 015606 010700          CMPN:  MOV     PC, RO      ;GET CURRENT PC

```



```

3706 015610 010002      MOV      R0,R2      ;SAVE IN R2
3707 015612 005202      INC      R2         ;MAKE R2 = R0+1
3708 015614 000277      SCC
3709 015616 000251      +CLC!CLN          ;CLEAR C & N BITS
3710 015620 020002      CMP      R0,R2     ;COMPARE # WITH #+1
3711 015622 103003      BCC     1$         ;CARRY BIT SHOULD SET
3712 015624 102402      BVS     1$         ;V BIT SHOULD CLEAR
3713 015626 001401      BEQ     1$         ;Z BIT SHOULD CLEAR
3714 015630 100401      BMI     .+4       ;N BIT SHOULD SET
3715 015632 104400      1$: HLT          ;ERROR! COMPARE # WITH #+1 FAILED TO
3716                                     ;SET CONDITION CODES IN PSW CORRECTLY
3717
3718 015634 000277      SCC
3719 015636 120200      CMPB   R2,R0      ;SET CONDITION CODES IN PSW
3720 015640 103403      BCS     2$         ;COMPARE #+1 WITH #
3721 015642 102402      BVS     2$         ;C BIT SHOULD CLEAR
3722 015644 001401      BEQ     2$         ;V BIT SHOULD CLEAR
3723 015646 100001      BPL     .+4       ;Z BIT SHOULD CLEAR
3724 015650 104400      2$: HLT          ;N BIT SHOULD CLEAR
3725                                     ;ERROR! COMPARE #+1 WITH # FAILED TO SET
3726                                     ;CONDITION CODES IN PSW CORRECTLY
3727
3728                                     ;24 NOP (240) INSTRUCTIONS FOLLOW. THESE NOPS MAY
3729                                     ;BE CHANGED TO TEST CODE IF THE NEED ARISES. THE TEST CODE SHOULD
3730                                     ;BE POSITION INDEPENDENT AND SHOULD RUN WHEN RELOCATED BY THE PROGRAM.
3730 015652 000240      NOP
3731 015654 000240      NOP
3732 015656 000240      NOP
3733 015660 000240      NOP
3734 015662 000240      NOP
3735 015664 000240      NOP
3736 015666 000240      NOP
3737 015670 000240      NOP
3738 015672 000240      NOP
3739 015674 000240      NOP
3740 015676 000240      NOP
3741 015700 000240      NOP
3742 015702 000240      NOP
3743 015704 000240      NOP
3744 015706 000240      NOP
3745 015710 000240      NOP
3746 015712 000240      NOP
3747 015714 000240      NOP
3748 015716 000240      NOP
3749 015720 000240      NOP
3750 015722 000240      NOP
3751 015724 000240      NOP
3752 015726 000240      NOP
3753 015730 000240      NOP
3754 015732 104000      SCOPE
3755
3756 015734 010702      MOV      PC,R2
3757 015736 062702 000012  ADD     #12,R2
3758 015742 012707 001132  MOV     #RELOC,PC      ;GO RELOCATE PROGRAM CODE
3759 015746 000240      NOP          ;PROGRAM RETURNS HERE+2
3760                                     ;33333333333333 LAST ADDRESS OF CODE TO BE RELOCATED 3333333333
3761

```

```

3762
3763
3764 015750 005037 001004      ;CHECK TTY INTERRUPT.
3765 015754 010701      ↑TTYCHK: CLR      @#FACTOR
3766 015756 032737 000100 177564  BIT      #100,@#TPS      ;CHECK IF TTY IS READY
3767 015764 001374      BNE      -6
3768 015766 012737 016042 000064  MOV      #35,@#TPVEC      ;SET TTY INTERRUPT VECTOR
3769 015774 012737 000200 000066  MOV      #200,@#TPVEC+2    ;PRIORITY LEVEL 4 ON INTERRUPT
3770 016002 012767 016100 000064  MOV      #NULLS,MSG        ;ADDRESS OF MESSAGE TO BE TYPED
3771 016010 117737 000060 177566  MOVVB   @MSG,@#TPB        ;TYPE FIRST CHARACTER OF MESSAGE
3772 016016 105737 177564      TSTB    @#TPS
3773 016022 100375      BPL      -4
3774 016024 006237 177564      ASR      @#TPS      ;SET IE BIT IN TTY CSR REG
3775 016030 000001      WAIT    ;WAIT FOR FIRST INTERRUPT
3776 016032 000424      BR
3777 016034 006337 177564      2$:    ASL      @#TPS      ;CLEAR IE BIT
3778 016040 000002      RTI
3779
3780 016042 122777 000012 000024  3$:    CMPB    #12,@MSG      ;BRANCH IF CHAR IS NOT <LF>
3781 016050 001004      BNE      4$
3782 016052 004767 163154      JSR      PC,.PRINT      ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
3783 016056 001760      $CRLF
3784 016060 000404      BR      5$
3785 016062 117737 000006 177566  4$:    MOVVB   @MSG,@#TPB      ;TYPE CHARACTER
3786 016070 001761      BEQ      2$            ;BRANCH IF TERMINATOR
3787 016072 005227      5$:    INC      (PC)+        ;SET MSG TO NEXT CHAR ADDRESS
3788 016074 000000      MSG:    .WORD    0      ;CONTAINS ADDRESS OF CHAR TO BE TYPED
3789 016076 000002      RTI      ;RETURN
3790 016100 020015 000015      NULLS: .ASCIZ  <15><40><15>
3791      .EVEN
3792
3793      ;ROUTINE TO TURN ON KW11-L LINE CLOCK IF AVAILABLE
3794 016104 012737 000002 000006  KW11:  MOV      #RTI,@#ERRVEC+2    ;SET UP DIRECT RTI ON TRAP
3795 016112 012737 016246 000100      MOV      #45,@#LKVEC      ;LOAD INTERRUPT VECTOR
3796 016120 012737 000300 000102      MOV      #300,@#LKVEC+2    ;SET PRIORITY LEVEL 6 ON INT.
3797 016126 000262      SEV
3798 016130 052737 000100 177546      BIS      #100,@#LKS      ;SET INTERRUPT ENABLE
3799 016136 102446      BVS      5$            ;SKIP PRIORITY ARBITRATION TEST
3800      ;BELOW IF NO KW11-L
3801
3802      ;ROUTINE TO CHECK PRIORITY ARBITRATION LOGIC
3803      ;THE BELOW TEST WILL INHIBIT INTERRUPTS ON LEVEL 6 AND ABOVE (LOCKING
3804      ;OUT THE LINE CLOCK) AND THEN SET UP THE TTY TO INTERRUPT. NEXT THE
3805      ;PRIORITY LEVEL WILL BE SET TO 0 ALLOWING INTERRUPTS IN WHICH CASE
3806      ;THE LINE CLOCK (AT LEVEL 6) SHOULD INTERRUPT BEFORE THE TTY (AT LEVEL 4).
3807
3808 016140 132737 000020 177776      BITB    #20,@#PSW      ;CHECK IF 'T' BIT IS SET
3809 016146 001042      BNE      5$            ;DO NOT DO TEST IF SET
3810 016150 112737 000300 177776      MOVVB   #300,@#PSW      ;SET PRIORITY LEVEL = 6
3811 016156 013727 000064      MOV      @#TPVEC,(PC)+    ;SAVE TTY INTERRUPT VECTOR
3812 016162 000000      1$:    .WORD    0      ;CONTAINS CURRENT TTY VECTOR
3813 016164 105737 177564      TSTB    @#TPS      ;CHECK IF READY
3814 016170 100375      BPL      -4      ;WAIT FOR TTY TO BECOME READY
3815 016172 012737 016216 000064      MOV      #25,@#TPVEC      ;SET NEW VECTOR
3816 016200 005227      INC      (PC)+        ;STALL WAITING FOR LINE CLOCK
3817 016202 000000      .WORD    0      ;TO BE READY

```

K06

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 75

3818	016204	012737	016222	000100		MOV	#35, @#LKVEC	;SET LINE CLOCK VECTOR
3819	016212	105037	177776			CLRB	@#PSW	;SET PRIORITY LEVEL 0
3820	016216	104400			2\$:	HLT		;ERROR! EITHER TTY INTERRUPTED
3821						;BEFORE THE LINE CLOCK OR BOTH FAILED TO INTERRUPT		
3822	016220	000415				BR	5\$;EXIT TEST
3823	016222	016737	177734	000064	3\$:	MOV	1\$, @#TPVEC	;RESTORE TTY VECTOR
3824	016230	012737	016246	000100		MOV	#4\$, @#LKVEC	;SET LINE CLOCK VECTOR
3825	016236	105037	177776			CLRB	@#PSW	;RESTORE PRIORITY LEVEL 0
3826	016242	012716	016254			MOV	#5\$, (SP)	;SET RETURN ADDRESS TO 5\$ BELOW
3827								
3828	016246	005267	162524		4\$:	INC	TICKS	;INCREMENT TICK COUNT
3829	016252	000002				RTI		;RETURN
3830								
3831	016254	005037	000006		5\$:	CLR	@#ERRVEC+2	;RESTORE ERROR TRAP TO HALT AT 6
3832								
3833	016260	000240			END:	NOP		
3834	016262	012767	000062	000242	END1:	MOV	#50., TEMP1	
3835	016270	000005			4\$:	RESET		
3836	016272	005367	000234			DEC	TEMP1	
3837	016276	001374				BNE	4\$	
3838	016300	032767	000100	161262		BIT	#SW06, SWR	;DO YOU WANT TO HALT ON
3839								;END OF PASS?
3840	016306	001402				BEG	5\$	
3841	016310	004767	161734			JSR	PC, EOPHLT	
3842	016314	005037	177776		5\$:	CLR	@#PSW	;CLEAR MODE BITS IN PSW
3843	016320	005046				CLR	-(SP)	;CLEAR PSW
3844	016322	012746	016330			MOV	#+6, -(SP)	
3845	016326	000002				RTI		;GO TO NEXT INST WITH PSW=0
3846	016330	012706	000600			MOV	#KPTR, SP	;SET KERNEL STACK PTR (NOT APPLICABLE
3847								;FOR 11/20, 11/05 CP'S)
3848	016334	032737	000100	177564		BIT	#100, @#TPS	;CHECK IF OUTPUT DEVICE IS BUSY
3849	016342	001374				BNE	.-6	;IS AVAILABLE
3850	016344	105737	177570			TSTB	@#SWR	;DELETE END OF PASS TYPE OUT IF SW7=0
3851	016350	100020				BPL	1\$;BRANCH IF SW7 IS DOWN
3852	016352	016702	162422			MOV	ICNT, R2	;GET PASS COUNT
3853	016356	004767	162746			JSR	PC, \$FORMD	;GO TO FORMAT ROUTINE
3854	016362	012702	001700			MOV	#DIGITS+2, R2	;GET ASCII VALUES
3855	016366	012703	001716			MOV	#PASSES, R3	;AND MOVE THEM INTO MESSAGE
3856	016372	012223				MOV	(R2)+, (R3)+	
3857	016374	012223				MOV	(R2)+, (R3)+	
3858	016376	012737	001706	016074		MOV	#PASCNT, @#MSG	;PASS MESSAGE ADRS TO TELETYPE SERVICE
3859	016404	052737	000100	177564		BIS	#100, @#TPS	;SET IE BIT
3860	016412	012737	000610	000024	1\$:	MOV	#PDWN, @#PFVEC	;ENABLE POWER FAIL TRAP
3861	016420	012737	000340	000026		MOV	#340, @#PFVEC+2	;PRIORITY 7 ON POWER FAI.
3862	016426	005267	162346			INC	ICNT	
3863	016432	116700	175632			MOVB	OPT, CP, R0	;GET CP TYPE
3864	016436	026067	016642	162334		CMP	PASTAB(R0), ICNT	;CHECK IF END OF TEST
3865	016444	001002				BNE	2\$;BRANCH IF NOT AT END
3866	016446	000167	000062			JMP	DONE	
3867	016452	016702	162322		2\$:	MOV	ICNT, R2	;GET PASS COUNT
3868	016456	006302				ASL	R2	
3869	016460	046002	016632			BIC	CPPASS(0), R2	;LIMIT PASS COUNT TO 0-6
3870	016464	005037	000016			CLR	@#16	;CLEAR T BIT TRAP ADDRESS
3871	016470	012737	000040	001122		MOV	#40, @#SCOPEF+2	;SET ITERATION COUNT = 40
3872	016476	016216	016626			MOV	PSWTAB(2), (SP)	;PUSH NEXT PASS PSW ON STACK
3873	016502	032716	000020			BIT	#20, (SP)	;WILL 'T' BIT BE SET ON NEXT PASS?

L06

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER.
DDQAAA.P11

MACY11 27(732) 22-SEP-76 14:39 PAGE 76

```

3874 016506 001406          BEQ      3$          ;BRANCH IF NOT
3875 016510 012737 000002 001122      MOV      #2, @#SCOPEF+2 ;SET ITERATION COUNT = 2 FOR 'T' BIT
3876 016516 016737 000006 000016      MOV      RTI1, @#16    ;SET 'T' BIT TRAP TO RETURN VIA 16
3877 016524 012746 002244          3$:      MOV      #START2, -(SP) ;RESART PROGRAM AT START2
3878 016530 000002          RTI1:    RTI          ;RESTART PROGRAM AT START2 WITH NEW PSW
3879                                     ;(FROM TABLE BELOW) NOTE: THE RTI IS
3880                                     ;CHANGED TO AN RTT IF NOT AN 11/05, 11/20
3881
3882 016532 000000          TEMP1:  .WORD 0
3883                                     ;ROUTINE TO SET UP MEMORY MANAGEMENT TO RELOCATE PROGRAM CODE ABOVE 28K
3884
3885 016534 032737 000100 177564      DONE:   BIT      #100, @#TPS ;WAIT FOR TTY OUTPUT TO FINISH
3886 016542 001374          BNE     DONE
3887 016544 105737 177564          TSTB   @#TPS          ;WAIT FOR LAST CHARACTER TO BE PRINTED
3888 016550 100375          BPL    .-4
3889 016552 005027          CLR   (PC)+
3890 016554 000000          1$:    .WORD 0
3891 016556 005267 177772          2$:    INC   1$          ;DELAY WAITING FOR TELETYPE TO FINISH
3892 016562 001375          BNE   2$          ;TYPING CHARACTER BEFORE ISSUING RESET
3893 016564 000005          RESET
3894 016566 105737 177570          TSTB   @#SWR
3895 016572 100003          BPL   3$
3896 016574 004767 162432          JSR   PC, .PRINT    ;PRINT MESSAGE BEGINING AT FOLLOWING ADRS
3897 016600 016766          ENDMSG
3898 016602 013702 000042          3$:    MOV   @#42, R2    ;CHECK DDP/ACT11 MONITOR HOOK
3899 016606 001405          BEQ   DONE1
3900 016610 000005          RESET
3901 016612 004712          LOGICAL: JSR   PC, (R2) ;GO TO DDP/ACT11 MONITOR VIA 42
3902 016614 000240          NOP
3903 016616 000240          NOP
3904 016620 000240          NOP
3905 016622 000137 002240          DONE1: JMP   @#START3      ;RESTART PROGRAM
3906
3907                                     ;THE BELOW TABLE REPRESENTS THE 'NEW' PSW SET BY THE PROGRAM ON
3908                                     ;SUCCESSIVE PASSES.
3909                                     ;NOTE THE BELOW TABLE MAY BE MODIFIED TO CAUSE THE PROGRAM TO RUN
3910                                     ;UNDER USER DEFINED PARAMETERS BY PATCHING IN THE DESIRED PASS PARAMETER
3911                                     ;FOR EXAMPLE TO CAUSE THE PROGRAM TO RUN WITHOUT SETTING THE 'T' BIT
3912                                     ;IN ALL PASSES PATCH OUT THE 'T' BIT IN THE TABLE.
3913 016626 000000          PSWTAB: 000000          ;ALL 11 FAMILY CP'S
3914 016630 000020          000020
3915
3916                                     ;THE BELOW TABLE IS THE 'BIT MASK' USED TO DETERMINE THE INDEX VALUE
3917                                     ;NEEDED TO SET THE 'NEW' PSW.
3918 016632 177774          CPPASS: 177774          ;11/05
3919 016634 177774          177774          ;11/20
3920
3921                                     ;THE BELOW TABLE REPRESENTS THOSE BITS IN THE CP WHICH CAN BE SET/CLEARED
3922 016636 000377          PSWBIT: 000377          ;11/05
3923 016640 000377          000377          ;11/20
3924
3925                                     ;THE BELOW TABLE CONTAINS THE # OF PASSES REQUIRED TO COMPLETE TEST
3926 016642 000002          PASTAB: .WORD 2          ;11/05
3927 016644 000002          .WORD 2          ;11/20
3928
3929 016646 005015 047514 020127      ;MESSAGES
MSG1:  .ASCIZ <15><12>'LOW LIMIT?'

```

3930	016654	044514	044515	037524	
3931	016662	000			
3932	016663	110	043511	020110	MSG2: .ASCIZ 'HIGH LIMIT'
3933	016670	044514	044515	037524	
3934	016676	000			
3935	016677	015	052012	044510	ILLTEST: .ASCIZ <15><12>'THIS TEST INVALID FOR 11/40-11/45 PLEASE RUN DCQKC'<15><12>
3936	016704	020123	042524	052123	
3937	016712	044440	053116	046101	
3938	016720	042111	043040	051117	
3939	016726	030440	027461	030064	
3940	016734	030455	027461	032464	
3941	016742	050040	042514	051501	
3942	016750	020105	052522	020116	
3943	016756	041504	045521	006503	
3944	016764	000012			
3945	016766	005015	042040	050504	ENDMSG: .ASCIZ <15><12>' DDQAA DONE'
3946	016774	040501	042040	047117	
3947	017002	000105			
3948					
3949		017400			
3950	017400	000000			SAVPC: .WORD 0
3951	017402	000000			SAVPS: .WORD 0
3952	017404	000000			SAVIC: .WORD 0
3953		000001			.END

ADC82	004600	1476	1478#						
ADC85	005410	1705	1707#						
ADC86	006076	1858	1859	1861#					
ADC87	006724	2066	2067	2068	2070#				
ADC0	002540	887	888	889	891#				
ADC1	003414	1110	1111	1112	1114#				
ADC2	004410	1408	1410#						
ADC5	005216	1632	1633	1635#					
ADC6	005706	1803	1804	1806#					
ADC7	006620	2030	2031	2033#					
ADD0	007412	2231	2232	2233	2235#				
ADD1	007616	2303	2304	2306#					
ADD1A	010042	2386	2387	2388	2390#				
ADD18	010060	2395	2396	2398#					
ADD2	010446	2532	2533	2535#					
ADD3	011122	2691	2693#						
ADD6	011464	2788	2789	2791#					
ADD7	012126	2900	2901	2902	2904#				
ASLB1	003756	1242	1243	1245#					
ASLB1A	004202	1329	1330	1332#					
ASLB3	005400	1699	1700	1702#					
ASLB4	004704	1513	1514	1515	1517#				
ASLB6	006060	1850	1851	1852	1854#				
ASLB7	007022	2098	2099	2101#					
ASL0	002662	931	932	933	934	936#			
ASL1	003570	1173	1174	1175	1177#				
ASL3	005132	1601	1602	1604#					
ASL4	004502	1440	1441	1442	1444#				
ASL6	005656	1791	1792	1794#					
ASL7	006446	1977	1978	1980#					
ASRB1	004052	1278	1280#						
ASRB1A	004066	1284	1285	1287#					
ASRB2	004650	1498	1499	1501#					
ASRB2A	004666	1506	1507	1509#					
ASRB5	005340	1680	1681	1683#					
ASRB6	006176	1890	1891	1893#					
ASRB7	007040	2105	2106	2108#					
ASR0	002710	945	946	947	949#				
ASR1	003456	1130	1131	1132	1134#				
ASR2	004424	1414	1415	1417#					
ASR3	005116	1595	1597#						
ASR6	005540	1753	1754	1756#					
ASR7	006502	1991	1992	1994#					
BELL	001763	697	720#						
BICB1	010234	2456	2457	2459#					
BICB1A	010256	2467	2470#						
BIC0	007324	2202	2203	2204	2206#				
BIC1	007740	2349	2350	2352#					
BIC2	010536	2561	2562	2563	2565#				
BIC3	011134	2696	2698#						
BIC7	012604	3030#	3032						
BINB7	012352	2965	2973#						
BIN1	010414	2499	2502	2505	2508	2511	2514	2517	2522#
BISB1	010222	2451	2453#						
BIS0	007302	2193	2194	2196#					
BIS0A	007360	2220	2222#						

DECB2	004716	1520	1521	1523#															
DECB5	005456	1728	1730#																
DECB6A	006230	1902	1903	1904	1906#														
DECB7	007006	2092	2093	2095#															
DECO	002602	904	905	906	907	909#													
DEC1	003374	1103	1105#																
DEC1A	003644	1201	1203#																
DEC2	004460	1431	1432	1434#															
DEC5	005150	1608	1609	1610	1612#														
DEC6	005672	1797	1798	1800#															
DEC7	006464	1984	1985	1987#															
DIGITS	001676	632	665	666	675	681	691	708#	3854										
DIGTAB	001666	644	704#																
DISPLA=	177570	421#	796#																
DONE	016534	3866	3885#	3886															
DONE1	016622	3899	3905#																
ENTVEC=	000030	401#	797#	3289	3290*	3291*	3293*	3330*	3331*										
ENT1	013740	3290	3298#																
ENT1B	014020	3298	3302	3303	3306	3307	3311	3312	3315	3316	3317	3322#							
ENT1C	014024	3296	3324#																
ENT1D	014036	3325	3326	3329#															
END	016260	3833#																	
ENDMSG	016766	3897	3945#																
END1	016262	3834#																	
EOPHLT	000250	481#	3841																
ERROR	001412	652#	798																
ERRPC	001723	673	713#																
ERRPCO	001735	689	715#																
ERRVEC=	000004	394#	523*	530*	770*	771*	794*	3045*	3046*	3047	3087*	3379*	3393*	3418*					
		3419*	3459*	3505*	3794*	3831*													
FACTOR	001004	541#	559	581	684	686	807*	1818	1942*	2765	2823	2932	2933	2937					
		2938	2962*	2991	3046	3170	3182	3190	3205	3229	3268	3291	3338	3410*					
		3512	3520	3531	3764*														
		523#	547																
FORXOR	000740																		
FPEVEC=	000244	406#																	
FRSTAD	001010	545#	576	804*	1939*	2604	2959*	3407*											
FRSTME	001012	546#	584	779*	786*														
GSTST	003174	1048#																	
HLT =	104400	442#																	
		918	927	936	942	949	956	964	986	996	1006	1013	1020	1029					
		1045	1080	1098	1105	1114	1121	1127	1134	1141	1149	1156	1164	1169					
		1177	1183	1191	1198	1203	1221	1227	1233	1239	1245	1251	1258	1265					
		1272	1280	1287	1291	1298	1303	1310	1317	1323	1327	1332	1338	1342					
		1348	1351	1368	1373	1380	1387	1393	1398	1402	1410	1417	1424	1428					
		1434	1438	1444	1448	1451	1470	1474	1478	1486	1494	1501	1509	1517					
		1523	1529	1535	1540	1545	1551	1559	1563	1566	1584	1591	1597	1604					
		1612	1618	1624	1629	1635	1642	1648	1672	1676	1683	1688	1696	1702					
		1707	1714	1718	1724	1730	1744	1750	1756	1764	1772	1778	1786	1794					
		1800	1806	1810	1813	1827	1832	1838	1847	1854	1861	1868	1874	1881					
		1897	1893	1899	1906	1911	1917	1923	1965	1972	1980	1987	1994	2002					
		2009	2015	2022	2026	2033	2038	2045	2062	2070	2076	2083	2089	2095					
		2101	2108	2114	2121	2128	2137	2145	2151	2160	2168	2183	2188	2196					
		2206	2214	2222	2225	2235	2240	2249	2258	2264	2269	2273	2279	2285					
		2306	2315	2323	2332	2339	2346	2352	2360	2371	2378	2383	2390	2398					
		2403	2426	2437	2448	2453	2459	2464	2470	2522	2535	2542	2547	2554					
		2565	2575	2582	2588	2596	2603	2620	2624	2630	2637	2642	2671	2679					

		1094*	1102*	1109*	1116*	1123*	1129*	1136*	1144*	1151*	1159*	1167*	1172*	1179*
		1186	1194*	1200*	1460*	1461*	1472*	1489*	1503*	1505*	1519*	1531*	1542	1548*
		1561	1564	1656*	1657	1658	1659	1661	1662	1663*	1665*	1666	1667	1669*
		1674	1685*	1698*	1704*	1716*	1726*	1727*	1928*	1929*	1955*	1956	1959	1960
		2139*	2141*	2148	2155	2162*	2163	2164	2165*	2166	2172*	2173*	2175*	2177
		2179	2184	2186	2190*	2192*	2244*	2247	2413*	2414*	2415*	2416*	2419	2428*
		2430*	2434	2444	2450	2455	2461*	2462	2466*	2468	2482*	2483*	2489*	2490
		2491*	2492	2493*	2495*	2497	2498	2509*	2510	2516	2528*	2531	2544	2550
		2556*	2560	2567*	2568	2571	2578	2586	2590*	2591*	2592*	2599*	2600	2601
		2604	2657*	2661*	2667	2673*	2675	2682	2685	2688*	2690	2695	2701	2765*
		2766	2823*	2824	2867*	2868	2869*	2870	2949*	2950*	2983*	2992*	3038*	3050*
		3052*	3054*	3056	3058*	3060*	3062*	3064*	3066*	3068*	3076	3107*	3118*	3120*
		3127*	3131*	3136*	3151*	3154*	3162*	3165*	3172*	3176*	3183*	3184*	3187	3188
		3198*	3209*	3224	3225	3243*	3244*	3245	3367*	3368*	3451*	3461*	3464*	3466*
		3468*	3470*	3472*	3478	3511*	3512*	3515*	3521	3523	3592*	3593	3596	3600
		3602	3606	3610*	3636*	3676*	3677	3679	3706*	3707*	3710	3719	3756*	3757*
		3852*	3854*	3856	3857	3867*	3868*	3869*	3898*	3901				
R3	=%000003	359#	580*	590	633*	636*	640*	645*	764*	973*	974*	975	981	1008*
		1009*	1011*	1034*	1038	1041*	1060*	1061	1077	1209*	1210*	1211	1213*	1216*
		1241*	1247*	1254*	1261*	1274*	1306*	1313*	1328*	1334*	1340*	1344*	1346*	1349
		1573*	1574*	1575*	1576	1577	1578*	1579	1582	1587*	1600*	1614*	1626*	1644*
		2148*	2155	2163*	2164*	2166	2198*	2201*	2218*	2219*	2223*	2226	2227*	2230
		2253*	2254	2255	2261	2281	2296*	2297*	2299*	2302	2310	2319	2328	2334*
		2335*	2337	2342	2348	2355	2362*	2364*	2365	2367	2374	2380	2385	2392*
		2393	2394	2439*	2441*	2442	2480*	2481*	2482	2492*	2494	2496*	2500	2501
		2503*	2504	2506*	2507	2518	2519*	2520	2612*	2613	2614*	2615*	2616	2622
		2626	2633	2640	2660*	2662*	2717*	2718	2719	2720*	2721*	2722*	2723	2732*
		2733*	2739	2740*	2741*	2745*	2746*	2747	2749	2753	2755	2825*	2826*	2827
		2870*	2871	2872*	2984*	2993*	3039*	3042	3043*	3044*	3068	3070	3072	3074*
		3108*	3111	3112*	3113	3114*	3115	3116	3123	3143*	3146	3147*	3148*	3149
		3153	3157	3220*	3221*	3222	3231*	3234	3236	3428*	3429*	3430	3460	3521*
		3524	3533	3535*	3601*	3602	3607*	3629*	3637	3640	3644*	3645*	3647	3649
		3654	3660*	3661*	3662	3664	3855*	3856*	3857*					
R4	=%000004	370#	578*	579*	585*	587	632*	765*	975*	976*	977	980	1015*	1016*
		1018	1038*	1043*	1061*	1062	1078	1211*	1212*	1218*	1223*	1229*	1235*	1268*
		1276*	1282*	1283*	1289*	1294*	1301*	1319	1325*	1357*	1358*	1359	1376*	1383*
		1390*	1400*	1413*	1420*	1436*	1439*	1446	1449	1579*	1594*	1607*	1621*	1631*
		1638*	1645*	2208*	2209*	2210*	2211*	2212	2216*	2217*	2218	2226*	2229*	2230*
		2254*	2255*	2256	2260*	2261*	2262	2266*	2267	2271	2275*	2276*	2277	2281*
		2282*	2283	2293*	2294	2295*	2296	2300*	2302*	2310	2319	2327*	2328*	2337*
		2340*	2342*	2348*	2355	2365*	2366*	2367*	2374*	2380*	2385*	2393*	2394*	2400*
		2401	2478*	2479*	2480	2490*	2500*	2501	2504	2510	2513	2515*	2516*	2518*
		2526	2616*	2617*	2618*	2622*	2626	2633	2640*	2658*	2663*	2681*	2723*	2725*
		2726*	2727*	2730*	2731*	2732	2733	2740	2741	2745	2746	2747	2749	2753
		2755	2824*	2825	2985*	2994*	3128*	3129	3133	3134	3135	3139	3199*	3200*
		3201	3205*	3208	3211	3214*	3222	3229*	3230	3236*	3245	3247	3248*	3523*
		3533	3581*	3583	3599*	3600*	3601	3606*	3607	3612*	3621*	3622	3684	
R5	=%000005	371#	577*	579	595	727*	736*	737*	738*	746	747*	750*	751*	752*
		753*	766*	784*	789*	977*	978*	979	1022*	1023*	1025*	1027	1062*	1064
		1079	1359*	1360*	1364	1370*	1395*	1405*	1407*	1426*	1430*	1449	1457*	1458*
		1459	1466*	1475*	1481*	1497*	1512*	1525*	1537	1554*	1564	1667*	1679*	1691*
		1710*	1721*	2177*	2179	2184*	2186	2410*	2411	2412*	2413	2419*	2424*	2431*
		2432*	2434	2442*	2444	2450*	2455*	2462	2466	2475*	2476	2477*	2478	2494*
		2497*	2498	2507	2512*	2513	2520*	2526*	2527*	2531*	2538	2544*	2550*	2557*
		2560*	2568*	2569*	2571	2578*	2584*	2585*	2586	2600*	2601	2654*	2655	2656*
		2657	2667*	2674*	2675*	2682	2685	2690*	2695*	2700*	2701	2718*	2736	2742

SUB2A	010632	2593	2594	2596#										
SUB3	011046	2668	2669	2671#										
SUB3A	011070	2676	2677	2679#										
SUB6	011504	2795	2797#											
SUB7	012102	2892	2893	2894	2896#									
SWA80	002742	960	961	962	964#									
SWA81	004150	1314	1315	1317#										
SWA82	004354	1391	1393#											
SWA84	004774	1549	1551#											
SWA86	006242	1909	1911#											
SWA87	006554	2013	2015#											
SWR =	177570	420#	555	560	562	564	574	652	694	698	3838	3850	3894	
SW06 =	000100	440#	3838											
T =	000020	388#												
TBITVE =	000014	396#	3580	3595*	3598*	3613*								
TEMP1	016532	3834*	3836*	3882#										
TICKS	000776	536#	3828*											
TKB =	177562	416#	730											
TKS =	177560	415#	728											
TPB =	177566	418#	625*	734*	741*	3771*	3785*							
TPS =	177564	417#	623	3766	3772	3774*	3777*	3813	3848	3859*	3885	3887		
TPVEC =	000064	403#	3768*	3769*	3811	3815*	3823*							
TRAPVE =	000034	402#	798*	799*	3336	3337*	3338*	3340*	3360*	3361*				
TRAP1	014126	3337	3350#											
TRAP1C	014152	3349	3360#											
TRTVEC =	000014	397#												
TSTB1	004162	1320	1321	1323#										
TSTB2	004752	1538	1540#											
TSTB2A	004762	1543	1545#											
TSTB6	005770	1823	1824	1825	1827#									
TST0	002500	868	869	870	871	873#								
TST1	003620	1187	1188	1189	1191#									
TST2	004302	1365	1366	1368#										
TST6	006272	1920	1921	1923#										
TTYCHK	015750	3764#												
UBM6	006276	1817	1819*	1822	1830	1834*	1841*	1843*	1849*	1857*	1864*	1870*	1877*	1883*
		1889*	1895*	1901*	1908*	1913*	1919	1926#						
UBPEAK =	177770	413#	562*											
UB7	011770	2854	2865#											
UM6	005464	1735#	1739*	1746*	1752*	1759*	1767*	1774*	1781*	1788*	1790*	1796*	1802*	1808*
UM7	006346	1948#												
UM7	006352	1946	1951#											
V =	000002	385#												
Z =	000004	386#												
SCRLF	001760	719#	745	3783										
SFILLS	001002	539#	616											
SFORM0	001330	630#	664	671	679	687	3853							
SRESTR	000254	489#	648	693										
SSAVR	000214	471#	631	661										
.	= 017406	460#	462#	464#	480#	499#	520#	535#	537#	593	624	699	721#	774
		806	818	826	834	841	849	862	872	882	890	899	908	917
		926	935	941	948	955	963	985	995	1005	1012	1019	1028	1044
		1087	1097	1104	1113	1120	1126	1133	1140	1148	1155	1163	1168	1176
		1182	1190	1197	1202	1207	1220	1226	1232	1238	1244	1250	1257	1264
		1271	1279	1286	1290	1297	1302	1309	1316	1322	1326	1331	1337	1341
		1347	1350	1355	1367	1372	1379	1386	1392	1397	1401	1409	1416	1423

1427	1433	1437	1443	1447	1450	1455	1469	1473	1477	1485	1493	1500
1508	1516	1522	1528	1534	1539	1544	1550	1558	1562	1565	1570	1583
1590	1596	1603	1611	1617	1623	1628	1634	1641	1647	1652	1671	1675
1682	1687	1695	1701	1706	1713	1717	1723	1729	1743	1749	1755	1763
1771	1777	1785	1793	1799	1805	1809	1812	1826	1831	1837	1846	1853
1860	1867	1873	1880	1886	1892	1898	1905	1910	1916	1922	1925	1941
1964	1971	1979	1986	1993	2001	2008	2014	2021	2025	2032	2037	2044
2061	2069	2075	2082	2088	2094	2100	2107	2113	2120	2127	2136	2144
2150	2159	2167	2187	2195	2205	2213	2221	2224	2234	2239	2248	2257
2263	2268	2272	2278	2284	2290	2305	2314	2322	2331	2338	2345	2351
2359	2370	2377	2382	2389	2397	2402	2407	2425	2429	2436	2447	2452
2458	2463	2469	2472	2521	2534	2541	2546	2553	2564	2574	2581	2587
2595	2602	2609	2619	2623	2629	2636	2641	2670	2678	2683	2686	2692
2697	2702	2734	2737	2743	2750	2756	2761	2774	2779	2783	2790	2796
2803	2809	2813	2834	2838	2841	2846	2850	2878	2881	2882	2888	2895
2903	2907	2911	2915	2945	2961	2999	3004	3010	3014	3024	3034	3040
3076	3078*	3097	3099	3104	3109	3117	3119	3124	3140	3144	3158	3166
3177	3260	3275	3282	3320	3321	3328	3345	3347	3350	3354	3409	3417
3517	3584	3603	3650	3657	3665	3671	3682	3699	3714	3723	3767	3773
3814	3844	3849	3888	3949#								
512	602#	667	672	674	676	680	688	690	696	744	782	787
3398	3782	3896										

.PRINT 001232

ADC	886	1109	1167	1407	1631	1802	2029	2928	3695								
ADCB	1216	1276	1472	1475	1704	1843	1857	2055	2125	3066	3300	3324					
ADD	585	604	1016	1818	1929	2164	2230	2243	2302	2385	2394	2400	2531	2674	2690		
	2787	2805	2899	2926	2931	2938	2950	2992	2993	2994	2995	3046	3052	3082	3094		
	3114	3148	3170	3184	3200	3208	3211	3221	3244	3268	3291	3338	3352	3368	3417		
	3420	3425	3459	3467	3502	3512	3515	3520	3531	3577	3617	3635	3675	3686	3757		
ASL	635	639	750	751	752	930	972	974	976	978	1023	1025	1071	1172	1436		
	1439	1600	1790	1976	2165	2175	2770	2771	2781	2793	2806	2880	3054	3388	3464		
	3466	3468	3470	3472	3610	3693	3777	3868									
ASLB	1241	1328	1512	1698	1849	2097	2428	3277									
ASR	737	738	944	1009	1037	1129	1413	1594	1752	1788	1990	2898	3774				
ASRB	1274	1282	1283	1497	1505	1679	1889	2104	2509	2519	3319						
BCC	839	878	887	904	923	931	945	953	1010	1017	1026	1103	1117	1130	1145		
	1173	1180	1195	1201	1230	1236	1242	1255	1262	1269	1284	1307	1335	1371	1377		
	1396	1414	1421	1427	1440	1467	1482	1513	1520	1532	1555	1601	1608	1615	1622		
	1627	1639	1670	1680	1686	1692	1699	1728	1747	1760	1768	1791	1797	1845	1850		
	1871	1878	1890	1896	1902	1914	1969	1977	1991	2006	2019	2042	2058	2073	2080		
	2086	2092	2098	2105	2118	2134	2180	2193	2202	2220	2231	2303	2320	2343	2349		
	2356	2368	2386	2420	2445	2451	2456	2532	2539	2545	2561	2572	2579	2634	2676		
	2696	2772	2788	2877	2892	2900	3022	3032	3101	3194	3239	3278	3315	3325	3611		
	3711																
BCS	528	811	858	868	895	913	939	960	993	1110	1137	1152	1160	1187	1224		
	1248	1278	1295	1314	1320	1329	1365	1391	1431	1476	1490	1498	1506	1526	1538		
	1549	1582	1632	1646	1705	1722	1740	1753	1775	1782	1803	1823	1835	1858	1865		
	1984	1909	1920	1984	2013	2030	2066	2111	2142	2149	2156	2311	2329	2395	2551		
	2593	2627	2668	2734	2795	2838	2886	3345	3633	3720							
BEC	553	556	561	593	685	695	813	880	897	906	915	925	985	995	1005		
	1012	1019	1028	1044	1139	1147	1154	1162	1202	1232	1244	1250	1271	1279	1290		
	1326	1341	1347	1350	1367	1385	1401	1442	1447	1450	1473	1484	1492	1515	1528		
	1539	1543	1557	1562	1565	1583	1610	1641	1647	1675	1723	1729	1742	1762	1770		
	1784	1812	1831	1837	1898	1904	1916	1922	1964	2044	2060	2068	2100	2113	2127		
	2144	2150	2167	2213	2224	2233	2239	2248	2257	2263	2268	2272	2278	2284	2305		
	2313	2322	2331	2358	2381	2402	2422	2425	2447	2457	2463	2469	2499	2502	2511		
	2521	2534	2553	2563	2574	2587	2602	2619	2636	2641	2670	2683	2702	2737	2743		
	2756	2774	2779	2783	2796	2803	2809	2813	2834	2841	2846	2850	2882	2894	2902		
	2907	2915	2999	3004	3010	3014	3024	3034	3124	3140	3158	3166	3177	3202	3226		
	3260	3282	3296	3328	3347	3354	3437	3490	3494	3517	3522	3534	3584	3594	3597		
	3603	3608	3635	3650	3657	3665	3671	3682	3699	3713	3722	3786	3840	3874	3899		
BGE	824	834	890	926	941	963	1097										
BGT	825	841	847	899	917												
BHI	588	818	840	848													
BIC	731	749	2201	2210	2348	2560	2584	2695	2777	2906	2944	3060	3606	3869			
BICB	1068	2255	2455	2466	2516	2520	2640	2745	2746	2837	2844	3013	3030	3661			
BIS	554	753	980	981	982	983	2192	2219	2342	2442	2544	2578	2673	2769	2832		
	2910	3083	3443	3445	3474	3478	3591	3600	3798	3859							
BISB	1063	2261	2281	2450	2497	2500	2622	2740	2741	3000	3020	3064					
BIT	552	555	560	564	574	652	694	1048	2179	2185	2319	2571	2682	2778	2808		
	2925	3357	3593	3596	3766	3838	3848	3873	3885								
BITB	2271	2444	2498	2501	2510	2633	2747	2749	2833	2849	2930	2933	2941	2998	3076		
	3516	3668	3808														
BLE	816	833	849	862	908	940	955	1104	3514								
BLOS	817	872	935	1095	1124	1711	1999	2375									
BLT	619	815	826	832	882	948	3397	3436									
BMI	814	861	871	907	934	1140	1148	1155	1168	1182	1217	1226	1238	1297	1309		
	1322	1331	1372	1386	1392	1397	1409	1416	1423	1443	1469	1485	1500	1516	1534		
	1544	1550	1590	1596	1603	1617	1634	1671	1682	1701	1706	1717	1749	1755	1771		

	1785	1793	1805	1809	1842	1860	1867	1873	1886	1892	1905	1910	1971	1979	1986
	1993	2008	2014	2021	2025	2037	2061	2075	2088	2094	2107	2120	2176	2187	2221
	2314	2345	2351	2370	2382	2397	2436	2452	2541	2546	2564	2581	2595	2623	2678
BNE	2692	2697	2878	2888	2895	2911	3256	3312	3350	3714					
	511	565	568	575	582	591	596	607	614	642	647	653	733	743	846
	860	870	889	933	947	962	1003	1040	1042	1049	1070	1072	1113	1119	1132
	1175	1190	1197	1257	1286	1477	1694	1825	1852	2136	2158	2182	2195	2204	2246
	2388	2467	2505	2508	2514	2517	2605	2748	2750	2754	2816	3103	3121	3137	3155
	3189	3191	3196	3216	3223	3241	3246	3280	3326	3378	3492	3648	3655	3663	3669
BPL	3678	3680	3767	3781	3809	3837	3849	3865	3886	3892					
	624	699	729	823	881	898	916	1096	1112	1120	1126	1133	1163	1176	1189
	1264	1316	1337	1379	1433	1493	1508	1522	1558	1611	1628	1687	1695	1713	1743
	1763	1777	1799	1826	1853	1880	2001	2032	2069	2082	2159	2205	2234	2338	2359
	2377	2389	2423	2458	2629	2790	2903	3104	3197	3242	3303	3307	3317	3590	3723
	3773	3814	3851	3888	3895										
BR	547	571	621	638	739	754	774	780	1073	1087	1207	1219	1277	1355	1455
	1570	1652	1844	1925	1946	2174	2290	2407	2472	2609	2646	2707	2761	2854	2945
	2965	3040	3080	3097	3099	3109	3117	3119	3130	3132	3144	3150	3152	3161	3163
	3171	3173	3192	3212	3217	3232	3233	3238	3252	3253	3273	3321	3349	3439	3441
	3446	3456	3475	3529	3536	3776	3784	3822							
BVC	831	896	905	914	924	932	946	1024	1111	1118	1131	1138	1161	1174	1225
	1231	1243	1263	1275	1285	1296	1330	1378	1384	1432	1499	1521	1589	1595	1609
	1623	1633	1640	1693	1754	1776	1783	1798	1804	1851	1859	1866	1885	1915	2036
	2043	2081	2087	2099	2232	2369	2387	2396	2429	2435	2628	2677	2691	2789	2881
	2993	2901	3102	3240	3275	3279	3298	3302	3306	3311	3316	3697			
BVS	812	859	869	879	888	954	961	1125	1146	1153	1181	1188	1196	1220	1237
	1249	1256	1270	1302	1308	1315	1321	1336	1366	1408	1415	1422	1437	1441	1468
	1483	1491	1507	1514	1527	1533	1556	1602	1616	1681	1700	1712	1741	1748	1761
	1769	1792	1824	1836	1846	1872	1879	1891	1897	1903	1921	1970	1978	1985	1992
	2000	2007	2020	2031	2059	2067	2074	2093	2106	2112	2119	2135	2143	2157	2181
	2194	2203	2304	2312	2321	2330	2344	2350	2357	2376	2421	2446	2533	2540	2552
	2562	2573	2580	2594	2635	2669	2686	2887	3195	3320	3712	3721	3799		
CCC	810	875	911	990	1035	1100	1107	1382	1496	1581	1593	1637	1829	1982	2040
	2153	2185	2433	2529	2689	2786	3304	3630							
CLC	735	1260	1267	1375	1465	1690	1975	1996	2078	2103	2326	3709			
CLN	1185	1599	2011	2309	2559	3098	3582	3709							
CLR	508	530	633	645	727	760	761	762	763	764	765	766	771	772	793
	857	968	999	1032	1091	1213	1360	1400	1462	1575	1660	1734	1819	1954	2052
	2172	2190	2260	2295	2297	2362	2412	2414	2439	2590	2662	2663	2767	2921	2922
	2983	2989	2990	3107	3127	3198	3206	3271	3285	3331	3363	3387	3393	3394	3415
	3444	3462	3463	3518	3574	3575	3595	3599	3612	3645	3764	3831	3842	3843	3870
	3889														
CLRB	1325	2334	2477	2479	2481	2483	3017	3819	3825						
CLRD	3074														
CLRF	3058														
CLV	902	921	1222	1300	1780	1807	1856	3186	3308	3313	3318				
CLZ	856	866	1363	1821	1962	2147	2200	2318	2326	2549	2639	2666	3605		
CMP	587	590	592	595	984	1069	1349	1446	1449	1561	1564	1737	1811	2155	2166
	2212	2237	2238	2245	2247	2256	2262	2267	2277	2310	2355	2538	2586	2604	2653
	2655	2685	2701	2716	2736	2739	2742	2773	2782	2812	2815	2866	2868	2871	2885
	2914	2929	2932	2940	2942	2974	3023	3123	3133	3138	3139	3157	3190	3201	3222
	3225	3234	3245	3258	3259	3434	3488	3493	3533	3602	3647	3649	3656	3662	3664
	3670	3681	3698	3710	3864										
CMPB	613	732	742	2434	2462	2504	2507	2513	2601	2626	2753	2755	2840	2845	2924
	3009	3353	3356	3396	3435	3513	3677	3679	3719	3780					
COM	877	1004	1033	1179	1370	1621	1746	2018	2209	2211	2217	2227	2299	2327	2366

	2392	2557	2569	2585	2591	2688	2700	2807	2811	3118	3131	3151	3162	3172	3218
COMB	3660														
DEC	1268	1334	1466	1669	1870	2072	2126	2275	2416	2461	3299				
DECB	566	641	646	903	1002	1039	1102	1200	1430	1607	1796	1983	3836		
EMT	618	1254	1289	1301	1306	1519	1727	1901	2091	2431	2515	3007	3310		
HALT	443	3295													
INC	460	481	502	700	3400										
	510	912	994	1011	1041	1043	1056	1144	1212	1383	1461	1626	1644	1664	1781
	2035	2050	2173	2223	2300	2415	2721	2726	2826	2977	2981	3008	3044	3120	3136
INCB	3154	3165	3176	3215	3535	3634	3707	3787	3816	3828	3862	3891			
	1218	1223	1344	1503	1554	1685	1841	1895	2085	2123	2124	2282	2424	2503	3003
	3314	3465	3653												
IOT	3272	3335	3469												
JMP	514	3096	3116	3135	3153	3164	3174	3866	3905						
JSR	512	612	620	631	648	661	664	667	671	672	674	676	679	680	687
	688	690	693	696	744	782	784	787	789	3187	3214	3236	3255	3398	3471
MFPD	3782	3841	3853	3896	3901										
MOV	3056	3072													
	465	466	467	471	472	473	474	475	476	477	478	489	490	491	492
	493	494	495	496	501	505	507	523	529	531	557	558	569	570	576
	577	578	580	583	584	589	597	598	602	603	609	616	632	634	637
	643	654	656	658	660	662	663	665	666	669	678	682	726	767	770
	776	779	786	791	794	795	796	797	798	799	902	804	805	807	808
	971	973	975	977	979	998	1008	1022	1034	1038	1050	1051	1053	1057	1058
	1059	1060	1061	1062	1064	1065	1067	1074	1075	1076	1077	1078	1079	1082	1083
	1089	1209	1211	1357	1359	1457	1459	1460	1573	1576	1578	1579	1656	1659	1663
	1665	1666	1667	1736	1817	1928	1930	1937	1939	1940	1942	1943	1951	1955	1956
	1959	1960	2053	2133	2139	2148	2162	2163	2177	2184	2198	2208	2216	2218	2226
	2236	2242	2244	2253	2254	2293	2296	2337	2365	2380	2393	2410	2413	2475	2478
	2480	2482	2526	2527	2528	2567	2568	2599	2600	2612	2616	2651	2652	2654	2656
	2657	2658	2660	2661	2715	2717	2718	2720	2722	2723	2725	2727	2730	2765	2766
	2768	2776	2800	2801	2823	2824	2825	2827	2829	2831	2865	2867	2869	2870	2872
	2875	2876	2923	2949	2951	2957	2959	2960	2962	2963	2973	2975	2976	2979	2980
	2984	2985	2986	2991	3038	3039	3043	3045	3047	3086	3087	3088	3093	3108	3112
	3113	3115	3128	3129	3143	3147	3149	3169	3182	3183	3199	3205	3207	3209	3210
	3220	3229	3230	3231	3243	3254	3267	3270	3284	3289	3290	3293	3330	3336	3337
	3340	3360	3361	3362	3367	3369	3373	3379	3380	3389	3405	3407	3408	3410	3411
	3414	3416	3418	3419	3421	3422	3423	3424	3428	3429	3430	3438	3440	3442	3450
	3451	3452	3460	3461	3483	3487	3498	3499	3501	3504	3505	3509	3510	3511	3519
	3521	3523	3524	3530	3568	3573	3576	3580	3581	3587	3592	3598	3601	3607	3613
	3614	3616	3621	3622	3624	3629	3636	3637	3640	3643	3644	3674	3676	3684	3685
	3691	3693	3705	3706	3756	3758	3765	3768	3769	3770	3794	3795	3796	3811	3815
	3816	3823	3824	3826	3834	3844	3846	3852	3854	3855	3856	3857	3858	3860	3861
	3867	3871	3872	3875	3876	3877	3898								
MOVB	562	606	625	644	730	734	741	991	1015	2266	2276	2419	2490	2492	2494
	2519	2614	2615	2617	2618	2731	2732	2733	2799	2830	2997	3028	3342	3586	3623
	3631	3646	3771	3785	3810	3863									
NEG	922	958	1194	1395	1614	1759	2005	2556	2681						
NEGB	1247	1531	1726	1877	2110	3078	3305								
NOP	532	551	1931	2934	2939	2952	3370	3730	3731	3732	3733	3734	3735	3736	3737
	3738	3739	3740	3741	3742	3743	3744	3745	3746	3747	3748	3749	3750	3751	3752
	3753	3759	3833	3902	3903	3904									
RESET	3835	3893	3900												
ROL	636	640	938	970	992	1116	1123	1420	1638	1739	2041				
ROLB	1229	1489	1710	1883	1913	2117	2489	2491	2493	2495	2496	2506	3021	3274	
ROR	736	894	1001	1036	1094	1136	1376	1405	1587	1767	1998	2229	2364	2441	3632

RJRB	1261	1294	1481	1691	1864	2079	2430	2432	2512	3031	3301				
RTI	523	533	563	701	757	3084	3276	3323	3358	3379	3426	3479	3484	3503	3532
RTS	3578	3598	3618	3626	3687	3778	3789	3794	3829	3845	3878				
SBC	485	610	626	649	747	3193	3219	3248	3263	3481					
SBCB	952	1151	1159	1426	1645	1774	1968	3383	3386						
SCC	1235	1346	1525	1721	1834	2057									
	855	865	901	920	951	969	1253	1312	1362	1464	1488	1504	1553	1606	1709
	1738	1758	1766	1820	1961	1967	1974	2028	2064	2132	2178	2191	2199	2308	2317
SEC	2325	2373	2418	2548	2558	2665	3049	3095	3185	3235	3708	3718			
	524	838	885	893	929	1000	1093	1101	1108	1143	1158	1166	1171	1215	1293
	1305	1345	1389	1404	1406	1480	1547	1586	1720	1789	1840	1863	2056	2228	2341
SEN	2363	2440	2443	2487	2632	2694	2729	2828	2874	3269	3294	3309	3341	3381	3384
SEV	822	2154	2336	2443	3262	3339									
SEZ	830	876	1193	1412	1419	1480	1511	1620	1678	1876	1989	1997	2004	2017	2056
SUB	2116	2140	2154	2301	2341	2354	2443	2530	2537	2570	2577	3292	3797		
	845	876	1511	3294	3343										
	559	579	657	686	775	778	806	1090	1210	1358	1458	1574	1941	2141	2328
	2367	2374	2550	2592	2667	2675	2794	2802	2891	2937	2943	2961	3050	3062	3068
	3409	3694													
SWAB	959	1313	1340	1390	1548	1716	1808	1908	2012	2024	2335	2340	3027		
TRAP	442	3344	3365												
TST	525	581	608	670	683	684	698	746	773	803	867	1018	1027	1186	1364
	1577	1582	1657	1658	1661	1662	1919	1938	1952	1953	1957	1958	1963	2049	2051
	2283	2294	2401	2411	2476	2613	2659	2719	2724	2958	2978	3033	3042	3070	3111
	3122	3134	3146	3156	3188	3224	3247	3281	3327	3377	3382	3406	3433	3583	3589
	3354														
TSTB	623	728	1319	1537	1542	1674	1822	1830	2468	2927	3385	3491	3772	3813	3850
	3887	3894													
WAIT	3775														
.ABS	361														
.ASCII	710	711													
.ASCIZ	516	708	712	713	714	715	719	720	3790	3929	3932	3935	3945		
.END	3953														
.EVEN	721	3791													
.LIST	7	359	360	460											
.MACR	446	447	448	449	450	451	452	453	454	456	457	458	459		
.MACRO	445	454													
.MLIST	7	359	460												
.REM	8														
.REPT	460	3730													
.TITLE	362	758													
.WORD	509	536	539	545	546	655	659	777	785	790	1052	1054	1066	1088	1208
	1356	1456	1571	1572	1653	1654	1655	1735	1926	1947	1948	1949	2291	2292	2408
	2409	2473	2474	2610	2611	2647	2648	2649	2650	2708	2709	2710	2711	2712	2713
	2762	2763	2855	2856	2860	2861	2862	2863	2966	2967	2968	2969	2970	2971	3041
	3110	3145	3175	3213	3390	3483	3500	3525	3615	3692	3788	3812	3817	3882	3890
	3926	3927	3950	3951	3952										

ERRORS DETECTED: 0 HARD 2 SOFT
DEFAULT GLOBALS GENERATED: 0

*.DDQAAA.SEQ/SOL/CRF/PAGNUM=DDQAAA
RUN-TIME: 12 24 6 SECONDS

C08

DDQAA-A BASIC 11 FAMILY INSTRUCTION EXER. MACY11 27(732) 22-SEP-76 14:39 PAGE 96
DDQAAA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

RUN-TIME RATIO: 221.43=5.1
CORE USED: 11K (21 PAGES)

