

KD11-K

BASIC LOGIC TESTS MD-11-DQKDA-A

EP-DQKDA-A-DL-A
COPYRIGHT © 1977
FICHE 1 OF 2

APR 1977
digital
MADE IN USA

The microfiche card contains a grid of 100 small frames, arranged in 10 rows and 10 columns. Each frame contains technical data, likely logic test results, organized in columns and rows. The frames are arranged in a 10x10 grid. The text within the frames is too small to read but appears to be structured data.

The image displays a grid of 120 small diagrams, arranged in 10 rows and 12 columns. Each diagram represents a different logic test or waveform pattern. The diagrams are densely packed and cover most of the page area. The content includes various logic symbols, truth tables, and timing diagrams showing signal transitions over time. The diagrams are arranged in a regular grid pattern, with each cell containing a unique test pattern or logic diagram. The diagrams are small and detailed, typical of technical documentation for digital logic testing.

11

B01

EOF1DQFPDRSBQ411
.REM %

00010000

DOCUMENT LISTING

DDP10 411

HDR1DQKDAASEQ

00010000

770323

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DQKDA-A-D
PRODUCT NAME: KD11-K BASIC LOGIC TESTS
DATE: 01-FEB-77
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: JACK RICH

COPYRIGHT (C) 1977
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

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

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

TABLE OF CONTENTS

- 1.0 GENERAL PROGRAM INFORMATION
 - 1.1 PROGRAM PURPOSE
 - 1.2 SYSTEM REQUIREMENTS
 - 1.3 RELATED DOCUMENTS AND STANDARDS
 - 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 - 1.5 FAILURE ASSUMPTIONS
- 2.0 OPERATING INSTRUCTIONS
 - 2.1 LOADING AND STARTING PROCEDURES
 - 2.2 SPECIAL ENVIRONMENTS
 - 2.3 PROGRAM OPTIONS
 - 2.4 EXECUTION TIMES
- 3.0 ERROR INFORMATION
 - 3.1 ERROR REPORTING PROCEDURES
 - 3.2 ERROR HALTS
- 4.0 PERFORMANCE AND PROGRESS REPORTS
 - 4.1 PERFORMANCE REPORTS
 - 4.2 PROGRESS REPORTS
 - 4.3 MAINTENANCE BREAKPOINT FEATURE
- 5.0 MAINTENANCE PROCEDURES
 - 5.1 THE KD11-K PROCESSOR
 - 5.2 CONDITION CODE SCOPE SYNC FEATURE

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

"DQKDA" IS A DIAGNOSTIC PROGRAM DESIGNED TO DETECT, REPORT, AND IDENTIFY LOGIC FAULTS IN THE KD11-K CENTRAL PROCESSING UNIT OF THE PDP11/6X SYSTEM. IT CONSISTS OF 504(10) INDIVIDUAL TESTS CAREFULLY DESIGNED AND SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY LOGIC FAULTS AT A MINIMUM HARDWARE/SOFTWARE LEVEL. THESE TESTS ARE PARTITIONED INTO FOUR MAJOR SECTIONS AS DESCRIBED BELOW:

A. BASIC CPU TESTS (BCPT)

THIS IS THE BASIC CPU TEST TO VERIFY THE "HARDCORE". ANY FAULT DETECTED IN THIS SECTION CAUSES THE PROGRAM TO HALT WITH THE PC+2 OF THE HALT INSTRUCTION DISPLAYED ON THE CONSOLE.

B. BASIC INSTRUCTION TESTS (BIT)

THIS SECTION CONSISTS OF A LOGICALLY SEQUENCED SET OF BASIC INSTRUCTION TESTS DESIGNED TO VERIFY THE INTEGRITY OF THOSE INSTRUCTIONS AND LOGIC OPERATIONS USED BY THE UTILITY ROUTINES THAT PROVIDE ERROR LOGGING AND SCOPE LOOPING FACILITIES FOR THE SUBSEQUENT TWO MAJOR SECTIONS. NO UTILITY IS CALLED UNTIL ITS INSTRUCTION COMPLEMENT HAS BEEN VERIFIED. THIS SCHEME ACCOMPLISHES TWO IMPORTANT MAINTENANCE OBJECTIVES: 1)IT MINIMIZES THE POSSIBILITY OF THE ERROR REPORTING ROUTINES CONVEYING AMBIGUOUS ERROR INFORMATION TO THE USER, AND 2)IT MAXIMIZES THE POSSIBILITY THAT THE ERROR WILL BE DETECTED BY A ROUTINE DESIGNED TO IDENTIFY FAILING OPERATIONS RATHER THAN HAVE THE ERROR MANIFEST ITSELF IN A MORE COMPLEX UTILITY ROUTINE THAT IS NOT STRUCTURED TO DIAGNOSE FAULTS.

ANY FAULT DETECTED IN THIS SECTION CAUSES THE PROGRAM TO HALT WITH THE CONSOLE ADDRESS INDICATING THE PC+2 OF THE HALT INSTRUCTION IN THE FAILING TEST. ADDITIONAL FAULT IDENTIFICATION INFORMATION IS AVAILABLE IN THE PROCESSOR'S GENERAL REGISTERS, PSW, STACK, AND PROGRAM ANNOTATION FOR THE FAILING TEST. A LOCK ON HARD ERROR FEATURE IS EMPLOYED TO PREVENT THE PROGRAM FROM CONTINUING ON ONCE A SOLID ERROR IS DETECTED. DEPRESSING CONTINUE AFTER THE ERROR HALT CAUSES A RETRY OF THE FAILING TEST.

C. COMPREHENSIVE INSTRUCTION TESTS (CIT)

THIS SECTION, COMPRISED OF THE BULK OF THE TESTS, CONSISTS OF A LOGICALLY SEQUENCED AND PARTITIONED SET OF INSTRUCTION TESTS DESIGNED TO TEST AND VERIFY ALL THE BASIC INSTRUCTIONS OF THE KD11-K PROCESSOR. THIS EXCLUDES TESTING THOSE LOGIC FUNCTIONS THAT SUPPORT THE CONSOLE FUNCTIONS (LOAD ADDRESS, DEPOSIT, ETC.). EACH TEST IN THIS SECTION CALLS A "SCOPE LOOP" UTILITY THAT FACILITATES USER CONTROL OF TEST SELECTION AND EXECUTION VIA THE CONSOLE SWITCH REGISTER.

UPON DETECTION OF A LOGIC FAULT, EACH TEST IN THIS SECTION CALLS AN "ERROR SERVICE" ROUTINE THAT LOGS THE ERROR AND REPORTS IT AS HARD COPY ON THE CONSOLE TERMINAL DEVICE. THE ERROR SERVICE ROUTINE ALSO FACILITATES USER CONTROL OF THE PROGRAM

SEQUENCE VIA CONSOLE SWITCH REGISTER OPTIONS. AFTER REPORTING THE ERROR THE PROGRAM CONTINUES ON IN ITS NORMAL SEQUENCE UNLESS MODIFIED BY THE USER ACTIVATING THE "LOCK ON HARD ERROR" SWITCH OPTION. 501

D. COMBINED INSTRUCTION EXERCISER (IEX)

THIS SECTION CONSISTS OF A MORE COMPLEX SET OF INSTRUCTION TESTS DESIGNED TO TEST THE INSTRUCTIONS WHEN USED IN VARIOUS COMBINATIONS MANIPULATING VARIABLE DATA PATTERNS. IT ALSO TESTS THE MED AND ERROR LOGGING FEATURES OF THE CPU. LIKE THE PREVIOUS SECTION, IT CALLS THE "ERROR SERVICE" AND "SCOPE LOOP" UTILITIES TO REPORT ERRORS AND ALLOW USER CONTROL OF TEST EXECUTION.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

1. PDP11/6X CPU WITH OPERATOR'S CONSOLE
2. 16K OF CORE STORAGE - MF11/U OR EQUIVALENT
3. DL11-W ASYNCHRONOUS LINE INTERFACE WITH LINE CLOCK

B. SOFTWARE REQUIREMENTS

1. PDP11 ABSOLUTE LOADER PROGRAM FOR PAPER TAPE SYSTEMS
2. XXDP MONITOR FOR DECTAPE, MAGTAPE, CASSETTE, OR DISK SYSTEMS.

1.3 RELATED DOCUMENTS AND STANDARDS

DQKDAE USES THE STANDARD APT SOFTWARE INTERFACES FOUND IN THE MACY11 SYSMAC PACKAGES.

1.4 DIAGNOSTIC HIERARCHY REQUIREMENTS

DQKDAE WILL NORMALLY BE THE FIRST DIAGNOSTIC TO BE RUN AS PART OF PDP 11/6X CPU CHECKOUT.

1.5 FAILURE ASSUMPTIONS

"DQKDAE" ASSUMES THAT THE STORAGE MEDIUM USED TO STORE THE PROGRAM IS INTACT AND THAT IT CAN BE LOADED INTO CORE.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

A. LOADING PROCEDURES

- 1) STANDARD PDP11 ABSOLUTE LOADER PROCEDURES FOR PAPER TAPE.
- 2) STANDARD XXDP MONITOR LOADING PROCEDURES.
- 3) STANDARD APT OR ACT LOADING

B. MANUAL STARTING PROCEDURES

- FO1
- 1) LOAD SWITCH REG WITH 000000 (NO SWITCH OPTIONS)
 - 2) SET DISPLAY TO 000200
 - 3) DEPRESS LOAD ADDRESS
 - 4) PRESS CNTRL AND START BUTTONS SIMULTANEOUSLY

2.2 SPECIAL ENVIRONMENTS

16K PDP11/6X SERIES SYSTEMS

FOR 16K SYSTEMS USING THE "XXDP" PACKAGE YOU WILL BE UNABLE TO USE THE "UPDATE" PROGRAMS TO LOAD, SAVE, UPDATE ETC. SINCE THE SIZE OF "DQKDA" WILL NOT PERMIT SIMULTANEOUS RESIDENCY OF THE UPDATE PROGRAMS. SUFFICIENT FREE CORE IS AVAILABLE FOR THE "XXDP" MONITOR SO THAT "DQKDA" CAN BE LOADED BY THE MONITOR.

2.3 PROGRAM OPTIONS

A. SWITCH REGISTER OPTIONS

THE FOLLOWING CONSOLE SWITCH REGISTER OPTIONS ARE ACTIVE UPON ENTERING THE COMPREHENSIVE INSTRUCTION TESTS (CIT) SECTION: (SWITCH OPTION IS ACTIVE WHEN SW IS SET TO A "1")

- SW15 HALT ON ERROR. IF ERROR PRINTING IS ENABLED THE HALT OCCURS AFTER THE PRINTOUT. DEPRESSING "CONTINUE" CAUSES THE PROGRAM TO PROCEED ON IN NORMAL SEQUENCE FROM THE POINT OF ERROR.
- SW14 CONTINUOUSLY LOOP ON THE CURRENT TEST
- SW13 INHIBIT NORMAL ERROR PRINTOUTS - THIS DOES NOT INCLUDE POWER FAIL, BUS ERROR, OR RSVD INSTR TRAPS.
- SW12 INHIBIT ALL PRINTOUTS NOT COVERED UNDER SW13. THIS INCLUDES I.D., BUS ERROR, AND RSVD INSTR TRAPS. NOTE THAT IT IS NOT POSSIBLE TO INHIBIT END PASS OR POWER FAIL PRINTOUTS.
- SW11 INHIBIT SUB-TEST ITERATIONS. TEST ITERATIONS ARE AUTOMATICALLY INHIBITED ON THE FIRST PASS.
- SW10 SEARCH FOR AND CONTINUOUSLY LOOP ON THE TEST NUMBER SELECTED BY THE CONTENTS OF SW<08:00>. ONLY USE THIS OPTION FOR TESTS TST176 THRU TST767 SINCE THE "SCOPE" UTILITY IS NOT ACTIVE UNTIL TEST TST176. LOOPING ON TST176 WILL CAUSE A LOOP ON THE ENTIRE "BIT" SECTION (TESTS 0-176).
- SW09 LOCK ON HARD ERROR
- SW<8:0> USED TO SELECT A PARTICULAR TEST FOR LOOPING IF SW10=1. TEST NUMBER MUST BE BETWEEN 176 AND 767.

B. MEMORY LOCATIONS

4. BPTLOC: THERE IS A LOCATION TAGGED "BPTLOC" THAT PROVIDES THE USER THE MECHANISM FOR SETTING SIXTEEN "BREAKPOINT HALTS" THROUGHOUT THE PROGRAM. THIS ENABLES RAPIDLY "HOMING IN" ON THE FAILING TEST IN THOSE CASES WHERE THE FAULT CAUSES A RUNAWAY OR HUNG PROGRAM. REFER TO

PARA. 4.2 FOR A DETAILED DESCRIPTION OF THE USE ^{GO 1}
THIS FEATURE.

2.4 EXECUTION TIMES

ONE COMPLETE ERROR FREE PASS OF DQKDAE WITH NO TEST ITERATIONS SHOULD TAKE LESS THAN 7 SECONDS. A SUCCESSFUL PASS WILL BE INDICATED BY THE FOLLOWING PRINTOUT ON THE CONSOLE DEVICE:

END PASS # 000001 ERROR COUNT = 000000

THIS ERROR COUNT IS NOT CLEARED AT THE BEGINNING OF A NEW PASS. WITH ITERATIONS ENABLED A COMPLETE ERROR FREE PASS SHOULD TAKE LESS THAN 2.5 MINUTES.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

A. ERROR MESSAGE FORMATS

THERE ARE SEVERAL DIFFERENT ERROR FORMATS. EACH IS DESCRIBED BELOW.

1.) ERROR 1 IS OF THE FORM

S/B	DST	WAS	DST	DEST	(IR)	TEST	(PC)	(SP)	(PSW)
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX

WHERE:

S/B DST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS COLUMN CONTAINS WHAT THE RESULT (DEST. OPERAND) SHOULD HAVE BEEN (S/B).

WAS DST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS COLUMN CONTAINS WHAT THE RESULT (DEST. OPERAND) ACTUALLY WAS AFTER THE TEST.

DEST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS COLUMN CONTAINS THE DESTINATION ADDRESS.

(IR) THIS IS A COPY OF THE TEST INSTRUCTION. THIS WILL BE THE FIRST WORD IN THE CASE OF TWO OR THREE WORD INSTRUCTIONS.

TEST INDICATES THE TEST NO. (IN OCTAL) THAT FAILED

(PC) INDICATES THE CONTENTS OF THE PROGRAM COUNTER AT THE TIME OF THE ERROR CALL. THIS IS AN ADDRESS NORMALLY USED TO LOCATE THE ERROR CALL STATEMENT IN THE FAILING TEST.

(SP) INDICATES THE CONTENTS OF THE STACK POINTER (R6) AT THE TIME OF THE ERROR. NOTE THAT THE ERROR CALL WILL PUSH THE STACK TWICE. IN SP TESTS WHERE THE SP MUST BE RESTORED PRIOR TO CALLING THE ERROR ROUTINE, THEN THE ORIGINAL (UNRESTORED) SP IS TYPED, WITHOUT ADDITIONAL PUSHES FROM THE ERROR CALL.

(PSW) INDICATES THE CONTENTS OF THE PROCESSOR STATUS WORD AT THE TIME OF THE ERROR CALL.

XXXXXX IS AN OCTAL NUMBER.

2.) ERROR 2 AND ERROR 4 ARE THE SAME AS FOR ERROR 1 ABOVE EXCEPT THAT IN THIS CASE THE DESTINATION IS A GENERAL REGISTER (WHICH DOES NOT HAVE A UNIBUS ADDRESS). THE OCTAL NUMBER TYPED OUT IN THE "DEST" COLUMN SHOULD BE IGNORED. THE TYPED OUT WOULD LOOK AS FOLLOWS:

```
S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
          IS R3
XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
```

3.) ERROR 5, ERROR 6, AND ERROR 7 ARE IDENTICAL TO ERROR 1 EXCEPT THAT ONLY THE LAST 5, 6, OR 7 COLUMNS (RESPECTIVELY) ARE PRINTED.

4.) ERROR 3 IS USED IN CASES WHERE THE STACK POINTER IS SPECIFICALLY IN ERROR. THE COLUMNS HAVE THE SAME MEANING AS DESCRIBED FOR ERROR 1 EXCEPT:

S/B SP IS WHAT THE STACK POINTER SHOULD HAVE BEEN (S/B)

WAS SP IS WHAT THE STACK POINTER ACTUALLY WAS

5.) OTHER ERRORS TYPE OUT THEIR SPECIFIC ERROR MESSAGE, FOLLOWED BY SELF EXPLANATORY DATA HEADERS, DEPENDING ON THE ERROR. AN EXAMPLE FOLLOWS:

```
BAD DATA READ BY A MED
PC MEDCODE EXPECTD RECEIVD
XXXXXX XXXXXX XXXXXX XXXXXX
```

6.) WHEN THE SCOPE ROUTINE BECOMES ACTIVE, IT CHECKS THAT THE TEST NUMBER (IN RD) IS EXACTLY ONE GREATER THAN THE TEST NUMBER ON THE PREVIOUS SCOPE CALL. IF A MACHINE ERROR CAUSES TESTS TO BE SKIPPED, OR THE PROGRAM TO JUMP BACKWARDS, ERROR 11 WILL REPORT THIS AS FOLLOWS:

```
TESTS SKIPPED
PC EXPCTD ACTUAL (TEST #'S)
XXXXXX XXXXXX XXXXXX
```

EXPCTD THIS IS THE TEST NUMBER THE SCOPE WAS EXPECTING TO BE CALLED FROM.

ACTUAL THIS IS THE TEST NUMBER THAT IT FOUND IN RD

7.) RESERVED INSTRUCTION TRAP ERROR MESSAGE

ANY RESERVED INSTRUCTION TRAP DETECTED AFTER THE BASIC TESTS RESULTS IN THE FOLLOWING PRINTOUT:

TRAPPED TO 10 PC = XXXXXX

WHERE: XXXXXX IS THE VALUE OF THE PROGRAM COUNTER PUSHED ON THE STACK WHEN THE TRAP WAS SPRUNG.

AFTER REPORTING THE ERROR, THE PROGRAM IS RESTARTED FROM THE BEGINNING.

IF A RSVD INSTRUCTION TRAP OCCURS WHILE IN THE PROCESS OF TRYING TO SERVICE A PREVIOUS RSVD INSTRUCTION TRAP

OR A BUS ERROR TRAP THE PROGRAM HALTS. A DESCRIPTION ¹⁰¹
OF THIS HALT IS CONTAINED IN PARA. 3.2.3 BELOW.

IF A RSVD INSTRUCTION TRAP OCCURS PRIOR TO COMPLETION
OF THE BASIC INSTRUCTION TEST SECTION THE PROGRAM WILL
HALT VIA A TRAPCATCHER IN THE VECTOR. A DESCRIPTION OF
THIS HALT IS DESCRIBED IN PARA. 3.2.2 BELOW.

4. BUS ERROR TRAP ERROR MESSAGE

ANY UNEXPECTED BUS ERROR TRAPS (BUS TIMEOUT, ODD
ADDRESS ERROR, ILLEGAL INSTRUCTION, OR STACK OVERFLOW)
RESULTS IN THE FOLLOWING PRINTOUT:

TRAPPED TO 4 PC = XXXXXX

WHERE: XXXXXX IS THE VALUE OF THE PC PUSHED ONTO
THE STACK WHEN THE TRAP WAS SPRUNG.

AFTER REPORTING THE ERROR THE PROGRAM IS RESTARTED
FROM THE BEGINNING.

IF A BUS ERROR TRAP OCCURS WHILE A PREVIOUS BUS ERROR
OR RSVD INSTRUCTION IS STILL PENDING THE PROGRAM WILL
HALT. A DESCRIPTION OF THE HALT INTERPRETATION IS GIVEN
IN PARA. 3.2.3 BELOW.

IF A BUS ERROR OCCURS PRIOR TO THE COMPLETION OF THE
BASIC INSTRUCTION TESTS, THE PROGRAM WILL HALT VIA A
TRAPCATCHER IN THE VECTOR. A DESCRIPTION OF THIS HALT
IS INCLUDED IN PARA. 3.2.2 BELOW.

5. POWER FAIL

IF A POWER FAIL CONDITION IS DETECTED, THE FOLLOWING
MESSAGE IS PRINTED:

POWER

AFTER PRINTING AN ATTEMPT IS MADE TO RESTART THE PROGRAM AT
THE BEGINNING.

3.2 ERROR HALTS

1. BASIC INSTRUCTION TESTS (BIT)

ANY ERROR DETECTED IN THE BASIC TESTS CAUSES THE
PROGRAM TO HALT WITH THE PC+2 OF THE LOCATION CONTAINING
THE HALT INSTRUCTION DISPLAYED.

EXAMINING THE CONTENTS OF THE CPU'S GENERAL REGISTERS,
THE PSW, AND THE STACK WILL PROVIDE ADDITIONAL FAULT
IDENTIFICATION INFORMATION.

DEPRESSING "CONTINUE" AFTER THE HALT WILL CAUSE AN
AUTOMATIC RETRY OF THE FAILING TEST. IF THE ERROR IS
SOLID THE PROGRAM WILL LOCK ON THIS TEST, BUT IF IT
IS INTERMITTENT THE PROGRAM WILL CONTINUE ON IN NORMAL
SEQUENCE ONCE THE TEST IS SUCCESSFULLY EXECUTED.

J01

TO ESTABLISH A TIGHT SCOPE LOOP ON THE FAILING TEST, REPLACE THE "HALT" WITH A 400(8). AND DEPRESS "CONTINUE" THE "400" IS A "BR +2" WHICH FUNCTIONS AS A NOP. THIS IS NECESSARY TO PRESERVE THE INTEGRITY OF THE CONDITION CODE OPERATE INSTRUCTION THAT IS USED AS A SCOPE SYNC. THIS BUILT IN SYNC FEATURE IS DESCRIBED IN PARA. 5.0.

2. TRAPCATCHER HALTS

THE VECTOR AREA (LOC 000 - 776) IS PROGRAM LOADED WITH A STANDARD TRAPCATCHER AS SHOWN BELOW:

V / V+2
V+2/ HALT

AFTER THE BASIC INSTRUCTION TESTS THE FOLLOWING VECTORS ARE SET UP TO POINT TO APPROPRIATE SERVICE ROUTINES:

4/6	BUS ERROR SERVICE
10/12	RSVD INSTRUCTION TRAP SERVICE
20/22	SCOPE LOOP SERVICE
24/26	POWER FAIL SERVICE
30/32	ERROR SERVICE
34/36	PRINT SERVICE

AT THE APPROPRIATE POINTS IN THE COMPREHENSIVE INSTRUCTION TESTS THE LINE CLOCK VECTOR (100/102) AND THE DL11 VECTORS (60/62 - 64/66) ARE SET UP TO CHECK INTERRUPTS FROM THESE DEVICES. ALL OTHER VECTORS REMAIN SET UP TO "CATCH" UNEXPECTED TRAPS OR INTERRUPTS BY HALTING.

WHEN AN UNEXPECTED TRAP OR INTERRUPT NOT SUPPORTED BY AN APPROPRIATE SERVICE ROUTINE OCCURS THE CPU HALTS. WITH THE PC+4 OF THE VECTOR DISPLAYED IN THE CONSOLE. THIS IS USED TO IDENTIFY THE CAUSE OF THE UNEXPECTED TRAP OR INTERRUPT.

THE LAST ENTRY PUSHED ON THE STACK CAN BE EXAMINED TO DETERMINE WHERE THE PROGRAM WAS WHEN THE TRAP OR INTERRUPT WAS SPRUNG. REMEMBER THAT THE "OLD PC" GETS SAVED ON THE STACK WHEN A TRAP OR INTERRUPT OCCURS.

3. CATASTROPHIC ERROR HALTS

THERE ARE TWO HALTS, ONE IN THE BUS ERROR SERVICE ROUTINE AND THE OTHER IN THE RSVD INSTRUCTION TRAP SERVICE ROUTINE THAT HALT THE PROGRAM IF ONE OF THESE ERRORS OCCURS WHILE STILL SERVICING A PREVIOUS BUS ERROR OR RSVD INSTRUCTION TRAP. AFTER THE HALT THE CONSOLE DISPLAYS THE PC+2 OF THE ERROR HALT. THIS IS USED TO IDENTIFY WHICH OF THE TWO TYPES OF ERRORS - RSVD OR BUS ERROR - OCCURRED LAST.

THERE IS A SOFTWARE FLAG TAGGED "CATERR" THAT MAY BE EXAMINED TO OBTAIN THE FOLLOWING INFORMATION:

[CATERR]	= 000002	TWO SUCCESSIVE BUS ERRORS
[CATERR]	= 001000	TWO SUCCESSIVE RSVD INSTR. TRAPS
[CATERR]	= 000401	A COMBINATION OF THE TWO. THE CONTENTS OF THE ADDRESS DISPLAY IDENTIFIES WHICH TYPE OCCURRED LAST.

THE STACK PROVIDES THE FOLLOWING ADDITIONAL INFORMATION: ^{K01}

[SP] / PC OF THE 2ND TRAP
[SP+2] / PSW OF THE 2ND TRAP
[SP+4] / PC OF THE 1ST TRAP
[SP+6] / PSW OF THE 1ST TRAP

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THERE IS ONLY ONE PERFORMANCE REPORT SUPPLIED BY THE PROGRAM AND CONSISTS OF A SIMPLE END OF PASS MESSAGE OF THE FORMAT SHOWN BELOW:

PASCNT = XXXXXX ERRCNT = YYYYYY

WHERE: XXXXXX IS THE TOTAL NUMBER OF COMPLETE PASSES OF THE ENTIRE PROGRAM (OCTAL)

YYYYYY IS THE TOTAL ERROR COUNT IN OCTAL

4.2 PROGRESS REPORTS

THERE ARE TWO PROGRESS REPORTS PRINTED THAT REPORT NORMAL ERROR FREE EXECUTION OF THE PROGRAM.

A. END OF PASS PRINTOUT AS DESCRIBED IN 4.1 ABOVE.

B. PROGRAM IDENTIFICATION MESSAGE AS DESCRIBED BELOW:

MD-11-DQKDAK KD11-K BASIC LOGIC TESTS

THIS MESSAGE GETS PRINTED THE FIRST TIME THE PROGRAM ENTERS THE COMPREHENSIVE INSTRUCTION TEST SECTION UNLESS INHIBITED BY SW12=1. AFTER THE FIRST PASS THIS PRINTOUT IS AUTOMATICALLY INHIBITED UNLESS THE PROGRAM IS RESTARTED AT 200(8).

4.3 MAINTENANCE BREAKPOINT FEATURE

THERE IS A MANUAL PROGRESS REPORT FEATURE THAT ALLOWS THE USER TO STEP THROUGH THE PROGRAM, HALTING AFTER EVERY N'TH TEST WITH PROGRESS INFORMATION DISPLAYED IN THE CONSOLE ADDRESS DISPLAYS. TO ACTIVATE THIS FEATURE THE USER MUST SET THE DESIRED "BREAKPOINT HALT" BITS IN THE MEMORY LOCATION TAGGED "BPTLOC". THIS LOCATION PROVIDES SIXTEEN POSSIBLE HALTS DISPERSED EVENLY THROUGHOUT THE PROGRAM (APPROX. EVERY 20 TESTS). AT EACH CHECKPOINT THE PROGRAM EXAMINES A PARTICULAR BIT IN "BPTLOC" AND HALTS IF THE BIT IS SET TO A "1" OTHERWISE IT CONTINUES IN NORMAL SEQUENCE. AFTER THE HALT DEPRESSING "CONTINUE" WILL CAUSE RESUMPTION OF NORMAL PROGRAM EXECUTION. SETTING LOCATION "BPTLOC" TO ALL 1'S (177777) WILL RESULT IN THE FOLLOWING SIXTEEN HALTS WITH THE INFORMATION SHOWN DISPLAYED IN THE CONSOLE:

[BPTLOC]	ADDRESS DISPLAY
	HALT PC+2

BIT00=1	4302
BIT01=1	6266
BIT02=1	10606

BIT03=1	11736
BIT04=1	14316
BIT05=1	17056
BIT06=1	21502
BIT07=1	24310
BIT08=1	27122
BIT09=1	32116
BIT10=1	34602
BIT11=1	37412
BIT12=1	42064
BIT13=1	46102
BIT14=1	52542
BIT15=1	55366

NOTE: IF THE USER DEPOSITED A 000400(8) IN LOCATION "BPTLOC" ONLY ONE HALT WOULD OCCUR AND AT THAT TIME THE DISPLAY SHOULD CONTAIN 27114.

THIS FEATURE IS USEFUL FOR TRACKING DOWN THE TEST THAT CAUSES A "RUNAWAY" OR "HUNG" PROGRAM.

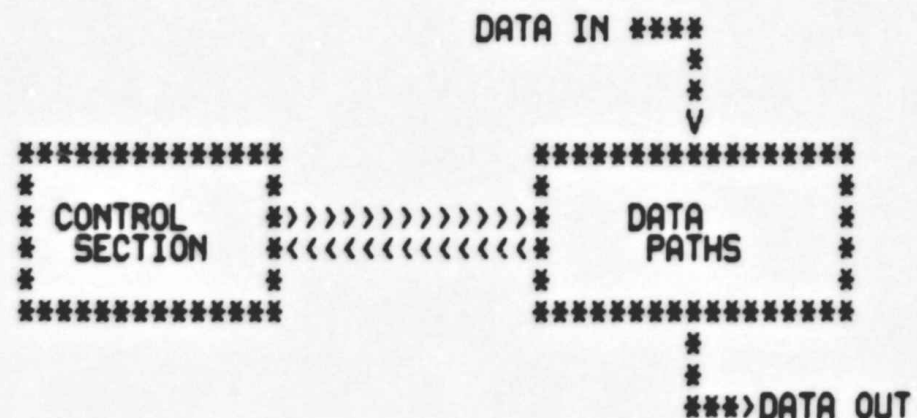
LOCATION "BPTLOC" IS PROGRAM LOADED AS 000000 TO INHIBIT ANY HALTS.

5.0 MAINTENANCE PROCEDURES

5.1 THE KD11-K PROCESSOR

THE PROCEDURES OUTLINED IN THIS SECTION ASSUME THAT "DQKDA" CAN BE LOADED INTO CORE AND STARTED. IF THE FAILURE MODE PREVENTS PROGRAM LOADING OR AFFECTS NORMAL POWER UP AND CONSOLE OPERATIONS, THE TECHNICIAN MUST REVERT TO THE MANUAL DEBUG AND CHECKOUT PROCEDURES.

THE KD11-K CENTRAL PROCESSING UNIT CAN BE VIEWED AS CONSISTING OF TWO MAJOR LOGIC AREAS AS DEPICTED BELOW:



THE DATA PATHS CONSIST OF A LOGICALLY INTERCONNECTED GROUP OF STATIC DATA FACILITIES (REGISTERS, MULTIPLEXORS, ALU'S ETC.) REQUIRED TO TEMPORARILY STORE, MODIFY, AND TRANSFER DATA ITEMS (16 BIT WORDS OR 8 BIT BYTES) ACCORDING TO THE DESIGN SPECIFICATIONS FOR THE PDP11.

THE CONTROL SECTION SUPPLIES PREDEFINED SEQUENCES OF CONTROL SIGNAL SETS TO ACTIVATE THE REQUIRED DATA FACILITIES WITHIN THE DATA PATHS. IN THE KD11-K THESE CONTROL SIGNAL SETS ARE STORED IN A READ ONLY MEMORY (ROM) AND GENERATED BY READING

OUT A UNIQUE SEQUENCE OF ROM WORDS FOR EACH OPERATION TO BE MO1
PERFORMED.

THE SEQUENCE GENERATED BY THE CONTROL SECTION IS VARIABLE
AND DEPENDENT UPON THE INSTRUCTION OR LOGIC OPERATION
BEING EXECUTED. THERE ARE HUNDREDS OF THESE SEQUENCES POSS-
IBLE DEPENDENT UPON OF THE PROGRAM CODING.

"DQKDA" IS DESIGNED TO GENERATE ALL POSSIBLE MICROINSTRUCTION
SEQUENCES AND COMBINATIONS OF DATA AND CONTROL SIGNALS. THE
INDIVIDUAL TESTS ARE LOGICALLY SEQUENCED AND STRUCTURED TO
DETECT AND ISOLATE PARTICULAR MICROPROGRAM SEQUENCES THAT ARE
FAULTY.

5.2 CONDITION CODE SCOPE SYNC FEATURE

FROM THE BIT SECTION TO THE MED TESTS IN THE CIT SECTION, ALL
TEST INSTRUCTIONS ARE PRECEDED BY A CONDITION CODE OPERATE
INSTRUCTION. THE LIBREAK REGISTER IS PROGRAM LOADED TO GENERATE
A SYNC PULSE NEAR THE END OF THIS INSTRUCTION. DURING THE MED
TESTS, THE PULSE IS GENERATED NEAR THE BEGINNING OF THE MED
EXECUTION. THIS PULSE IS GENERATED ON BACKPLANE PIN B03M2 AND
MAY BE USED IN CONJUNCTION WITH THE PROGRAM LOOPING FEATURES TO
PROBE THE KD11-K DURING THE FAILING TEST.

%

13	OPERATIONAL SWITCH SETTINGS
25	BASIC DEFINITIONS
135	TRAP CATCHER
144	STARTING ADDRESS(ES)
147	APT PARAMETER BLOCK
169	ACT11 HOOKS
179	COMMON TAGS
235	APT MAILBOX-ETABLE
262	ERROR POINTER TABLE
576	BT001 "BR" TEST - POSITIVE OFFSET
585	BT002 "BR" TEST - NEGATIVE OFFSET
599	BT003 "BASIC COND. BR" TEST - FLAGS CLEARED
612	BT004 "SCC AND COND. BR'S" TEST - FLAGS SET
627	BT005 "CCC AND COND. BR'S" TEST - FLAGS CLEARED
642	BT006 "CLR XR" TEST - SETS THE "Z" BIT
655	BT007 "TST XR" TEST - USING THE CLR
670	BT010 "COM XR" TEST - SHOULD SET "N" AND "C"
685	BT011 "COM XR AND ADC XR" TEST
701	BT012 "MOV #N,R" TEST WITH N=177777, [R]=000000
716	BT013 "MOV #N,R" TEST WITH N=000000, [R]=177777
733	BT014 "CLR (R)" TEST - [R] = 177776
748	BT015 "CLR (R)+" TEST - [R] = 177776
769	BT016 "COM (R)" TEST - [R] = 177776
786	BT017 "COM (R)+" TEST - [R] = 177776
811	BT020 "MOV RA, RB" TEST - WITH [RA]=177777, [RB]=000000
833	BT021 "MOV RA, RB" TEST WITH [RA]=000000, [RB]=177777
856	BT022 "MOV #N, J#A" TEST WITH N=17, A=177776
872	BT023 "MOV RA, (RB)+" TEST WITH [RA]=17, [RB]=177776
896	BT024 "CMP #N, J#A" TEST WITH N=(A)
911	BT025 "CMP #N, J#A" WITH N > (A)
924	BT026 "CMP #N, J#A" WITH N < (A)
938	BT027 "CMP R, #N" TEST WITH [R]=N
952	BT030 "CMP R, #N" TEST WITH [R] > N
965	BT031 "CMP R, #N" TEST WITH [R] < N
979	BT032 "CMP (RA)+, RB" TEST WITH [SOURCE]=[RB]
1001	BT033 "CMP (RA)+, RB" TEST WITH [SOURCE]>[RB]
1022	BT034 "CMP (RA)+, RB" TEST WITH [SOURCE]<[RB]
1043	BT035 "CMP RA, RB" TEST WITH [RA] = [RB]
1057	BT036 "CMP RA, RB" TEST WITH [RA] < [RB]
1071	BT037 "CMP RA, RB" TEST WITH [RA] > [RB]
1086	BT040 "MOV (RA), RB" TEST WITH [SOURCE]=[RB]=17
1102	BT041 "MOV (RA), RB" TEST WITH [SOURCE]=[RB]=17
1125	BT042 "XOR RA, RB" TEST WITH [RA] = [RB] = 000000
1141	BT043 "XOR RA, RB" TEST WITH [RA] = [RB] = 177777
1158	BT044 "XOR RA, RB" TEST WITH [RB]=052525, [RA]=125252
1173	BT045 "XOR RA, RB" TEST WITH [RA]=052525, [RB]=125252
1189	BT046 GPR ADDRESS INTERACTION TEST
1237	T0 BASIC "BNE" TEST WITH Z=0
1251	T1 BASIC "BNE" TEST WITH Z=1
1266	T2 BASIC "BEQ" TEST WITH Z=1
1279	T3 BASIC "BEQ" TEST WITH Z=0
1294	T4 BASIC "BPL" TEST WITH N=1
1310	T5 BASIC "BPL" TEST WITH N=0
1325	T6 BASIC "MOV (RA), RB" TEST - (RA)=177776
1343	T7 BASIC "CMP RA, (RB)" TEST - [RA] = [DEST]
1360	T10 BASIC "CMP RA, (RB)" TEST - [RA] NOT EQUAL TO [DEST]

1378	T11	BASIC "CMP #N,R" TEST - N = [R]
1401	T12	BASIC "CMP #N,R" TEST - N NOT EQUAL TO [R]
1424	T13	BASIC "MOV RA,(RB)" TEST
1443	T14	BASIC "MOV #N,(R)" TEST
1462	T15	BASIC "MOVB #N,X(R)" TEST - DEST EVEN
1482	T16	BASIC "MOVB #N,X(R)" TEST - DEST ODD
1502	T17	BASIC "TST #0A" TEST WITH [A] GT 0
1520	T20	BASIC "TST #0A" TEST WITH [A] LT 0
1544	T21	BASIC "TST #0A" WITH [A] = 0
1568	T22	BASIC "BIT #N,#0A" WITH BIT SET IN "A"
1586	T23	BASIC "BIT #N,#0A" WITH BIT CLEAR IN "A"
1609	T24	BASIC "TST (R)+" TEST
1615	USER CONTROLLED BREAKPOINT -- BIT0	
1636	T25	BASIC "TST -(R)" TEST
1667	T26	BASIC "COM #0A" TEST
1687	T27	BASIC "INC #0A" TEST
1706	T30	BASIC "DEC RN" TEST
1723	T31	BASIC "DEC #0A" TEST
1742	T32	BASIC "CLR X(R)" TESTS
1762	T33	BASIC "ASL RN" TEST WITH [DEST]=125252 AND C(0)
1784	T34	BASIC "ASL RN" TEST WITH [DEST]=052525 AND C(1)
1805	T35	BASIC "ROL RN" TEST WITH [DEST]=125252 AND C(0)
1827	T36	BASIC "ROL RN" TEST WITH [DEST]=052524 AND C(1)
1849	T37	BASIC "TSTB (R)" TEST - EVEN ADDRESS
1872	T40	BASIC "TSTB (R)" TEST - ODD ADDRESS
1897	T41	BASIC "TSTB #0A" TEST - EVEN ADDRESS
1921	T42	BASIC "TSTB #0A" TEST - ODD ADDRESS
1945	T43	BASIC "DECB 1(SP)"
1967	T44	BASIC "MOV #0A,R" TEST
1984	T45	BASIC "MOV #N,X(R)" TEST
2004	T46	BASIC "MOV #N,(R)" TEST
2023	T47	BASIC "MOV (RA)+,RB" TEST
2047	T50	BASIC "MOV #0A,#0B"
2065	T51	BASIC "MOV X(R),PC" TEST
2081	T52	BASIC "MOV #0A,(R)" TEST
2100	T53	BASIC "MOV X(RA),RB" TEST
2118	T54	BASIC "MOV RA,-(RB)" TEST
2144	T55	BASIC "MOV #0A,-(R)" TEST
2170	T56	BASIC "MOV (R),#0A" TEST
2190	T57	BASIC "MOV -(R),#0A" TEST
2215	T60	BASIC "MOV (RA),RB" TEST
2239	T61	BASIC "MOV X(RA),RB" TEST
2256	T62	BASIC "MOV #X(RA),RB" TEST
2275	T63	BASIC "MOV (R)+,X(R)" TEST
2302	T64	BASIC "CMP R,#0A" TEST WITH [R] = [A]
2308	USER CONTROLLED BREAKPOINT -- BIT1	
2331	T65	BASIC "CMP R,#0A" WITH [R] NOT EQUAL TO [A]
2350	T66	BASIC "BIS #N,#0A" TEST - N=177777,[A]=000000
2369	T67	BASIC "BIC #N,#0A" TEST
2388	T70	BASIC "BIC #N,R" TEST
2406	T71	BASIC "BIC #N,2(SP)" TEST
2431	T72	BASIC "ADD #N,RN" TEST
2448	T73	BASIC "ADD #N,(R)" TEST
2467	T74	BASIC "ADD #N,X(R)" TEST
2487	T75	BASIC "CMPB #N,(SP)+" TEST
2520	T76	BASIC "CMPB (RA)+,(RB)+" - SRC AND DEST EVEN

2558	T77	BASIC	"CMPB (RA)+,(RB)+"	- SRC AND DEST ODD
2596	T100	BASIC	"CMPB (RA)+,(RB)+"	- SRC / EVEN DEST / ODD
2634	T101	BASIC	"CMPB (RA)+,(RB)+"	- SRC / ODD DEST / EVEN
2672	T102	BASIC	"MOVB (RA)+,X(RB)"	- SRC EVEN / DEST EVEN
2698	T103	BASIC	"MOVB (RA)+,X(RB)"	- SRC ODD / DEST ODD
2724	T104	BASIC	"MOVB (RA)+,X(RB)"	- SRC EVEN / DEST ODD
2750	T105	BASIC	"MOVB (RA)+,X(RB)"	- SRC ODD / DEST EVEN
2777	T106	BASIC	"MOVB 2(RA),(RB)+"	TEST - SRC EVEN / DEST EVEN
2804	T107	BASIC	"MOVB 2(RA),(RB)+"	TEST - SRC ODD / DEST EVEN
2831	T110	BASIC	"MOVB 2(RA),(RB)+"	TEST - SRC EVEN / DEST ODD
2858	T111	BASIC	"MOVB 2(RA),(RB)+"	TEST - SRC ODD / DEST ODD
2885	T112	BASIC	"MOVB -(RA),RB"	TEST - SRC EVEN ADDR
2908	T113	BASIC	"MOVB -(RA),RB"	TEST - SRC ODD ADDR
2932	T114	BASIC	"MOVB (RA)+,-(SP)"	TEST - SRC ADDR EVEN
2969	T115	BASIC	"MOVB (RA)+,-(SP)"	TEST - SRC ADDR ODD
3006	T116	BASIC	"MOVB X(R),200A"	TEST - SRC EVEN / DEST EVEN
3025	T117	BASIC	"MOVB X(R),200A"	TEST - SRC ODD / DEST EVEN
3044	T120	BASIC	"MOVB X(R),200A"	TEST - SRC EVEN / DEST ODD
3063	T121	BASIC	"MOVB X(R),200A"	TEST - SRC ODD / DEST ODD
3069		USER CONTROLLED BREAKPOINT	-- BIT2	
3087	T122	BASIC	QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLG=0	
3104	T123	BASIC	QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLAG=1	
3133	T124	BASIC	BVC TEST WITH V=1	
3148	T125	BASIC	BVC TEST WITH V=0	
3163	T126	BASIC	BGE TEST WITH N,V = 00	
3178	T127	BASIC	BGE TEST WITH N,V = 01	
3194	T130	BASIC	BGE TEST WITH N,V = 10	
3210	T131	BASIC	BGE TEST WITH N,V = 11	
3226	T132	BASIC	BLT TEST WITH N,V = 00	
3241	T133	BASIC	BLT TEST WITH N,V = 01	
3257	T134	BASIC	BLT TEST WITH N,V = 10	
3273	T135	BASIC	BLT TEST WITH N,V = 11	
3289	T136	BASIC	BGT TEST WITH Z = 1 AND N,V = 01	
3305	T137	BASIC	BGT TEST WITH Z = 0 AND N,V = 01	
3321	T140	BASIC	BGT TEST WITH Z = 1 AND N,V = 00	
3337	T141	BASIC	BGT TEST WITH Z = 0 AND N,V = 00	
3352	T142	BASIC	BGT TEST WITH Z = 1 AND N,V = 01	
3368	T143	BASIC	BGT TEST WITH Z = 1 AND N,V = 10	
3384	T144	BASIC	BGT TEST WITH Z = 1 AND N,V = 11	
3400	T145	BASIC	BGT TEST WITH Z=0 AND N,V=11	
3416	T146	BASIC	BHI TEST WITH Z,C = 00	
3431	T147	BASIC	BHI TEST WITH Z,C = 01	
3447	T150	BASIC	BHI TEST WITH Z,C = 10	
3463	T151	BASIC	BHI TEST WITH Z,C = 11	
3479	T152	BASIC	NEG MODE 0 TEST : (DEST) GT 0	
3506	T153	BASIC	"SUB 0,00" TEST	
3525	T154	BASIC	"SUB 00,00" TEST	
3543	T155	BASIC	"RTS RN" TEST - <N:C> = 1111	
3580	T156	BASIC	"RTS PC" TEST	
3602	T157	BASIC	"JSR PC,00A" TEST	
3608		USER CONTROLLED BREAKPOINT	-- BIT3	
3627	T160	BASIC	"RTI" TEST - N:C=0000	
3659	T161	BASIC	"RTI" TEST WITH N:C=1111	
3682	T162	BASIC	"IOT" TEST -VERIFY LOADING PSW WITH 357	
3723	T163	BASIC	"IOT" TEST - VERIFY LINKAGE TO SCOPE SERVICE	
3745	T164	BASIC	"IOT" TEST -VERIFY LOADING PSW WITH 357	

3786	T165	BASIC IOT TEST - VERIFY LOADING PSW WITH 000
3810	T166	BASIC "TRAP" TEST - LINKAGE TO PRINT ROUTINE
3831	T167	BASIC "EMT" TEST - LINKAGE TO ERROR SERVICE
3851	T170	BASIC TEST OF RSVD INSTR. TRAP LINKAGE
3876	T171	BASIC TEST OF BUS TIMEOUT TRAP LINKAGE
3898	T172	BASIC TEST FOR ACCESSING DL11 REGISTERS
3927	T173	BASIC TEST OF DL11 - XCSR - READY(1)
3949	T174	BASIC TEST OF DL11 - XCSR - MAINT BIT (0)
3968	T175	BASIC TEST OF DL11 XCSR - MAINT BIT = 1
3987	T176	BASIC DL11 OUT / IN ECHO TEST (MAINT MODE)
4068	T177	BCC TEST WITH C=1
4083	T200	BCC TEST WITH C=0
4098	T201	VERIFY NO BRANCH MICROROUTINE DOES NOT CLR FLAGS
4119	T202	VERIFY BRANCH MICROROUTINE DOES NOT CLR FLAGS
4142	T203	VERIFY NO BRANCH MICROROUTINE DOES NOT SET FLAGS
4163	T204	VERIFY BRANCH MICROROUTINE DOES NOT SET FLAGS
4186	T205	BLE TEST WITH Z = 0, AND N,V = 00
4201	T206	BLE TEST WITH Z = 1 AND N,V = 00
4217	T207	BLE TEST WITH Z = 0 AND N,V = 01
4233	T210	BLE TEST WITH Z = 0 AND N,V = 10
4249	T211	BLE TEST WITH Z = 0 AND N,V = 11
4265	T212	BLOS TEST WITH Z,C = 00
4280	T213	BLOS TEST WITH Z,C = 01
4296	T214	BLOS TEST WITH Z,C = 10
4312	T215	BLOS TEST WITH Z,C = 11
4328	T216	SXT MODE 0 TEST WITH N = 0 AND C = 1
4355	T217	SXT MODE 0 TEST WITH N = 0 AND C = 0
4363	USER CONTROLLED BREAKPOINT -- BIT4	
4377	T220	SXT MODE 0 TEST WITH N = 1 AND C = 1
4404	T221	SXT MODE 0 TEST WITH N = 1 AND C = 0
4422	T222	SXT MODE 1 AND 2 TEST WITH N = 0 AND C = 1
4475	T223	SXT MODE 1 TEST WITH N = 0 AND C = 0
4493	T224	SXT MODE 1 TEST WITH N = 1 AND C = 1
4520	T225	SXT MODE 1 TEST WITH N = 1 AND C = 0
4539	T226	SWAB MODE 0 TEST WITH POS. RESULT
4566	T227	SWAB MODE 0 TEST WITH NEG. RESULT
4593	T230	SWAB MODE 1 AND 2 TEST WITH POS. RESULT
4646	T231	SWAB MODE 1 TEST WITH NEG. RESULT
4674	T232	NEG MODE 0 TEST : [DEST] = 0
4701	T233	NEG MODE 0 TEST : [DEST] LT 0
4728	T234	NEG MODE 0 TEST : [DEST] = 100000 (8)
4755	T235	NEG MODE 1 TEST : [DEST] = 0
4783	T236	NEG MODE 1 TEST : [DEST] GT 0
4811	T237	NEG MODE 1 TEST : [DEST] LT 0
4839	T240	NEG MODE 1 TEST: [DEST] = 100000 (8)
4867	T241	ROR TEST - DMO - N:C = 1110
4894	T242	ROR TEST - DMO - N:C = 1000
4921	T243	ROR TEST - DMO - N:C = 0111
4948	T244	ASR TEST - DMO - N:C = 1000
4975	T245	ASR TEST - DMO - N:C = 0101
5002	T246	ASR TEST - DMO - N:C = 1100
5029	T247	ROR TEST - DM1 - N:C = 1110
5057	T250	ROR TEST - DM1 - N:C = 1000
5086	T251	ROR TEST - DM1 - N:C = 0111
5115	T252	ASR TEST - DM1 - N:C = 1000
5144	T253	ASR TEST - DM1 - N:C = 1100

5173	T254	ASR TEST - DM1 - N:C = 0101
5202	T255	RORB TEST - DM2 - EVEN ADDRESS
5233	T256	RORB TEST - DM1 - EVEN ADDRESS
5260	T257	RORB TEST - DM2 - ODD ADDRESS
5268	USER CONTROLLED BREAKPOINT -- BITS	
5296	T260	RORB TEST - DM1 - ODD ADDRESS
5323	T261	ASRB TEST - DM2 - ODD ADDRESS
5355	T262	ASRB TEST - DM1 - ODD ADDRESS
5382	T263	ASRB TEST - DM2 - EVEN ADDRESS
5413	T264	ASRB TEST - DM1 - EVEN ADDRESS
5439	T265	TST DMO TEST - N:C = 1011
5466	T266	TST DMO TEST - N:C = 0100
5494	T267	CLR DMO TEST - N:C = 1011
5521	T270	CLR DMO TEST - N:C = 0000
5547	T271	COM DMO TEST - N:C = 0110
5574	T272	COM DMO TEST - N:C = 1001
5601	T273	INC DMO TEST - N:C = 1011
5628	T274	INC DMO TEST - N:C = 0100
5655	T275	DEC DMO TEST - N:C = 1011
5682	T276	DEC DMO TEST - N:C = 1100
5709	T277	DEC DMO TEST - N:C = 0000
5735	T300	ASL DMO TEST - N:C = 1000
5762	T301	ASL DMO TEST - N:C = 0101
5789	T302	ASL DMO TEST - N:C = 0010
5816	T303	ROL DMO TEST - N:C = 1101
5843	T304	ROL DMO TEST - N:C = 0101
5869	T305	ROL DMO TEST - N:C = 0010
5896	T306	ADC DMO TEST - N:C = 0101
5923	T307	ADC DMO TEST - N:C = 1011
5950	T310	ADC DMO TEST - N:C = 1010
5977	T311	SBC DMO TEST - N:C = 1011
6004	T312	SBC DMO TEST - N:C = 0101
6031	T313	SBC DMO TEST - N:C = 1110
6058	T314	SBC DMO TEST - N:C = 0111
6085	T315	TST DM1 TEST - N:C = 1011
6114	T316	TST DM1 TEST - N:C = 0100
6144	T317	CLR DM1 TEST - N:C = 1011
6152	USER CONTROLLED BREAKPOINT -- BIT6	
6177	T320	CLR DM2 TEST - N:C = 0000
6210	T321	COM DM1 TEST - N:C = 0110
6238	T322	COM DM1 TEST - N:C = 1001
6266	T323	INC DM1 TEST - N:C = 1011
6294	T324	INC DM1 TEST - N:C = 0100
6322	T325	DEC DM1 TEST - N:C = 1011
6350	T326	DEC DM1 TEST - N:C = 1100
6378	T327	DEC DM1 TEST - N:C = 0000
6405	T330	ASL DM1 TEST - N:C = 1000
6433	T331	ASL DM1 TEST - N:C = 0101
6461	T332	ASL DM1 TEST - N:C = 0010
6489	T333	ROL DM1 TEST - N:C = 1101
6517	T334	ROL DM1 TEST - N:C = 0101
6545	T335	ROL DM1 TEST - N:C = 0010
6573	T336	ADC DM1 TEST - N:C = 0101
6601	T337	ADC DM1 TEST - N:C = 1011
6629	T340	ADC DM1 TEST - N:C = 1010
6657	T341	SBC DM1 TEST - N:C = 1011

6685	T342	SBC DM1 TEST - N:C = 0101
6713	T343	SBC DM1 TEST - N:C = 1110
6741	T344	SBC DM1 TEST - N:C = 0111
6769	T345	NEGB - MODE 0 TEST - N:C = 0110
6796	T346	NEGB - MODE 0 TEST - N:C = 0011
6823	T347	NEGB - MODE 0 TEST - N:C = 1101
6850	T350	CLRB - MODE 0 TEST - N:C = 1011
6877	T351	CLRB - MODE 0 TEST - N:C = 0100
6904	T352	CLRB TEST - DM2 - ODD ADDRESS
6932	T353	CLRB TEST - DM1 - ODD ADDRESS
6955	T354	CLRB TEST - DM2 - EVEN ADDRESS
6982	T355	CLRB TEST - DM1 - EVEN ADDRESS
7004	T356	NEGB TEST - DM2 - ODD ADDRESS
7032	T357	NEGB TEST - DM1 - ODD ADDRESS
7040	USER CONTROLLED BREAKPOINT -- BIT7	
7059	T360	NEGB TEST - DM2 - EVEN ADDRESS
7086	T361	NEGB TEST - DM1 - EVEN ADDRESS
7108	T362	ADD TEST - SMO,DM0 - N:C = 1010
7136	T363	ADD TEST - SMO,DM0 - N:C = 0101
7164	T364	ADD SM1,DM0 TEST
7189	T365	ADD SM2,DM0 TEST
7214	T366	ADD SM3,DM0 TEST
7240	T367	ADD SM4,DM0 TEST
7265	T370	ADD SM5,DM0 TEST
7291	T371	ADD SM6,DM0 TEST
7311	T372	ADD SM7,DM0 TEST
7332	T373	ADD SM1,DM1 TEST
7354	T374	ADD SM2,DM1 TEST
7376	T375	ADD SM1,DM2 TEST
7406	T376	ADD SM2,DM2 TEST
7436	T377	ADD SM1,DM3 TEST
7466	T400	ADD SM2,DM3 TEST
7496	T401	ADD SM1,DM4 TEST
7526	T402	ADD SM2,DM4 TEST
7556	T403	ADD SM1,DM5 TEST
7586	T404	ADD SM2,DM5 TEST
7616	T405	ADD SM1,DM6 TEST
7639	T406	ADD SM2,DM6 TEST
7662	T407	ADD SM1,DM7 TEST
7685	T410	ADD SM2,DM7 TEST
7708	T411	"XOR RA, RB" TEST - A=B=000000 N:C=1010
7735	T412	"XOR RA, RB" TEST - A=B=177777 N:C=0101
7763	T413	"XOR RA, RB" TEST - A=125252, B=052525 N:C=0110
7791	T414	"XOR RA, RB" TEST - A=052525, B=125252 N:C=1001
7819	T415	"XOR RA, (RB)" TEST - A=B=000000 N:C=1010
7849	T416	"XOR RA, (RB)" TEST - A=B=177777 N:C=0101
7879	T417	"XOR RA, (RB)" TEST - A=125252, B=052525 N:C=0110
7887	USER CONTROLLED BREAKPOINT -- BIT8	
7913	T420	"XOR RA, (RB)" TEST - A=052525, B=125252 N:C=1001
7943	T421	SUB TEST SMO,DM0 - (SRC) = (DEST) = +, +
7971	T422	SUB TEST SMO,DM0 - (SRC) = (DEST) = -, -
7999	T423	SUB TEST SMO,DM0 - (SRC) = (DEST) = -, +
8027	T424	SUB TEST SMO,DM0 (SRC) = -(DEST) = +, -
8055	T425	SUB TEST SMO,DM0 - "V" BIT SETS
8083	T426	SUB TEST - SMO,DM1 - N:C = 0110
8113	T427	SUB TEST - SMO,DM1 - N:C = 1010

8143	T430	SUB TEST - SMO,DM1 - N:C = 0000
8172	T431	SUB TEST - SM1,DM0 - N:C = 0110
8200	T432	SUB TEST - SM1,DM0 - N:C = 1010
8228	T433	SUB TEST - SM1,DM0 - N:C = 0000
8256	T434	SUB SM1,DM1 TEST - N:C = 0110
8287	T435	SUB SM1,DM2 TEST - N:C = 0110
8319	T436	NEG DM2 TEST
8346	T437	NEG DM3 TEST
8373	T440	NEG DM4 TEST
8400	T441	NEG DM5 TEST
8427	T442	NEG DM6 TEST
8449	T443	NEG DM7 TEST
8471	T444	MOV SM1,DM1 TEST - N:C = 0100
8504	T445	MOV SM2,DM1 TEST - N:C = 0100
8537	T446	MOV SM1,DM1 TEST - N:C = 1011
8570	T447	MOV SM2,DM1 TEST - N:C = 1011
8603	T450	MOV SM1,DM2 TEST - N:C = 0100
8641	T451	MOV SM2,DM2 TEST - N:C = 0100
8679	T452	MOV SM1,DM3 TEST - N:C = 0100
8717	T453	MOV SM2,DM3 TEST - N:C = 0100
8755	T454	MOV SM1,DM4 TEST - N:C = 0100
8793	T455	MOV SM2,DM4 TEST - N:C = 0100
8831	T456	MOV SM1,DM5 TEST - N:C = 0100
8839	USER CONTROLLED BREAKPOINT -- BIT9	
8873	T457	MOV SM2,DM5 TEST - N:C = 0100
8911	T460	MOV SM1,DM6 TEST - N:C = 0100
8944	T461	MOV SM2,DM6 TEST - N:C = 0100
8977	T462	MOV SM1,DM7 TEST - N:C = 0100
9010	T463	MOV SM2,DM7 TEST - N:C = 0100
9043	T464	MOV SMO,DM1 TEST
9065	T465	MOV SMO,DM2 TEST
9087	T466	MOV SMO,DM3 TEST
9109	T467	MOV SMO,DM4 TEST
9131	T470	MOV SMO,DM5 TEST
9154	T471	MOV SMO,DM6 TEST
9176	T472	MOVB TEST - SMO,DM0 - EXTEND 1'S
9204	T473	MOVB TEST - SMO,DM0 - EXTEND 0'S
9233	T474	MOVB TEST - SM1,DM0 - SOURCE ADDR EVEN
9254	T475	MOVB TEST - SM1,DM0 - SOURCE ADDR ODD
9274	T476	MOVB TEST - SM2,DM0 - SOURCE ADDR ODD
9299	T477	MOVB TEST - SM2,DM0 - SOURCE ADDR EVEN
9324	T500	MOVB TEST - SM1,DM1 - SRC ADR ODD / DST ADR EVEN
9346	T501	MOVB TEST - SM1,DM2 - SRC ADR ODD / DST ADR EVEN
9374	T502	MOVB TEST - SM1,DM3 - SRC ADR ODD / DST ADR EVEN
9402	T503	MOVB TEST - SM1,DM4 - SRC ADR ODD / DST ADR EVEN
9430	T504	MOVB TEST - SM1,DM5 - SRC ADR ODD / DST ADR EVEN
9458	T505	MOVB TEST - SM1,DM6 - SRC ADR ODD / DST ADR EVEN
9481	T506	MOVB TEST - SM1,DM7 - SRC ADR ODD / DST ADR EVEN
9504	T507	MOVB SMO,DM1 TEST
9527	T510	MOVB SMO,DM2 TEST
9550	T511	MOVB SMO,DM3 TEST
9573	T512	MOVB SMO,DM4 TEST
9596	T513	MOVB SMO,DM6 TEST
9619	T514	BIS TEST - SMO,DM0 - N:C = 0111
9647	T515	BIS TEST - SMO,DM0 - N:C = 1000
9655	USER CONTROLLED BREAKPOINT -- BIT10	

9679	T516	BIC TEST - SMO,DMO - N:C = 0111
9707	T517	BIC TEST - SMO,DMO - N:C = 1000
9735	T520	BIT TEST - SMO,DMO - N:C = 0111
9764	T521	BIT TEST - SMO,DMO - N:C = 1000
9792	T522	CMP TEST - SMO,DMO - N:C = 0110
9820	T523	CMP TEST - SMO,DMO - N:C = 1010
9848	T524	CMP TEST - SMO,DMO - N:C = 0000
9875	T525	BIS TEST - SMO,DM1 - N:C = 0111
9905	T526	BIS TEST - SMO,DM1 - N:C = 1000
9935	T527	BIC TEST - SMO,DM1 - N:C = 0111
9965	T530	BIC TEST - SMO,DM1 - N:C = 1000
9995	T531	BIT TEST - SMO,DM1 - N:C = 0111
10025	T532	BIT TEST - SMO,DM1 - N:C = 1000
10054	T533	CMP TEST - SMO,DM1 - N:C = 1010
10084	T534	CMP TEST - SMO,DM1 - N:C = 0110
10114	T535	CMP TEST - SMO,DM1 - N:C = 0000
10143	T536	BIS TEST - SM1,DMO - N:C = 0111
10171	T537	BIS TEST - SM1,DMO - N:C = 1000
10199	T540	BIC TEST - SM1,DMO - N:C = 0111
10228	T541	BIC TEST - SM1,DMO - N:C = 1000
10256	T542	BIT TEST - SM1,DMO - N:C = 0111
10284	T543	BIT TEST - SM1,DMO - N:C = 1000
10311	T544	CMP TEST - SM1,DMO - N:C = 0110
10339	T545	CMP TEST - SM1,DMO - N:C = 1010
10367	T546	CMP TEST - SM1,DMO - N:C = 0000
10395	T547	BIS SM1,DM1 TEST - N:C = 0111
10425	T550	BIS SM1,DM1 TEST - N:C = 1000
10455	T551	BIC SM1,DM1 TEST - N:C = 0111
10486	T552	BIC SM1,DM1 TEST - N:C = 1000
10517	T553	BIT SM1,DM1 TEST - N:C = 1000
10548	T554	BIT SM1,DM1 TEST - N:C = 0111
10556	USER CONTROLLED BREAKPOINT -- BIT11	
10583	T555	CMP SM1,DM1 TEST - N:C = 1010
10614	T556	CMP SM1,DM1 TEST - N:C = 0110
10645	T557	CMP SM1,DM1 TEST - N:C = 0000
10675	T560	BISB SM1,DMO TEST - SOURCE ADDR ODD
10695	T561	BISB SM1,DM1 TEST - SOURCE ADDR ODD
10717	T562	BISB SM1,DM2 TEST - SOURCE ADDR ODD
10740	T563	BISB SM1,DM3 TEST - SOURCE ADDR ODD
10763	T564	BISB SM1,DM4 TEST - SOURCE ADDR ODD
10786	T565	BISB SM1,DM5 TEST - SOURCE ADDR ODD
10809	T566	BISB SM1,DM6 TEST - SOURCE ADDR ODD
10832	T567	BISB SM1,DM7 TEST - SOURCE ADDR ODD
10855	T570	BISB SMO,DM2 TEST - DEST ADDR EVEN
10877	T571	BISB SMO,DM1 TEST - DEST ADDR ODD
10900	T572	BISB SMO,DM1 TEST - DEST ADDR EVEN
10922	T573	BISB SM1,DM1 TEST - DEST ADDR ODD
10945	T574	JMP MODE 1 TEST; FLAGS = 1111
10968	T575	JMP MODE 1 TEST; FLAGS = 0000
10991	T576	JMP MODE 2 TEST; FLAGS = 1111
11019	T577	JMP MODE 2 TEST; FLAGS = 0000
11042	T600	JMP TEST MODE 3; FLAGS = 1111
11075	T601	JMP TEST MODE 3; FLAGS = 0000
11102	T602	JMP TEST MODE 4; FLAGS = 1111
11134	T603	JMP TEST MODE 4; FLAGS = 0000
11157	T604	JMP TEST MODE 5; FLAGS = 1111

11188	T605	JMP TEST MODE 5; FLAG = 0000
11215	T606	JMP TEST MODE 6; FLAGS = 1111
11242	T607	JMP TEST MODE 6; FLAGS = 0000
11269	T610	JMP TEST MODE 7; FLAGS = 1111
11300	T611	JMP TEST MODE 7; FLAGS = 0000
11331	T612	JSR MODE 1 TEST - LOAD PC / PUSH SP
11360	T613	JSR MODE 1 TEST - CHECK RN AND OLD PC
11392	T614	JSR MODE 1 TEST - N:C = 0000
11400	USER CONTROLLED BREAKPOINT -- BIT12	
11422	T615	JSR MODE 1 TEST - N:C = 1111
11448	T616	JSR MODE 2 TEST
11478	T617	JSR MODE 3 TEST
11512	T620	JSR MODE 4 TEST
11545	T621	JSR MODE 5 TEST
11579	T622	JSR MODE 6 TEST
11608	T623	JSR MODE 7 TEST
11643	T624	SOB TEST, [R] = 1, NO BRANCH
11660	T625	SOB TEST, [R] = 5, BRANCH 4 TIMES
11688	T626	SOB TEST, [R] = 1, FLAGS = 1111
11708	T627	SOB TEST, [R] = 1, FLAGS = 0000
11728	T630	SOB TEST, [R] = 5, FLAGS = 1111
11748	T631	SOB TEST, [R] = 5, FLAGS = 0000
11768	T632	RTS TEST - N:C = 0000
11810	T633	RTT TEST - N:C = 1111
11848	T634	RTT TEST - N:C = 0000
11886	T635	MARK INSTRUCTION TEST - N:C=0000
11947	T636	MARK INSTRUCTION TEST - N:C=1111
12008	T637	BASIC LINE CLOCK RESPONSE TEST
12033	T640	LINE CLOCK TEST - LKCSR BIT 7 SET
12052	T641	LINE CLOCK TEST - LKCSR BIT 6 CLEAR
12071	T642	LINE CLOCK TEST - LKCSR BIT 6 SET
12102	T643	LINE CLK BASIC INTERRUPT TEST
12135	T644	RESET TEST - N:C = 1111
12169	T645	RESET TEST - N:C = 0000
12203	T646	WAIT INSTRUCTION TEST - [PSW] = 151
12261	T647	WAIT INSTRUCTION TEST - [PSW] = 010
12319	T650	BR PRIORITY ARBITRATION TEST - LEVEL 1 USING LINE CLK
12351	T651	BR PRIORITY ARBITRATION TEST - LEVEL 2 USING LINE CLK
12383	T652	BR PRIORITY ARBITRATION TEST - LEVEL 3 USING LINE CLK
12415	T653	BR PRIORITY ARBITRATION TEST - LEVEL 4 USING LINE CLK
12447	T654	BR PRIORITY ARBITRATION TEST - LEVEL 5 USING LINE CLK
12479	T655	BR PRIORITY ARBITRATION TEST - LEVEL 6 USING LINE CLK
12487	USER CONTROLLED BREAKPOINT -- BIT13	
12516	T656	BR PRIORITY ARBITRATION TEST - LEVEL 7 USING DL11
12549	T657	"CLR 2#PSW" ALLOWS IMMEDIATE BR-BG-INTR SEQUENCE
12594	T660	"BR6 VS BR4" PRIORITY ARBITRATION TEST
12682	T661	"BPT" TRAP LINKAGE TEST
12703	T662	RED ZONE OVERFLOW TEST - MOV R, -(SP)
12744	T663	YELLOW ZONE OVERFLOW TEST - MOV R, -(SP)
12783	T664	YELLOW ZONE OVERFLOW TEST - (CMP RO, -(SP))
12809	T665	YELLOW ZONE OVERFLOW TEST - (BIT RO, -(SP))
12835	T666	YELLOW ZONE OVERFLOW TEST - (TST -(SP))
12861	T667	ODD ADDRESS ERROR TEST - SUB RA, (RB) - (RB) = ODD
12886	T670	TEST FOR ODD ADDR. ERROR TRAP FOR DEST. DEFERRED MODES
12926	T671	TEST FOR ODD ADDR. ERROR TRAP FOR SOURCE DEFERRED MODES
12965	T672	TEST FOR ODD ADDR. ERROR TRAP FOR JMP DEST DEFERRED MODES

13018	T673	TEST FOR STACK OVFLW FOR DEST MODES 1, 2, 4, AND 6.
13079	T674	TEST FOR STACK OVFLW FOR MOV DEST MODES 1, 2, 4, AND 6.
13140	T675	TEST THAT JSR CAN CAUSE OVERFLOW TRAP
13171	T676	TEST THAT 1ST PUSH IN TRAP MICROUTINE CAUSES OVFLW TRAP
13204	T677	TEST THAT 2ND PUSH IN TRAP MICROUTINE CAUSES OVFLW TRAP
13237	T700	ILLEGAL INSTRUCTION TEST - JSR RN,%R
13261	T701	ILLEGAL INSTRUCTION TEST - JMP %R
13285	T702	BUS TIMEOUT TRAP TEST - TST (R)
13308	T703	"T" BIT TRAP TEST
13340	T704	TEST PUSH INTO PSW WITH [SP] = 000000
13378	T705	TEST PUSH INTO SR WITH [SP] = 177572
13414	T706	TEST PUSH INTO SLR WITH [SP] = 177776
13450	T707	RSVD INSTRUCTION TEST - 000007 THRU 000077
13477	T710	RSVD INSTRUCTION TEST - 000210 THRU 000237
13505	T711	RSVD INSTRUCTION TEST - 007000 THRU 007777
13533	T712	RSVD INSTRUCTION TEST - 075000 THRU 076777
13564	T713	RSVD INSTRUCTION TEST - 106400 THRU 107777
13605	T714	BUT SERVICE -- ONE WORD INSTRUCTIONS--ALL MODES -- FROM TABLE
13636	T715	BUT SERVICE TEST - (RTI)
13652	T716	BUT SERVICE TEST - (JSR %R,%A)
13660		USER CONTROLLED BREAKPOINT -- BIT14
13678	T717	BUT SERVICE TEST - (JMP A)
13695	T720	BUT SERVICE TEST - (JMP %A)
13713	T721	BUT SERVICE TEST - (RTS PC)
13734	T722	ALU ADD FUNCTION TEST
13795	T723	ALU SUB FUNCTION TEST
13856	T724	ALU "AND" FUNCTION TEST USING BIC INSTRUCTION
13917	T725	ALU "OR" FUNCTION TEST USING BIS INSTRUCTION
13978	T726	INC / DEC / ADD TEST - CYCLE NO.S 000000-077777
14024	T727	INC / DEC / ADD TEST - CYCLE NO.S 077777-000000
14069	T730	MUL RA, RB TEST ; N:C = 1111
14100	T731	MUL (RA), RB TEST ; N:C = 0000-SET C
14132	T732	MUL (RA)+, RB TEST ; N:C = 0000-SET Z
14169	T733	MUL @ (RA)+, RB TEST ; N:C = 0000-SET N ; SRC, DST = -,+
14206	T734	MUL -(RA), RB TEST ; N:C = 1111-CLR ALL BUT N ; SRC, DST = +,-
14243	T735	MUL @-(RA), RB TEST ; N:C = 1111-CLR ALL BUT C ; SRC, DST = -,-
14280	T736	MUL X(RA), RB TEST ; N:C = 1111 TO 0100
14312	T737	MUL @X(RA), RB TEST
14337	T740	DIV #N, RA TEST ; N:C = 1111
14367	T741	DIV #N, RA TEST ; RA NEGATIVE ; N:C = 0000
14397	T742	DIV #N, RA TEST ; N:C = 0000 TO 0100
14427	T743	DIV #-N, RA TEST ; RA POS
14450	T744	DIV TEST - V BIT GETS SET
14497	T745	ASH #N, RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010
14522	T746	ASH #N, RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
14548	T747	ASH #N, RA TEST ; SHIFT LEFT ; N:C = 1111 TO 1000
14573	T750	ASHC #N, RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010
14602	T751	ASHC #N, RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
14632	T752	ASHC #N, RA TEST ; SHIFT RIGHT ; N:C = 1111 TO 1000
14639		USER CONTROLLED BREAKPOINT -- BIT15
14678	T753	CHECK MED IS ILLEGAL IN USER - EXECUTES IN KERNAL
14717	T754	MED TEST - R/W DATA PATTERNS TO REGS
14769	T755	MED TEST - VERIFY NOPS; READ R7 IN A & B SP
14816	T756	MED TEST - CSP CONSTANTS CHECK
14847	T757	MED TEST - MICROBK CHECK OF MICRO-POINTS
14935	T760	PHYSICAL ADDRESS & ODD ADDRESS ERROR LOGGING

K02

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23
DQKDA.A.P11 08-FEB-77 16:17 TABLE OF CONTENTS

14997	T761	CHECK DISABLE PARITY ERROR TRAP
15068	T762	CHECK PARITY ERROR BITS IN MEMERR REG. IN BACKUP MODE OF CACHE (TRAP)
15106	T763	CHECK UNIBUS TIMEOUT, ODD ADDRESS AND LOG CONTINUOUS MODE
15184	T764	CHECK ILLEGAL INTERNAL ADDRESS TRAP
15224	T765	CHECK LOG SERVICE & MEMERR LOGS LO-HI BYTE & TAG, IN CACHE ABORT MODE
15347	T766	CHECK "LOG FIRST" MODE OF ERROR LOGGING
15435	T767	CHECK LAST INTERRUPT VECTOR IS LOGGED IN FLAG REG.
15479		END OF PASS ROUTINE
15517	/ / / / /	UTILITIES / / / / /
15520		POWER DOWN AND UP ROUTINES
15574	"T"	BIT SERVICE ROUTINE
15581		MICROBREAK TRAP SERVICE ROUTINE
15592		RSVD INSTRUCTION TRAP SERVICE ROUTINE
15653		BUS ERROR TRAP SERVICE ROUTINE
15714		SCOPE HANDLER ROUTINE
15797		ERROR HANDLER ROUTINE
15860		ERROR MESSAGE TYPEOUT ROUTINE
15909	PRINT	ROUTINES
15916		TYPE ROUTINE
15996		BINARY TO OCTAL (ASCII) AND TYPE
16074		APT COMMUNICATIONS ROUTINE
16132		TRAP DECODER
16155		TRAP TABLE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```

.TITLE MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
.*COPYRIGHT (C) JULY 1976
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY JACK RICH
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
.*
.SBTTL OPERATIONAL SWITCH SETTINGS
.*
.*      SWITCH          USE
.*      -----
.*      15             HALT ON ERROR
.*      14             LOOP ON TEST
.*      13             INHIBIT ERROR TYPEOUTS
.*      12             INHIBIT ID MESSAGE & UNEXPECTED TRAP MESSAGES
.*      11             INHIBIT ITERATIONS
.*      10             LOOP ON TEST IN SWR<8:0>
.*      9              LOOP ON ERROR
.ENABLE ABS
.SBTTL BASIC DEFINITIONS

.*INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
STACK= 1000
001000
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

.*MISCELLANEOUS DEFINITIONS
HT= 11                ;;CODE FOR HORIZONTAL TAB
000011
LF= 12                ;;CODE FOR LINE FEED
000012
CR= 15                ;;CODE FOR CARRIAGE RETURN
000015
CRLF= 200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
000200
PS= 177776            ;;PROCESSOR STATUS WORD
177776
.EQUIV PS,PSW
STKLM= 177774         ;;STACK LIMIT REGISTER
177774
PIRQ= 177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177772
DSWR= 177570          ;;HARDWARE SWITCH REGISTER
177570
DDISP= 177570         ;;HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0                ;;GENERAL REGISTER
000000
R1= %1                ;;GENERAL REGISTER
000001
R2= %2                ;;GENERAL REGISTER
000002
R3= %3                ;;GENERAL REGISTER
000003
R4= %4                ;;GENERAL REGISTER
000004
R5= %5                ;;GENERAL REGISTER
000005
R6= %6                ;;GENERAL REGISTER
000006
R7= %7                ;;GENERAL REGISTER
000007
SP= %6                ;;STACK POINTER
000006
PC= %7                ;;PROGRAM COUNTER
000007

.*PRIORITY LEVEL DEFINITIONS
PRO= 0                ;;PRIORITY LEVEL 0
000000

```

57	000040	PR1=	40	::	PRIORITY LEVEL	1
58	000100	PR2=	100	::	PRIORITY LEVEL	2
59	000140	PR3=	140	::	PRIORITY LEVEL	3
60	000200	PR4=	200	::	PRIORITY LEVEL	4
61	000240	PR5=	240	::	PRIORITY LEVEL	5
62	000300	PR6=	300	::	PRIORITY LEVEL	6
63	000340	PR7=	340	::	PRIORITY LEVEL	7

.*"SWITCH REGISTER" SWITCH DEFINITIONS

65		SW15=	100000
66	100000	SW14=	40000
67	040000	SW13=	20000
68	020000	SW12=	10000
69	010000	SW11=	4000
70	004000	SW10=	2000
71	002000	SW09=	1000
72	001000	SW08=	400
73	000400	SW07=	200
74	000200	SW06=	100
75	000100	SW05=	40
76	000040	SW04=	20
77	000020	SW03=	10
78	000010	SW02=	4
79	000004	SW01=	2
80	000002	SW00=	1
81	000001	.EQUIV	SW09, SW9
82		.EQUIV	SW08, SW8
83		.EQUIV	SW07, SW7
84		.EQUIV	SW06, SW6
85		.EQUIV	SW05, SW5
86		.EQUIV	SW04, SW4
87		.EQUIV	SW03, SW3
88		.EQUIV	SW02, SW2
89		.EQUIV	SW01, SW1
90		.EQUIV	SW00, SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

93		BIT15=	100000
94	100000	BIT14=	40000
95	040000	BIT13=	20000
96	020000	BIT12=	10000
97	010000	BIT11=	4000
98	004000	BIT10=	2000
99	002000	BIT09=	1000
100	001000	BIT08=	400
101	000400	BIT07=	200
102	000200	BIT06=	100
103	000100	BIT05=	40
104	000040	BIT04=	20
105	000020	BIT03=	10
106	000010	BIT02=	4
107	000004	BIT01=	2
108	000002	BIT00=	1
109	000001	.EQUIV	BIT09, BIT9
110		.EQUIV	BIT08, BIT8
111		.EQUIV	BIT07, BIT7
112			

```

113 .EQUIV BIT06,BIT6
114 .EQUIV BIT05,BIT5
115 .EQUIV BIT04,BIT4
116 .EQUIV BIT03,BIT3
117 .EQUIV BIT02,BIT2
118 .EQUIV BIT01,BIT1
119 .EQUIV BIT00,BIT0
120
121 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
122 ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS
123 RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS
124 TBITVEC=14 ;: "T" BIT
125 TRTVEC= 14 ;: TRACE TRAP
126 BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)
127 IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) **SCOPE**
128 PWRVEC= 24 ;: POWER FAIL
129 EMTVEC= 30 ;: EMULATOR TRAP (EMT) **ERROR**
130 TRAPVEC=34 ;: "TRAP" TRAP
131 TKVEC= 60 ;: TTY KEYBOARD VECTOR
132 TPVEC= 64 ;: TTY PRINTER VECTOR
133 PIRQVEC=240 ;: PROGRAM INTERRUPT REQUEST VECTOR
134 .SBTTL TRAP CATCHER
135
136 .=0
137 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
138 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
139 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
140 .=174
141 000174 000000 DISPREG: .WORD 0 ;: SOFTWARE DISPLAY REGISTER
142 000176 000000 SWREG: .WORD 0 ;: SOFTWARE SWITCH REGISTER
143 .SBTTL STARTING ADDRESS(ES)
144 000200 000137 001630 JMP @#START ;: JUMP TO STARTING ADDRESS OF PROGRAM
145 .=700 ;: PUT APT HEADER IN STACK AREA
146 .SBTTL APT PARAMETER BLOCK
147
148 ;:*****
149 ;:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
150 ;:*****
151 .SX=. ;: SAVE CURRENT LOCATION
152 .=24 ;: SET POWER FAIL TO POINT TO START OF PROGRAM
153 000024 000200 200 ;: FOR APT START UP
154 .=44 ;: POINT TO APT INDIRECT ADDRESS PNTR.
155 000044 000700 $APTHDR ;: POINT TO APT HEADER BLOCK
156 .=.SX ;: RESET LOCATION COUNTER
157 ;:*****
158 ;:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
159 ;:INTERFACE SPEC.
160
161 000700 $APTHD:
162 000700 000000 SHIBTS: .WORD 0 ;: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
163 000702 001120 $MADR: .WORD $MAIL ;: ADDRESS OF APT MAILBOX (BITS 0-15)
164 000704 000000 $STMT: .WORD ;: RUN TIM OF LONGEST TEST
165 000706 000000 $PASTM: .WORD ;: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
166 000710 000000 $UNITH: .WORD ;: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
167 000712 000014 .WORD $ETEND-$MAIL/2 ;: LENGTH MAILBOX-ETABLE(WORDS)
168 .SBTTL ACT11 HOOKS

```

169
170
171
172 000714
173 000046
174 000046 060570
175 000052 000052
176 000052 000000
177 000714

```
;;*****  
;HOOKS REQUIRED BY ACT11  
    $SVPC=.                ;SAVE PC  
    .=46  
    $ENDAD                ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP  
    .=52  
    .WORD 0                ;;2)SET LOC.52 TO ZERO  
    .=$SVPC                ;; RESTORE PC
```

178
179
180
181
182
183
184
185 001000
186 001000 000000
187 001002 000
188 001003 000
189 001004 000000
190 001006 000000
191 001010 000000
192 001012 000000
193 001014 000
194 001015 001
195 001016 000000
196 001020 000000
197 001022 000000
198 001024 000000
199 001026 000000
200 001030 000000
201 001032 000000
202 001034 000
203 001035 000
204 001036 000000
205 001040 177570
206 001042 177570
207 001044 177560
208 001046 177562
209 001050 177564
210 001052 177566
211 001054 000
212 001055 002
213 001056 012
214 001057 000
215 001060 000000
216
217 001062 000000
218 001064 000000
219 001066 000000
220 001070 000000
221 001072 000000
222 001074 000000
223 001076 000000
224 001100 000000
225 001102 000000
226 001104 000000
227 001106 000000
228 001110 000000
229 001112 000000
230 001114 077
231 001115 015
232 001116 000012
233

.SBTTL COMMON TAGS

*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
*USED IN THE PROGRAM.

SCMTAG: .=1000

;; START OF COMMON TAGS

.WORD 0
STSTNM: .BYTE 0
SERFLG: .BYTE 0
SICNT: .WORD 0
SLPADR: .WORD 0
SLPERR: .WORD 0
SERTTL: .WORD 0
SITEMB: .BYTE 0
SERMAX: .BYTE 1
SERRPC: .WORD 0
SGDADR: .WORD 0
SBDADR: .WORD 0
SGDDAT: .WORD 0
SBDAT: .WORD 0

;; CONTAINS THE TEST NUMBER
;; CONTAINS ERROR FLAG
;; CONTAINS SUBTEST ITERATION COUNT
;; CONTAINS SCOPE LOOP ADDRESS
;; CONTAINS SCOPE RETURN FOR ERRORS
;; CONTAINS TOTAL ERRORS DETECTED
;; CONTAINS ITEM CONTROL BYTE
;; CONTAINS MAX. ERRORS PER TEST
;; CONTAINS PC OF LAST ERROR INSTRUCTION
;; CONTAINS ADDRESS OF 'GOOD' DATA
;; CONTAINS ADDRESS OF 'BAD' DATA
;; CONTAINS 'GOOD' DATA
;; CONTAINS 'BAD' DATA
;; RESERVED--NOT TO BE USED

SAUTOB: .BYTE 0
SINTAG: .BYTE 0
SWR: .WORD DSWR
DISPLAY: .WORD DDISP
STKS: 177560
STKB: 177562
STPS: 177564
STPB: 177566

;; AUTOMATIC MODE INDICATOR
;; INTERRUPT MODE INDICATOR

SNUL: .BYTE 0
SFILLS: .BYTE 2
SFILLC: .BYTE 12
STPFLG: .BYTE 0
SREGAD: .WORD 0

;; ADDRESS OF SWITCH REGISTER
;; ADDRESS OF DISPLAY REGISTER
;; TTY KBD STATUS
;; TTY KBD BUFFER
;; TTY PRINTER STATUS REG. ADDRESS
;; TTY PRINTER BUFFER REG. ADDRESS
;; CONTAINS NULL CHARACTER FOR FILLS
;; CONTAINS # OF FILLER CHARACTERS REQUIRED
;; INSERT FILL CHARS. AFTER A "LINE FEED"
;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
;; CONTAINS THE ADDRESS FROM WHICH (SREG0) WAS OBTAINED

SREG0: .WORD 0
SREG1: .WORD 0
SREG2: .WORD 0
SREG3: .WORD 0
SREG4: .WORD 0
SREG5: .WORD 0
STMP0: .WORD 0
STMP1: .WORD 0
STMP2: .WORD 0
STMP3: .WORD 0
STMP4: .WORD 0

;; CONTAINS ((SREGAD)+0)
;; CONTAINS ((SREGAD)+2)
;; CONTAINS ((SREGAD)+4)
;; CONTAINS ((SREGAD)+6)
;; CONTAINS ((SREGAD)+10)
;; CONTAINS ((SREGAD)+12)

STIMES: 0
SESCAPE: 0
SQUES: .ASCII /?/
SCRLF: .ASCII <15>
SLF: .ASCII <12>

;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; USER DEFINED
;; MAX. NUMBER OF ITERATIONS
;; ESCAPE ON ERROR ADDRESS
;; QUESTION MARK
;; CARRIAGE RETURN
;; LINE FEED

.SBTTL APT MAILBOX-ETABLE

234
235
236
237
238 001120
239 001120 000000
240 001122 000000
241 001124 000000
242 001126 000000
243 001130 000000
244 001132 000000
245 001134 000000
246 001136 000000
247 001140
248 001140 000
249 001141 000
250 001142 000000
251 001144 000000
252 001146 000000
253
254
255
256
257
258
259 001150
260

```
*****
;EVEN
$MAIL:
$MSGTY: .WORD  AMSGTY  ;; APT MAILBOX
$FATAL: .WORD  AFATAL   ;; MESSAGE TYPE CODE
$TESTN: .WORD  ATESTN   ;; FATAL ERROR NUMBER
$PASS:  .WORD  APASS    ;; TEST NUMBER
$DEVCT: .WORD  ADEVCT   ;; PASS COUNT
$UNIT:  .WORD  AUNIT    ;; DEVICE COUNT
$MSGAD: .WORD  AMSGAD   ;; I/O UNIT NUMBER
$MSGLG: .WORD  AMSGLG   ;; MESSAGE ADDRESS
$ETABLE:      ;; MESSAGE LENGTH
           ;; APT ENVIRONMENT TABLE
$ENV:      .BYTE  AENV   ;; ENVIRONMENT BYTE
$ENVM:     .BYTE  AENVM  ;; ENVIRONMENT MODE BITS
$SWREG:    .WORD  ASWREG  ;; APT SWITCH REGISTER
$USWR:     .WORD  AUSWR   ;; USER SWITCHES
$CPUOP:    .WORD  ACPUOP  ;; CPU TYPE, OPTIONS
           ;; BITS 15-11=CPU TYPE
           ;; 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
           ;; 11/70=06,PDQ=07,Q=10
           ;; BIT 10=REAL TIME CLOCK
           ;; BIT 9=FLOATING POINT PROCESSOR
           ;; BIT 8=MEMORY MANAGEMENT
$ETEND:
.MEXIT
```

.SBTTL ERROR POINTER TABLE

```

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

```

```

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

```

```

261
262
263
264
265
266
267
268
269
270
271
272
273
274
275 001150      $ERRTB:
276
277
278 001150 064564 ;ITEM 1      EM1      ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
279 001152 000000      0
280 001154 067720      DT1      ;$REG4, $REG3, $REG2, $REG1,$REG0,$ERRPC,$REG5,$REG6
281 001156 000000      0
282
283 001160 064564 ;ITEM 2      EM2      ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
284 001162 065005      DH2      ;$REG4, $REG3, $REG2, $REG1, $REG0,$ERRPC,$REG5,$REG6
285 001164 067720      DT2      ;$REG4, $REG3, $REG2, $REG1, $REG0,$ERRPC,$REG5,$REG6
286 001166 000000      0
287
288 001170 064737 ;ITEM 3      EM3      ;S/B SP WAS SP (IR) TEST (PC) (PSW)
289 001172 000000      0
290 001174 067742      DT3      ;$REG4, $REG3, $REG1,$REG0,$ERRPC,$REG6
291 001176 000000      0
292
293 001200 064564 ;ITEM 4      EM4      ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
294 001202 065016      DH4      ;$REG4, $REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
295 001204 067720      DT4      ;$REG4, $REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
296 001206 000000      0
297
298 001210 064612 ;ITEM 5      EM5      ;(IR) TEST (PC) (SP) (PSW)
299 001212 000000      0
300 001214 067726      DT5      ;$REG1, $REG0, $ERRPC, $REG5, $REG6
301 001216 000000      0
302
303 001220 064604 ;ITEM 6      EM6      ; DEST (IR) TEST (PC) (SP) (PSW)
304 001222 000000      0
305 001224 067724      DT6      ;$REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
306 001226 000000      0
307
308 001230 064574 ;ITEM 7      EM7      ;WAS DST DEST (IR) TEST (PC) (SP) (PSW)
309 001232 000000      0
310 001234 067722      DT7      ;$REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
311 001236 000000      0
312
313 001240 064650 ;ITEM 10     EM10     ;S/B RES WAS RES DST OP STC OP TEST (PC) (SP) (PSW)
314 001242 000000      0
315 001244 067720      DT10     ;$REG4, $REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
316 001246 000000      0

```



```

317
318 001250 065215
319 001252 065233
320 001254 067760
321 001256 000000
322
323
324 001260 065271
325 001262 067327
326 001264 067620
327 001266 000000
328
329
330 001270 065330
331 001272 067327
332 001274 067620
333 001276 000000
334
335
336 001300 065362
337 001302 067327
338 001304 067620
339 001306 000000
340
341
342 001310 065402
343 001312 067240
344 001314 067564
345 001316 067712
346
347
348
349 001320 066771
350 001322 067543
351 001324 067624
352 001326 000000
353
354
355
356 001330 066742
357 001332 067543
358 001334 067624
359 001336 000000
360
361
362
363 001340 065641
364 001342 067543
365 001344 067624
366 001346 000000
367
368
369 001350 065471
370 001352 067272
371 001354 067574
372 001356 067714

;ITEM 11
EM11 ;TESTS SKIPPED
DH11 ;PC EXPCTD ACTUAL (TEST #'S)
DT11 ;SERRPC, $TESTN,$REGO
0

;ITEM 12
EM12 ;MED DID NOT ABORT IN USER MODE
DH23 ;PC
DT23 ;SERRPC
0

;ITEM 13
EM13 ;MED EXECUTED IN USER MODE
DH23 ;PC
DT23 ;SERRPC
0

;ITEM 14
EM14 ;MED CHANGED PSW
DH23 ;PC
DT23 ;SERRPC
0

;ITEM 15
EM15 ;MICROBREAK TRAP-TO-4 DID NOT OCCUR
DH15 ;ERRPC MEDCODE MICROBK REG.
DT15 ;SERRPC,$TMP0,$TMP1,0
DF15 ;0,0

;ITEM 16
EM16 ;CACHE DATA LOGGED INCORRECTLY
DH44 ;PC EXPCT RECVD
DT24 ;SERRPC,$REG1,$REG0,0
0

;ITEM 17
EM45 ;CACHE TAG LOGGED WRONG
DH44 ;PC EXPCT RECVD
DT24 ;SERRPC,$REG0,$REG1,0
0

;ITEM 20
EM26 ;PHYS. BA LOGGED WRONG
DH44 ;PC EXPCT RECVD
DT24 ;SERRPC,$REG1,$REG0,0
0

;ITEM 21
EM21 ;CSP CONSTANT WRONG
DH17 ;PC MEDCODE EXPECTD RECEIVD
DT21 ;SERRPC,$TMP1,$TMP2,$REG0,0
DF17 ;0,0,0

```

373				
374				
375	001360	065514		
376	001362	067272		
377	001364	067606		
378	001366	067714		
379				
380				
381	001370	065543		
382	001372	067327		
383	001374	067620		
384	001376	000000		
385				
386				
387				
388	001400	065562		
389	001402	067334		
390	001404	067624		
391	001406	000000		
392				
393				
394				
395	001410	065445		
396	001412	067543		
397	001414	067624		
398	001416	000000		
399				
400				
401				
402	001420	065641		
403	001422	067373		
404	001424	067642		
405	001426	000000		
406				
407				
408				
409	001430	065666		
410	001432	067455		
411	001434	067656		
412	001436	000000		
413				
414				
415				
416	001440	065736		
417	001442	067327		
418	001444	067620		
419	001446	000000		
420				
421				
422				
423	001450	066541		
424	001452	067327		
425	001454	067620		
426	001456	000000		
427				
428				

				: ITEM 22
	EM22			:BAD DATA READ BY A MED
	DH17			:PC MEDCODE EXPECTD RECEIVD
	DT22			:SERRPC,\$TMP1,\$TMP2,\$TMP3,0
	DF17			:0,0,0
				: ITEM 23
	EM23			:NO ODD PC TRAP
	DH23			:PC
	DT23			:SERRPC
	0			
				: ITEM 24
	EM24			:ODD ADR. BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
	DH24			:PC CPUERR LOGJAM
	DT24			:SERRPC,\$REG1,\$REG0
	0			
				: ITEM 25
	EM17			:LOG CUA LOGGED INCORRECT U-ADDR
	DH44			:PC EXPCTD RECVD
	DT24			:SERRPC \$REG1 \$REG0
	0			
				: ITEM 26
	EM26			:PHYS. BA LOGGED WRONG
	DH26			:PC PA<17:16>-EXPCT-PA<15:0> PA<17:16>-RECVD-PA<15:0>
	DT26			:SERRPC,\$REG1,\$REG2,\$REG0,\$REG3,0
	0			
				: ITEM 27
	EM27			:CACHE PARITY ERROR LOGGED IN BACK UP MODE
	DH27			:PC LOGPBA LOGDATA LOGTAG
	DT27			:SERRPC,\$REG3,\$REG1,\$REG2
	0			
				: ITEM 30
	EM30			:CACHE PARITY TRAPPED WHEN DISABLED
	DH23			:PC
	DT23			:SERRPC
	0			
				: ITEM 31
	EM31			:NO CACHE PARITY TRAP
	DH23			:PC
	DT23			:SERRPC
	0			
				: ITEM 32

H03

429				
430	001460	066050	EM32	: MEMORY ERROR REGISTERS INCORRECT
431	001462	067510	DH32	: PC MEMERR
432	001464	067634	DT25	: SERRPC, \$REG0
433	001466	000000	0	
434				
435				: ITEM 33
436				
437	001470	066101	EM33	: TIMEOUT BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
438	001472	067334	DH24	: PC CPUERR LOGJAM
439	001474	067624	DT24	: SERRPC, \$REG1, \$REG0
440	001476	000000	0	
441				
442				: ITEM 34
443				
444	001500	066157	EM34	: NO ILLEGAL INTERNAL ADDRESS TRAP
445	001502	067327	DH23	: PC
446	001504	067620	DT23	: SERRPC
447	001506	000000	0	
448				
449				: ITEM 35
450				
451	001510	066214	EM35	: INTERNAL ADDRESS ERROR BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
452	001512	067334	DH24	: PC CPUERR LOGJAM
453	001514	067624	DT24	: SERRPC, \$REG1, \$REG0
454	001516	000000	0	
455				
456				: ITEM 36
457				
458	001520	066302	EM36	: LAST INTERRUPT/TRAP VECTOR NOT LOGGED IN FLAG REGISTER
459	001522	067357	DH25	: PC FLGREG
460	001524	067634	DT25	: SERRPC, \$REG0
461	001526	000000	0	
462				
463				: ITEM 37
464				
465	001530	066357	EM37	: LOG FIRST MODE DID NOT INHIBIT ERROR LOG AFTER FIRST ERROR
466	001532	067334	DH24	: PC CPUERR LOGJAM
467	001534	067624	DT24	: SERRPC, \$REG1, \$REG0
468	001536	000000	0	
469				
470				: ITEM 40
471				
472	001540	066452	EM40	: ERROR LOG WAS NOT RE-ENABLED, ODD ADR BIT CLR IN CPUERR
473	001542	067334	DH24	: PC CPUERR LOGJAM
474	001544	067624	DT24	: SERRPC, \$REG1, \$REG0
475	001546	000000	0	
476				
477				: ITEM 41
478				
479	001550	066001	EM41	: INSTRUCTION NOT ABORTED IN CACHE ABORT MODE
480	001552	067327	DH23	: PC
481	001554	067620	DT23	: SERRPC
482	001556	000000	0	
483				
484				: ITEM 42

```

485
486 001560 066566 EM42 ;LO BYTE & TAG PARITY BITS NOT SET IN LOG SERVICE
487 001562 067524 DH42 ;PC LOGSERVICE
488 001564 067634 DT25 ;SERRPC,$REG0,0
489 001566 000000 0
490
491 ;ITEM 43
492
493 001570 066654 EM43 ;LO BYTE & TAG PARITY BITS NOT SET IN MEM ERR REGISTER
494 001572 067510 DH32 ;PC MEMERR
495 001574 067634 DT25 ;SERRPC,$REG0
496 001576 000000 0
497
498 ;ITEM 44
499
500 001600 067021 EMEIS1 ;EIS SET COND CODES WRONG
501 001602 067156 DHEIS1 ; PSW REG-WAS-REG+1 REG-S/B-REG+1 PC TEST (IR)
502 001604 067670 DTEIS1 ;$REGAD $REG2 $REG3 $REG1 $REG4 $ERRPC $REG0 $TMPO
503 001606 000000 0
504
505 ;ITEM 45
506
507 001610 067052 EMEIS2 ;EIS GAVE WRONG RESULT
508 001612 067156 DHEIS1 ; PSW REG-WAS-REG+1 REG-S/B-REG+1 PC TEST (IR)
509 001614 067670 DTEIS1 ;$REGAD $REG2 $REG3 $REG1 $REG4 $ERRPC $REG0 $TMPO
510 001616 000000 0
511
512 ;ITEM 46
513
514 001620 067100 EM46 ;AUTO-INCREMENT (DECREMENT) DID NOT OCCUR
515 001622 067217 DH46 ; PC (IR) TEST
516 001624 067702 DT46 ;SERRPC $TMPO $REG0
517 001626 000000 0
518
519 076600 MED = 076600
520 140000 UM= 140000
521 177770 UBREAK= 177770
522 177744 MEMERR=177744
523 177766 CPUERR=177766
524 177746 CCR=177746
525 000100 WWP=BIT6
526 000001 DPTRP=BIT0
527 000200 PABORT=BIT7
528 000100 LO=BIT6
529 000200 HI=BIT7
530 000040 TAG=BITS
531
532 .EQUIV SP,KSP
533
534
535 ;* MED OPERATION CODE DEFINITIONS
536
537 000226 WCNSSW=226
538 000022 RDWHAMI=022
539 000222 WRWHAMI=222
540 000144 RDFLAG=144
    
```

541	000344	WRFLAG=344
542	000100	RDLJAM=100
543	000300	WRLJAM=300
544	000101	RDLSERVICE=101
545	000301	WRLSERVICE=301
546	000102	RDLPBA=102
547	000302	WRLPBA=302
548	000103	RDLCUA=103
549	000303	WRLCUA=303
550	000104	RDLFGINT=104
551	000304	WRLFGINT=304
552	000105	RDLWHAMI=105
553	000305	WRLWHAMI=305
554	000106	RDLDATA=106
555	000306	WRLDATA=306
556	000107	RDLTAG=107
557	000307	WRLTAG=307
558	000071	SWB01=71

;MICRO ADDR. IN SWAB INST.

;ADDRESS ASSIGNMENTS FOR DL11 CONSOLE TERMINAL INTERFACE

564	177560	RCSR=177560	;RCVR. CONTROL / STATUS REG. ADDRESS
565	177562	RDBR = 177562	;RECEIVER DATA BUFFER REG. ADDR.
566	177564	XCSR = 177564	;TRANSMITTER CONTROL / STATUS REG. ADDR
567	177566	XDBR = 177566	;TRANSMIT DATA BUFFER REG. ADDR.
568	177546	LKCSR= 177546	;LINE CLOCK ADDRESS
569			

//////////
"BCPT" TESTS
//////////

570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625

001630
001630 000401
001632 000000

001634 000402
001636 000403
001640 000000
001642 000775
001644 000000

001646 100403
001650 001402
001652 102401
001654 103002

001656 000000
001660 000772

001662 000277

001664 100003
001666 001002
001670 102001
001672 103402

001674 000000
001676 000771

```

; *****
; .SBTTL BT001 "BR" TEST - POSITIVE OFFSET
; *****
START:
BT001: BR      BT002      ;TEST THE BR FORWARD
E001:  HALT          ;BR FAILED TO LOAD PC PROPERLY
; *****
; .SBTTL BT002 "BR" TEST - NEGATIVE OFFSET
; *****
BT002: BR      I002      ;GO TO TEST INSTRUCTION
A002:  BR      BT003      ;GO TO NEXT TEST
EX002: HALT          ;JUST IN CASE
I002:  BR      A002      ;TEST THE BR - NEG. OFFSET
E2002: HALT          ;BR FAILED WITH NEG. OFFSET
; *****
; .SBTTL BT003 "BASIC COND. BR" TEST - FLAGS CLEARED
; *****
BT003: BMI      E003      ;BR IF "N" SET
      BEQ      E003      ;BR IF "Z" SET
      BVS      E003      ;BR IF "V" SET
      BCC      BT004      ;BR IF "C" CLEAR
E003:  HALT          ;ERROR - ONE OF THE ABOVE BR'S FAILED
      BR      BT003      ;OR THE FLAGS FAILED TO CLEAR ON "START"
      ;LOCK ON HARD ERROR
; *****
; .SBTTL BT004 "SCC AND COND. BR'S" TEST - FLAGS SET
; *****
BT004: SCC          ;MAKE N:C=1111
I004:  BPL      E004      ;BR IF "N" FAILED TO SET
      BNE      E004      ;BR IF "Z" FAILED TO SET
      BVC      E004      ;BR IF "V" FAILED TO SET
      BCS      BT005      ;BR IF "C" SET OK
E004:  HALT          ;ERROR - ONE OF THE ABOVE BR'S FAILED
      BR      BT004      ;OR THA SCC FAILED TO SET ALL THE FLAGS
      ;LOCK ON HARD ERROR
; *****
```


M03

```

682
683 ; *****
684 .SBTTL BT011 "COM %R AND ADC %R" TEST
685 ; *****
686
687 001762 005000 BT011: CLR RO ;MAKE [RO] = 000000
688 001764 000257 CCC ;MAKE N:C=0000
689
690 001766 005100 I011: COM RO ;TEST THE COM - [RO] S/B = 177777
691 001770 005500 ADC RO ;TEST THE ADC - [RO] S/B = 000000
692
693 001772 001001 BNE E011 ;BR IF "Z" DID NOT SET
694 001774 103402 BCS BT012 ;BR IF "C" SET OK
695
696 001776 000000 E011: HALT ;ERROR - COM OR ADC FAILED
697 002000 000770 BR BT011 ;LOCK ON HARD ERROR
698
699 ; *****
700 .SBTTL BT012 "MOV #N,R" TEST WITH N=177777,[R]=000000
701 ; *****
702
703 002002 005000 BT012: CLR RO ;MAKE [RO] = 000000
704 002004 000257 CCC ;MAKE N:C=0000
705
706 002006 012700 177777 I012: MOV #-1,RO ;TEST THE MOV - [RO] S/B = 177777
707
708 002012 005100 COM RO ;MAKE [RO] = 000000
709 002014 001402 BEQ BT013 ;BR IF "Z" SET
710
711 002016 000000 E012: HALT ;ERROR - MOV FAILED TO LOAD RO WITH ALL 1'S
712 002020 000770 BR BT012 ;LOCK ON HARD ERROR
713
714 ; *****
715 .SBTTL BT013 "MOV #N,R" TEST WITH N=000000,[R]=177777
716 ; *****
717
718 002022 005000 BT013: CLR RO ;MAKE [RO] = 000000
719 002024 005100 COM RO ;MAKE [RO] = 177777
720 002026 000257 CCC ;SCOPE SYNC
721
722 002030 012700 000000 I013: MOV #0,RO ;TEST THE MOV - [RO] S/B = 000000
723
724 002034 005100 COM RO ;MAKE [RO] = 177777, SET "C"
725 002036 005500 ADC RO ;MAKE [RO] = 000000
726 002040 001402 BEQ BT014 ;BR IF "Z" GOT SET
727
728 002042 000000 E013: HALT ;ERROR - MOV FAILED TO CLEAR RO
729 002044 000766 BR BT013 ;LOCK ON HARD ERROR
730
731 ; *****
732 .SBTTL BT014 "CLR (R)" TEST - [R] = 177776
733 ; *****
734
735 002046 012706 001000 BT014: MOV #STACK,SP ;SET UP STACK POINTER
736 002052 012700 177776 MOV #PSW,RO ;RO POINTS TO PSW
737 002056 000277 SCC ;MAKE [PSW] = 017
    
```



```

738
739 002060 005010      I014:  CLR      (RO)          ;TEST THE CLR - IT SHOULD CLEAR PSW
740
741 002062 001002      BNE      BT015          ;BR IF CLR MADE "Z" = 0 - IT SHOULD
742
743 002064 000000      E014:  HALT          ;ERROR- CLR FAILED TO CLEAR PSW
744 002066 000767      BR      BT014          ;LOCK ON HARD ERROR
745
746      ; *****
747      .SBTTL BT015 "CLR (R)+" TEST - [R] = 177776
748      ; *****
749
750 002070 012700 177776 BT015:  MOV      #PSW,RO      ;RO POINTS TO PSW
751 002074 000277      SCC          ;MAKE [PSW] = 017
752
753 002076 005020      I015:  CLR      (RO)+      ;TEST THE CLR - IT SHOULD CLEAR PSW
754
755 002100 001002      BNE      A015          ;BR IF CLR MADE "Z" = 0 - IT SHOULD
756
757 002102 000000      E1015A: HALT          ;ERROR- CLR FAILED TO CLEAR PSW
758 002104 000771      BR      BT015          ;LOCK ON HARD ERROR
759
760 002106 005700      A015:  TST      RO          ;AUTO INC SHOULD ZERO RO
761
762 002110 001402      BEQ      BT016          ;BR IF IT DID
763
764 002112 000000      E2015: HALT          ;ERROR - AUTOINC. FAILED
765 002114 000765      BR      BT015          ;LOCK ON HARD ERROR
766
767      ; *****
768      .SBTTL BT016 "COM (R)" TEST - [R] = 177776
769      ; *****
770
771 002116 012700 177776 BT016:  MOV      #PSW,RO      ;RO POINTS TO PSW
772 002122 000257      CCC          ;MAKE [PSW] = 000
773
774 002124 005110      I016:  COM      (RO)          ;TEST THE COM - [PSW] S/B = 357
775
776 002126 100003      BPL      E016          ;N:C=1111 ?
777 002130 001002      BNE      E016
778 002132 102001      BVC      E016
779 002134 103402      BCS      BT017
780
781 002136 000000      E016:  HALT          ;ERROR - COM FAILED TO MAKE [PSW] = 357
782 002140 000766      BR      BT016          ;LOCK ON HARD ERROR
783
784      ; *****
785      .SBTTL BT017 "COM (RO)+" TEST - [RO] = 177776
786      ; *****
787
788 002142 012700 177776 BT017:  MOV      #PSW,RO      ;RO POINTS TO PSW
789 002146 005010      CLR      (RO)          ;MAKE [PSW] = 000
790 002150 000257      CCC          ;SCOPE SYNC
791
792 002152 005120      I017:  COM      (RO)+      ;TEST THE COM - [PSW] S/B = 357
793
    
```

```

794 002154 100003          BPL      EA017          ;N:C = 1111 ?
795 002156 001002          BNE      EA017
796 002160 102001          BVC      EA017
797 002162 103402          BCS      A017
798
799 002164 000000          EA017:  HALT          ;COM FAILED TO SET ALL FLAGS
800 002166 000765          BR      BT017        ;LOCK ON HARD ERROR
801
802 002170 005100          A017:   COM      RO      ;SHOULD MAKE [RO] = 177777
803 002172 005500          ADC      RO      ;SHOULD MAKE [RO] = 000000
804 002174 001402          BEQ      BT020
805
806 002176 000000          E2017:  HALT          ;ERROR - COM FAILED TO AUTO INC. RO
807 002200 000760          BR      BT017        ;LOCK ON HARD ERROR
808
809          ; *****
810          ; .SBTTL BT020 "MOV RA,RB" TEST - WITH [RA]=177777,[RB]=000000
811          ; *****
812
813 002202 005000          BT020:  CLR      RO      ;MAKE [RO]=000000
814 002204 005001          CLR      R1      ;MAKE [R1]=000000
815 002206 005101          COM      R1      ;MAKE [R1]=0207777
816 002210 000257          CCC          ;SCOPE SYNC
817
818 002212 010100          I020:   MOV      R1,R0  ;TEST THE MOV
819
820 002214 100402          BMI      A020      ;BR IF "N" GOT SET
821
822 002216 000000          EA020:  HALT          ;ERROR-MOV FAILED TO SET "N"
823 002220 000770          BR      BT020      ;LOCK ON HARD ERROR
824
825 002222 005100          A020:   COM      RO      ;[RO] SHOULD GO TO 000000
826 002224 001402          BEQ      BT021      ;BR IF IT DID
827
828 002226 000000          E2020:  HALT          ;ERROR-MOV FAILED TO LOAD RO WITH 1'S
829 002230 000764          BR      BT020      ;LOCK ON HARD ERROR
830
831          ; *****
832          ; .SBTTL BT021 "MOV RA,RB" TEST WITH [RA]=000000,[RB]=177777
833          ; *****
834
835 002232 005000          BT021:  CLR      RO      ;MAKE [RO]=000000
836 002234 005100          COM      RO      ;MAKE [RO]=177777
837 002236 005001          CLR      R1      ;MAKE [R1]=000000
838 002240 000257          CCC          ;SCOPE SYNC
839
840 002242 010100          I021:   MOV      R1,R0  ;TEST THE MOV
841
842 002244 001402          BEQ      A021      ;BR IF "Z" GOT SET
843
844 002246 000000          EA021:  HALT          ;MOV FAILED TO SET "Z"
845 002250 000770          BR      BT021      ;LOCK ON HARD ERROR
846
847 002252 005100          A021:   COM      RO      ;SHOULD MAKE [RO]=177777 AND SET "C"
848 002254 005500          ADC      RO      ;SHOULD MAKE [RO]=000000
849 002256 001402          BEQ      BT022      ;BR IF "Z" SET

```

```

850
851 002260 000000      E2021:  HALT          ;MOV FAILED TO ZERO RO
852 002262 000763      BR          BT021      ;LOCK ON HARD ERROR
853
854 ; *****
855 ; .SBTTL BT022 "MOV #N, @#A" TEST WITH N=17, A=177776
856 ; *****
857
858 002264 000257      BT022:  CCC          ;MAKE [PSW]=000
859
860 002266 012737 000017 177776 I022:  MOV          #17, @#PSW ;TEST THE MOV
861
862 002274 100003      BPL          E022      ;N:C=1111
863 002276 001002      BNE          E022
864 002300 102001      BVC          E022
865 002302 103402      BCS          BT023
866
867 002304 000000      E022:  HALT          ;MOV FAILED TO LOAD PSW
868 002306 000766      BR          BT022      ;LOCK ON HARD ERROR
869
870 ; *****
871 ; .SBTTL BT023 "MOV RA, (RB)+" TEST WITH [RA]=17, [RB]=177776
872 ; *****
873
874 002310 012700 177776      BT023:  MOV          #PSW, RO ;RO POINTS TO PSW
875 002314 012701 000017      MOV          #17, R1 ;[SOURCE]=017
876 002320 000257      CCC          ;SCOPE SYNC - MAKE <N:C> = 0000
877
878 002322 010120      I023:  MOV          R1, (RO)+ ;TEST THE MOV
879
880 002324 100003      BPL          EA023     ;N:C = 1111 ?
881 002326 001002      BNE          EA023
882 002330 102001      BVC          EA023
883 002332 103402      BCS          A023
884
885 002334 000000      EA023:  HALT          ;MOV FAILED TO LOAD PSW
886 002336 000764      BR          BT023      ;LOCK ON HARD ERROR
887
888 002340 005700      A023:  TST          RO ;DID AUTO INC MAKE RO GO TO 0?
889 002342 001402      BEQ          BT024      ;BR IF IT DID
890
891 002344 000000      E2023:  HALT          ;MOV FAILED TO AUTO INC. RO
892 002346 000760      BR          BT023      ;LOCK ON HARD ERROR
893
894 ; *****
895 ; .SBTTL BT024 "CMP #N, @#A" TEST WITH N=(A)
896 ; *****
897
898 002350 012700 177776      BT024:  MOV          #PSW, RO ;RO POINTS TO PSW
899 002354 005010      CLR          (RO) ;MAKE [PSW]=000
900 002356 000273      273        ;MAKE N:C=1011
901
902 002360 022737 000013 177776 I024:  CMP          #13, @#PSW ;TEST THE CMP
903
904 002366 001402      BEQ          BT025      ;BR IF "Z" GOT SET
905

```

```

906 002370 000000      E024:  HALT                ;CMP FAILED TO SET "Z"
907 002372 000766      BR          BT024          ;LOCK ON HARD ERROR
908
909      ; *****
910      .SBTTL BT025 "CMP #N, @#A" WITH N > (A)
911      ; *****
912
913 002374 000257      BT025: CCC                ;MAKE [PSW]=000
914
915 002376 022737 000017 177776 I025:  CMP          #17, @#PSW    ;TEST THE CMP
916
917 002404 001401      BEQ          E025          ;BR IF "Z" GOT SET
918 002406 000402      BR          BT026          ;GO TO NEXT TEST
919
920 002410 000000      E025:  HALT                ;CMP FAILED TO CLEAR "Z"
921 002412 000770      BR          BT025          ;LOCK ON HARD ERROR
922
923      ; *****
924      .SBTTL BT026 "CMP #N, @#A" WITH N < (A)
925      ; *****
926 002414 000277      BT026: SCC                ;MAKE [PSW]=017
927
928 002416 022737 000000 177776 I026:  CMP          #0, @#PSW    ;TEST THE CMP
929
930 002424 001401      BEQ          E026          ;BR IF "Z" GOT SET
931 002426 000402      BR          BT027          ;GO TO NEXT TEST
932
933 002430 000000      E026:  HALT                ;CMP FAILED TO CLEAR "Z"
934 002432 000770      BR          BT026          ;LOCK ON HARD ERROR
935
936      ; *****
937      .SBTTL BT027 "CMP R, #N" TEST WITH [R]=N
938      ; *****
939
940 002434 012700 177777      BT027: MOV          #-1, R0    ;MAKE [R0]=177777
941 002440 000257      CCC                ;N:C=0000
942
943 002442 020027 177777      I027:  CMP          R0, #-1    ;TEST THE CMP
944
945 002446 001402      BEQ          BT030          ;BR IF CMP SET "Z"
946
947 002450 000000      E027:  HALT                ;CMP FAILED
948 002452 000770      BR          BT027          ;LOCK ON HARD ERROR
949
950      ; *****
951      .SBTTL BT030 "CMP R, #N" TEST WITH [R] > N
952      ; *****
953
954 002454 012700 000001      BT030: MOV          #1, R0    ;MAKE [R0]=000001
955 002460 000264      SEZ                ;SET THE "Z" BIT
956
957 002462 020027 177777      I030:  CMP          R0, #-1    ;TEST THE CMP
958
959 002466 001002      BNE          BT031          ;BR IF CMP CLEARED "Z"
960
961 002470 000000      E030:  HALT                ;CMP FAILED

```

E04

```

962 002472 000770          BR      BT030          ;LOCK ON HARD ERROR
963                          ; *****
964                          ; .SBTTL BT031 "CMP R,#N" TEST WITH [R] < N
965                          ; *****
966
967 002474 012700 000001    BT031:  MOV      #1,RO          ;MAKE [RO] = 000001
968 002500 000264          SEZ              ;SET THE "Z" BIT
969
970 002502 020027 000017    I031:  CMP      RO,#17         ;TEST THE CMP
971
972 002506 001002          BNE      BT032         ;BR IF CMP CLEARED "Z"
973
974 002510 000000          E031:  HALT     ;CMP FAILED TO SET "Z"
975 002512 000770          BR      BT031         ;LOCK ON HARD ERROR
976
977                          ; *****
978                          ; .SBTTL BT032 "CMP (RA)+,RB" TEST WITH [SOURCE]=[RB]
979                          ; *****
980
981 002514 012700 177776    BT032:  MOV      #PSW,RO        ;RO POINTS TO PSW
982 002520 012737 000340 177776  MOV      #340,2#PSW         ;MAKE [PSW]=340
983 002526 012701 000340    MOV      #340,R1           ;MAKE [DEST]=340
984 002532 000257          CCC              ;N:C=0000
985
986 002534 022001          I032:  CMP      (RO)+,R1       ;TEST THE CMP
987
988 002536 001402          BEQ      A032          ;BR IF "Z" GOT SET
989
990 002540 000000          EA032: HALT     ;CMP FAILED TO ACCESS PSW
991 002542 000764          BR      BT032         ;LOCK ON HARD ERROR
992
993 002544 005700          A032:  TST      RO           ;"Z" SHOULD SET
994 002546 001402          BEQ      BT033         ;BR IF "Z" SET
995
996 002550 000000          E2032: HALT     ;CMP FAILED TO AUTO INC. RO
997 002552 000760          BR      BT032         ;LOCK ON HARD ERROR
998
999                          ; *****
1000                          ; .SBTTL BT033 "CMP (RA)+,RB" TEST WITH [SOURCE]>[RB]
1001                          ; *****
1002
1003 002554 012700 177776    BT033:  MOV      #PSW,RO        ;RO POINTS TO PSW
1004 002560 012737 000340 177776  MOV      #340,2#PSW         ;MAKE [PSW]=340
1005 002566 012701 000330    MOV      #330,R1           ;MAKE [DEST]=330
1006 002572 000264          SEZ              ;SET THE "Z" BIT
1007
1008 002574 022001          I033:  CMP      (RO)+,R1       ;TEST THE CMP
1009
1010 002576 001002          BNE      A033          ;BR IF "Z" GOT CLEARED
1011
1012 002600 000000          EA033: HALT     ;CMP FAILED TO ACCESS PSW
1013 002602 000764          BR      BT033         ;LOCK ON HARD ERROR
1014
1015 002604 005700          A033:  TST      RO           ;"Z" SHOULD SET
1016 002606 001402          BEQ      BT034         ;BR IF "Z" SET
1017
    
```



```

1074 002716 012701 000017          MOV    #17,R1          ;MAKE [R1] = 000017
1075 002722 000264                SEZ                    ;SCOPE SYNC - SET "Z"
1076
1077 002724 020100          I037:  CMP    R1,R0          ;TEST THE CMP
1078
1079 002726 001002                BNE    BT040           ;BR IF "Z" GOT CLEARED
1080
1081 002730 000000          E037:  HALT                    ;ERROR - CMP FAILED TO SET "Z"
1082 002732 000770                BR     BT037           ;LOCK ON HARD ERROR
1083
1084 ; *****
1085 ; .SBTTL BT040 "MOV (RA),RB" TEST WITH [SOURCE]=[RB]=17
1086 ; *****
1087
1088 002734 012700 177776          BT040: MOV    #PSW,R0          ;RO POINTS TO PSW
1089 002740 005010                CLR    (R0)            ;MAKE [PSW]=000
1090 002742 005001                CLR    R1              ;MAKE [R1]=000000
1091 002744 000277                SCC                    ;MAKE N:C=1111
1092
1093 002746 011001          I040:  MOV    (R0),R1        ;TEST THE MOV
1094
1095 002750 020127 000017          CMP    R1,#17          ;DID R1 GET LOADED WITH 000017 ?
1096 002754 001402                BEQ    BT041           ;BR IF YES
1097
1098 002756 000000          E040:  HALT                    ;MOV FAILED TO LOAD R1
1099 002760 000765                BR     BT040           ;LOCK ON HARD ERROR
1100
1101 ; *****
1102 ; .SBTTL BT041 "MOV (RA)+,RB" TEST WITH [SOURCE]=[RB]=17
1103 ; *****
1104
1104 002762 012700 177776          BT041: MOV    #PSW,R0          ;RO POINTS TO PSW
1105 002766 005010                CLR    (R0)            ;MAKE [PSW]=000
1106 002770 005001                CLR    R1              ;MAKE [R1]=000000
1107 002772 000277                SCC                    ;MAKE N:C=1111
1108
1109 002774 012001          I041:  MOV    (R0)+,R1        ;TEST THE MOV
1110
1111 002776 020127 000017          CMP    R1,#17          ;DID R1 GET LOADED WITH 000017 ?
1112 003002 001402                BEQ    A041            ;BR IF YES
1113
1114 003004 000000          EA041: HALT                    ;MOV FAILED TO LOAD R1
1115 003006 000765                BR     BT041           ;LOCK ON HARD ERROR
1116
1117 003010 005700          A041:  TST    R0              ;"Z" SHOULD SET
1118 003012 001402                BEQ    BT042           ;BR IF "Z" GOT SET
1119
1120 003014 000000          E2041: HALT                    ;MOV FAILED TO AUTO INC. RO
1121 003016 000761                BR     BT041           ;LOCK ON HARD ERROR
1122
1123 ; *****
1124 ; .SBTTL BT042 "XOR RA,RB" TEST WITH [RA] = [RB] = 000000
1125 ; *****
1126
1127 003020 005000          BT042: CLR    R0              ;MAKE [R0] = 000000
1128 003022 005001                CLR    R1              ;MAKE [R1] = 000000
1129 003024 000257                CCC                    ;SCOPE SYNC
    
```

H04

```

1130
1131 003026 074100      I042:  XOR      R1,R0      ;TEST THE XOR
1132
1133 003030 005700      ;TST      RO      ;RESULT = 000000 ?
1134 003032 001402      BEQ      BT043      ;BR IF YES
1135
1136 003034 000000      E042:  HALT
1137 003036 000770      BR      BT042      ;XOR FAILED
1138
1139 ; *****
1140 ; .SBTTL BT043 "XOR RA,RB" TEST WITH [RA] = [RB] = 177777
1141 ; *****
1142
1143 003040 005000      BT043: CLR      RO      ;MAKE [RO] = 177777
1144 003042 005100      COM      RO
1145 003044 010001      MOV      RO,R1      ;MAKE [R1] = 177777
1146 003046 000257      CCC
1147 ;SCOPE SYNC
1148 003050 074100      I043:  XOR      R1,R0      ;TEST THE XOR
1149
1150 003052 005700      ;TST      RO      ;RESULT = 000000 ?
1151 003054 001402      BEQ      BT044      ;BR IF YES
1152
1153 003056 000000      E043:  HALT
1154 003060 000767      BR      BT043      ;XOR FAILED
1155 ;LOCK ON HARD ERROR
1156
1157 ; *****
1158 ; .SBTTL BT044 "XOR RA,RB" TEST WITH [RB]=052525,[RA]=125252
1159 ; *****
1160 003062 012701 125252      BT044: MOV      #125252,R1  ;MAKE [R1]=125252
1161 003066 012700 052525      MOV      #052525,R0  ;MAKE [R0]=052525
1162 003072 000257      CCC      ;SCOPE SYNC
1163
1164 003074 074100      I044:  XOR      R1,R0      ;TEST THE XOR
1165
1166 003076 020027 177777      CMP      RO,#-1      ;RESULT = 177777 ?
1167 003102 001402      BEQ      BT045      ;BR IF YES
1168
1169 003104 000000      E044:  HALT
1170 003106 000400      BR      BT045      ;XOR FAILED
1171 ;LOCK ON HARD ERROR
1172 ; *****
1173 ; .SBTTL BT045 "XOR RA,RB" TEST WITH [RA]=052525,[RB]=125252
1174 ; *****
1175 003110 012700 125252      BT045: MOV      #125252,R0  ;MAKE [R0]=125252
1176 003114 012701 052525      MOV      #052525,R1  ;MAKE [R1]=052525
1177 003120 000257      CCC      ;SCOPE SYNC
1178
1179 003122 074100      I045:  XOR      R1,R0      ;TEST THE XOR
1180
1181 003124 020027 177777      CMP      RO,#-1      ;RESULT = 177777 ?
1182 003130 001402      BEQ      BT046      ;BR IF YES
1183
1184 003132 000000      E045:  HALT
1185 003134 000765      BR      BT045      ;XOR FAILED
;LOCK ON HARD ERROR

```



```

1186
1187 ; *****
1188 .SBTTL BT046 GPR ADDRESS INTERRACTION TEST
1189 ; *****
1190
1191 003136 012700 125252 BT046: MOV #125252,R0 ;[R0] = 125252
1192 003142 010001 MOV R0,R1
1193 003144 005101 COM R1 ;[R1] = 052525
1194 003146 010102 MOV R1,R2
1195 003150 005102 COM R2 ;[R2] = 125252
1196 003152 010203 MOV R2,R3
1197 003154 005103 COM R3 ;[R3] = 052525
1198 003156 010304 MOV R3,R4
1199 003160 005104 COM R4 ;[R4] = 125252
1200 003162 010405 MOV R4,R5
1201 003164 005105 COM R5 ;[R5] = 052525
1202
1203 003166 074100 I046: XOR R1,R0 ;[R0] S/B = 177777
1204 003170 074200 XOR R2,R0 ;[R0] S/B = 125252
1205 003172 074300 XOR R3,R0 ;[R0] S/B = 177777
1206 003174 074400 XOR R4,R0 ;[R0] S/B = 125252
1207 003176 074500 XOR R5,R0 ;[R0] S/B = 177777
1208 003200 005100 COM R0 ;[R0] S/B = 000000
1209
1210 003202 001402 BEQ A046 ;BR IF [R0] WAS 000000
1211
1212 003204 000000 EA046: HALT ;GPR ADDRESSING PROBLEM
1213 003206 000753 BR BT046 ;LOCK ON HARD ERROR
1214
1215 003210 020627 001000 A046: CMP SP,#STACK ;DID R6 GET DISTURBED
1216 003214 001402 BEQ BASIC ;BR IF NOT
1217
1218 003216 000000 E2046: HALT ;R6 ADDRESS PROBLEM
1219 003220 000746 BR BT046 ;LOCK ON HARD ERROR
    
```

```

1220 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1221 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1222 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1223 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1224 003222 005037 063200 BASIC: CLR @ONCE ;SIGNAL PROGRAM HEADER TO BE PRINTED
1225 003226 005037 001012 CLR @SERTTL ;CLEAR ERROR COUNT FIRST TIME THROUGH
1226 003232 005037 001126 CLR @SPASS ;CLEAR PASS COUNT FIRST TIME THROUGH
1227 003236 012701 063162 INIT: MOV @PRIFLG,R1 ;SET UP TO INIT. COUNTERS AND FLAGS
1228 003242 005021 1S: CLR (R1)+ ;CLEAR ONE WORD
1229 003244 020127 063200 CMP R1,@ONCE ;CLEARED ALL FLAGS AND COUNTERS?
1230 003250 001374 BNE 1S ;BR IF NOT
1231 003252 012706 001000 MOV @STACK,SP ;SET UP THE STACK POINTER
1232 003256 012737 004030 177770 MOV @4030,@UBREAK ;SET SCOPE SYNC FOR COND CODE OPERATE
1233 003264 012737 177777 001074 MOV @-1,@SREGS ;FLAG CURRENT STACK POINTER TO BE TYPED
1234 ;IN FIRST ERROR CALL
1235
1236 ;*****
1237 ;*TEST 0 BASIC "BNE" TEST WITH Z=0
1238 ;*****
1239 003272 TST0:
1240 003272 012700 000000 1S: MOV @0,R0 ;:LOAD R0 WITH TEST NUMBER
1241 003276 000257 CCC ;MAKE Z=0
1242
1243 2S:
1244 003300 BNE TST1 ;:TEST THE BNE - IT SHOULD BR
1245
1246 3S:
1247 003302 000000 HALT ;BNE FAILED TO LOAD PC
1248 003304 000774 BR 1S ;LOCK ON HARD ERROR
1249
1250 ;*****
1251 ;*TEST 1 BASIC "BNE" TEST WITH Z=1
1252 ;*****
1253 003306 TST1:
1254 003306 012700 000001 1S: MOV @1,R0 ;:LOAD R0 WITH TEST NUMBER
1255 003312 000264 SEZ ;SET THE "Z" BIT
1256
1257 2S:
1258 003314 001001 BNE 3S ;TEST THE BNE - IT SHOULD NOT BR
1259
1260 3S:
1261 003316 000402 BR TST2 ;:GO TO NEXT TEST
1262
1263 3S:
1264 003320 000000 HALT ;BNE BRANCHED WITH Z=1
1265 003322 000773 BR 1S ;LOCK ON HARD ERROR
1266
1267 ;*****
1268 ;*TEST 2 BASIC "BEQ" TEST WITH Z=1
1269 ;*****
1270 003324 TST2:
1271 003324 012700 000002 1S: MOV @2,R0 ;:LOAD R0 WITH TEST NUMBER
1272 003330 000264 SEZ ;MAKE Z=1
1273
1274 2S:
1275 003332 BEQ TST3 ;:TEST THE BEQ - IT SHOULD BR
1276
1277 3S:
1278 003334 000000 HALT ;BEQ FAILED TO LOAD THE PC
1279 003336 000774 BR 1S ;LOCK ON HARD ERROR
1280 ;*****

```

K04

```

1276 ;*TEST 3 BASIC "BEQ" TEST WITH Z=0
1277 ;*****
1278 003340 TST3: MOV #3,R0 ;:LOAD R0 WITH TEST NUMBER
1279 003340 012700 000003 1S: CCC ;:MAKE Z=0
1280 003344 000257
1281
1282 003346 001401 2S: BEQ 3S ;:TEST THE BEQ - IT SHOULD NOT BR
1283
1284 003350 000402 BR TST4 ;:GO TO NEXT TEST
1285
1286 003352 000000 3S: HALT ;:BEQ BRANCHED WITH Z=0
1287 003354 000773 BR 1S ;:LOCK ON HARD ERROR
1288
1289 ;*****
1290 ;*TEST 4 BASIC "BPL" TEST WITH N=1
1291 ;*****
1292 003356 TST4: MOV #4,R0 ;:LOAD R0 WITH TEST NUMBER
1293 003356 012700 000004 1S: CLR #PSW ;:CLEAR THE PSW
1294 003362 005037 177776 SEN ;:MAKE N=1
1295 003366 000270
1296
1297 003370 100001 2S: BPL 3S ;:TEST THE BPL - IT SHOULDN'T BR
1298
1299 003372 000402 BR TST5 ;:GO TO NEXT TEST
1300
1301 003374 000000 3S: HALT ;:BPL BRANCHED WITH N=1
1302 003376 000771 BR 1S ;:LOCK ON HARD ERROR
1303
1304 ;*****
1305 ;*TEST 5 BASIC "BPL" TEST WITH N=0
1306 ;*****
1307 003400 TST5: MOV #5,R0 ;:LOAD R0 WITH TEST NUMBER
1308 003400 012700 000005 1S: CLR #PSW ;:CLEAR THE PSW
1309 003404 005037 177776 CCC ;:SCOPE SYNC
1310 003410 000257
1311
1312 003412 2S: BPL TST6 ;:TEST THE BPL - IT SHOULD BR
1313 003412 100002
1314
1315 003414 000000 3S: HALT ;:BPL FAILED TO LOAD THE PC
1316 003416 000772 BR 1S ;:LOCK ON HARD ERROR
1317
1318 ;*****
1319 ;*TEST 6 BASIC "MOV (RA),RB" TEST - (RA)=177776
1320 ;*****
1321 003420 TST6: MOV #6,R0 ;:LOAD R0 WITH TEST NUMBER
1322 003420 012700 000006 1S: MOV #PSW,R5 ;:SOURCE ADDR = 177776
1323 003424 012705 177776 CLR (R5) ;:MAKE [PSW]=000
1324 003430 005015 CLR R3 ;:[DEST] = 000000
1325 003432 005003 SCC ;:MAKE [PSW]=017
1326 003434 000277
1327
1328 003436 011503 2S: MOV (R5),R3 ;:TEST THE MOV
1329
1330 003440 020327 000017 3S: CMP R3,#17 ;:CORRECT RESULT ?
1331 003444 001402 BEQ TST7 ;:BR IF YES
  
```

```

1332
1333 003446 000000
1334 003450 000767
1335
1336
1337
1338 003452
1339 003452 012700 000007
1340 003456 012702 063236
1341 003462 012704 125252
1342 003466 012737 125252 063236 1S:
1343 003474 000257
1344
1345 003476 020412 2S:
1346
1347 003500 001402
1348
1349 003502 000000 3S:
1350 003504 000770
1351
1352
1353
1354 003506
1355 003506 012700 000010
1356 003512 012702 063236
1357 003516 012704 000001
1358 003522 005037 063236 1S:
1359 003526 000264
1360
1361 003530 020412 2S:
1362
1363 003532 001002
1364
1365 003534 000000 3S:
1366 003536 000771
1367
1368
1369
1370
1371 003540
1372 003540 012700 000011
1373 003544 012704 125252
1374 003550 010403
1375 003552 000257
1376
1377 003554 022703 125252 2S:
1378
1379 003560 001402
1380
1381 003562 000000 3S:
1382 003564 000771
1383
1384 003566 020403 4S:
1385 003570 001402
1386
1387 003572 000000 5S:

```

```

          HALT          ;ERROR-MOV FAILED
          BR            1S ;LOCK ON HARD ERROR
;*****
;TEST 7          BASIC "CMP RA,(RB)" TEST - [RA] = [DEST]
;*****
TST7:
          MOV          #7,R0          ;:LOAD R0 WITH TEST NUMBER
          MOV          #MBUF0,R2      ;:DEST ADDR = MBUF0
          MOV          #125252,R4     ;:RESULT S / B = 125252
1S:      MOV          #125252,@#MBUF0 ;:MAKE [DEST] = 125252
          CCC          ;:MAKE N:C=0000
2S:      CMP          R4,(R2)        ;:TEST THE CMP
          BEQ          TST10         ;;BR IF "Z" GOT SET
3S:      HALT          ;ERROR - CMP FAILED TO SET "Z"
          BR            1S ;LOCK ON HARD ERROR
;*****
;TEST 10         BASIC "CMP RA,(RB)" TEST - [RA] NOT EQUAL TO [DEST]
;*****
TST10:
          MOV          #10,R0         ;:LOAD R0 WITH TEST NUMBER
          MOV          #MBUF0,R2      ;:DEST ADDR = MBUF0
          MOV          #1,R4          ;:RESULT S / B = 000001
1S:      CLR          @#MBUF0        ;:MAKE [DEST] = 000000
          SEZ          ;:MAKE N:C=0100
2S:      CMP          R4,(R2)        ;:TEST THE CMP
          BNE          TST11         ;;BR IF "Z" GOT CLEARED
3S:      HALT          ;ERROR - CMP FAILED TO CLR "Z"
          BR            1S ;LOCK ON HARD ERROR
;*****
;TEST 11         BASIC "CMP #N,R" TEST - N = [R]
;*****
TST11:
          MOV          #11,R0         ;:LOAD R0 WITH TEST NUMBER
          MOV          #125252,R4     ;:RESULT S / B = 125252
1S:      MOV          R4,R3          ;:[DEST] = 125252
          CCC          ;:SCOPE SYNC
2S:      CMP          #125252,R3     ;:TEST THE CMP
          BEQ          4S            ;:BR IF N = [R]
3S:      HALT          ;CMP FAILED
          BR            1S ;LOCK ON HARD ERROR
4S:      CMP          R4,R3          ;:DID CMP ALTER [DEST]?
          BEQ          TST12         ;;BR IF NO
5S:      HALT          ;CMP DELIVERED A RESULT

```

```

1388 003574 000765          BR      1$          ;LOCK ON HARD ERROR
1389
1390
1391          ;*****
1392          ;#TEST 12      BASIC "CMP #N,R" TEST - N NOT EQUAL TO [R]
1393          ;*****
1394          TST12:
1394 003576 012700 000012      MOV     #12,R0          ;;LOAD R0 WITH TEST NUMBER
1395 003602 005004              CLR     R4              ;;RESULT S / B = 000000
1396 003604 010403          1$:    MOV     R4,R3          ;[DEST] = 125252
1397 003606 000264              SEZ              ;SCOPE SYNC
1398
1399 003610 022703 000001      2$:    CMP     #1,R3          ;TEST THE CMP
1400
1401 003614 001002              BNE     4$              ;BR IF N NOT EQUAL TO [R]
1402
1403 003616 000000          3$:    HALT              ;CMP FAILED
1404 003620 000771              BR      1$              ;LOCK ON HARD ERROR
1405
1406 003622 020403          4$:    CMP     R4,R3          ;DID CMP ALTER [DEST]?
1407 003624 001402              BEQ     TST13          ;;BR IF NO
1408
1409 003626 000000          5$:    HALT              ;CMP DELIVERED A RESULT
1410 003630 000765              BR      1$              ;LOCK ON HARD ERROR
1411
1412          ;*****
1413          ;#TEST 13      BASIC "MOV RA,(R)" TEST
1414          ;*****
1415          TST13:
1416 003632 012700 000013      MOV     #13,R0          ;;LOAD R0 WITH TEST NUMBER
1417 003636 012702 063236      MOV     #MBUFO,R2       ;;DEST ADDR=MBUFO
1418 003642 012704 177777      MOV     #-1,R4          ;;RESULT S / B = 177777
1419 003646 005012              CLR     (R2)           ;MAKE [DEST] = 000000
1420 003650 000257              CCC          ;SCOPE SYNC - N:C=0000
1421
1422 003652 010412          2$:    MOV     R4,(R2)          ;TEST THE MOV
1423
1424 003654 020412              CMP     R4,(R2)         ;RESULT CORRECT ?
1425 003656 001402              BEQ     TST14          ;;BR IF YES
1426
1427 003660 000000          3$:    HALT              ;ERROR - MOV FAILED
1428 003662 000771              BR      1$              ;LOCK ON HARD ERROR
1429
1430          ;*****
1431          ;#TEST 14      BASIC "MOV #N,(R)" TEST
1432          ;*****
1433          TST14:
1434 003664 012700 000014      MOV     #14,R0          ;;LOAD R0 WITH TEST NUMBER
1435 003670 012702 063236      MOV     #MBUFO,R2       ;;DEST ADDR = MBUFO
1436 003674 012704 177777      MOV     #-1,R4          ;;RESULT S / B = 177777
1437 003700 005012              CLR     (R2)           ;MAKE [DEST] = 000000
1438 003702 000257              CCC          ;SCOPE SYNC
1439
1440 003704 012712 177777          2$:    MOV     #-1,(R2)          ;TEST THE MOV
1441
1442 003710 020412              CMP     R4,(R2)         ;RESULT OK ?
1443 003712 001402              BEQ     TST15          ;;BR IF YES
    
```

N04

MAINDEC-11-DOKDA-A KD11-K BASIC LOGIC TESTS
 DOKDAA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 29
 BASIC "MOV #N,(R)" TEST

```

1444
1445 003714 000000
1446 003716 000770
1447
1448
1449
1450
1451 003720
1452 003720 012700 C00015
1453 003724 012704 177401
1454 003730 012702 063242
1455 003734 012705 063236
1456 003740 012712 177777
1457 003744 000257
1458
1459 003746 112765 000001 000004
1460
1461 003754 020412
1462 003756 001402
1463
1464 003760 000000
1465 003762 000766
1466
1467
1468
1469
1470 003764
1471 003764 012700 000016
1472 003770 012704 000777
1473 003774 012702 063242
1474 004000 012705 063236
1475 004004 012712 177777
1476 004010 000257
1477
1478 004012 112765 000001 000005
1479
1480 004020 020412
1481 004022 001402
1482
1483 004024 000000
1484 004026 000766
1485
1486
1487
1488
1489 004030
1490 004030 012700 000017
1491 004034 012702 063236
1492 004040 012704 000377
1493 004044 010412
1494 004046 000257
1495
1496 004050 005737 063236
1497
1498 004054 001401
1499 004056 100002
    
```

```

3S:  HALT           ;ERROR - MOV FAILED
      BR            1S ;LOCK ON HARD ERROR

;*****
;*TEST 15      BASIC "MOVB #N,X(R)" TEST - DEST EVEN
;*****
†ST15:
      MOV          #15,R0 ;:LOAD R0 WITH TEST NUMBER
      MOV          #177401,R4 ;:RESULT S / B = 177401
      MOV          #MBUF1,R2 ;:DEST ADDR = MBUF1
      MOV          #MBUFO,R5 ;:BASE DEST ADDR = MBUFO
1S:   MOV          #-1,(R2) ;:[DEST] = 177777
      CCC          ;SCOPE SYNC

2S:   MOVB        #1,4(R5) ;TEST THE MOVB

      CMP          R4,(R2) ;:RESULT OK?
      BEQ          TST16 ;:BR IF YES

3S:   HALT           ;MOVB DELIVERED WRONG RESULT
      BR            1S ;LOCK ON HARD ERROR

;*****
;*TEST 16      BASIC "MOVB #N,X(R)" TEST - DEST ODD
;*****
†ST16:
      MOV          #16,R0 ;:LOAD R0 WITH TEST NUMBER
      MOV          #777,R4 ;:RESULT S / B = 777
      MOV          #MBUF1,R2 ;:DEST ADDR = MBUF1
      MOV          #MBUFO,R5 ;:BASE DEST ADDR = MBUFO
1S:   MOV          #-1,(R2) ;:[DEST] = 177777
      CCC          ;SCOPE SYNC

2S:   MOVB        #1,5(R5) ;TEST THE MOVB

      CMP          R4,(R2) ;:RESULT OK?
      BEQ          TST17 ;:BR IF YES

3S:   HALT           ;MOVB DELIVERED WRONG RESULT
      BR            1S ;LOCK ON HARD ERROR

;*****
;*TEST 17      BASIC "TST @#A" TEST WITH [A] GT 0
;*****
†ST17:
      MOV          #17,R0 ;:LOAD R0 WITH TEST NUMBER
      MOV          #MBUFO,R2 ;:DEST ADDR = MBUFO
      MOV          #377,R4 ;:RESULT S / B = 377 (NO CHANGE)
1S:   MOV          R4,(R2) ;:[DEST] = 377
      CCC          ;SCOPE SYNC

2S:   TST         @#MBUFO ;TEST THE TST

      BEQ          3S ;BR IF "Z" SET - IT SHOULDN'T BE
      BPL          TST20 ;:BR IF "N" CLEAR - IT SHOULD BE
    
```

```

1500
1501 004060 000000      3$:  HALT                ;TST FAILED TO ALTER CODES PROPERLY
1502 004062 000770      BR          1$          ;LOCK ON HARD ERROR
1503      ;*****
1504      ;*TEST 20          BASIC "TST @#A" TEST WITH [A] LT 0
1505      ;*****
1506 004064
1507 004064 012700 000020  †ST20:  MOV      #20,R0        ;:LOAD R0 WITH TEST NUMBER
1508 004070 012702 063236  MOV      @MBUF0,R2      ;:DEST ADDR = MBUF0
1509 004074 012704 100000  MOV      #100000,R4     ;:MAKE S / B = 100000
1510 004100 010412      1$:  MOV      R4,(R2)      ;:MAKE [DEST] = 100000
1511 004102 000257      CCC                          ;:SCOPE SYNC
1512
1513 004104 005737 063236  2$:  TST      @MBUF0      ;TEST THE TST
1514
1515 004110 001401      BEQ      3$              ;BR IF "Z" SET - IT SHOULDN'T BE
1516 004112 100402      BMI      4$              ;BR IF "N" SET - IT SHOULD BE
1517
1518 004114 000000      3$:  HALT                ;TST FAILED TO ALTER CODES PROPERLY
1519 004116 000770      BR          1$          ;LOCK ON HARD ERROR
1520 004120 020412      4$:  CMP      R4,(R2)      ;:DID TST DISTURB [DEST] ?
1521 004122 001402      BEQ      †ST21         ;:BR IF NOT
1522
1523 004124 000000      5$:  HALT                ;TST DELIVERED A RESULT
1524 004126 000764      BR          1$          ;LOCK ON HARD ERROR
1525
1526      ;*****
1527      ;*TEST 21          BASIC "TST @#A" WITH [A] = 0
1528      ;*****
1529 004130
1530 004130 012700 000021  †ST21:  MOV      #21,R0        ;:LOAD R0 WITH TEST NUMBER
1531 004134 012702 063236  MOV      @MBUF0,R2      ;:DEST ADDR = MBUF0
1532 004140 005004      CLR      R4              ;:RESULT S / B = 0 (IT SHOULDN'T CHANGE
1533 004142 005012      1$:  CLR      (R2)        ;:[DEST] = 0
1534 004144 000257      CCC                          ;:SCOPE SYNC - Z=0
1535
1536 004146 005737 063236  2$:  TST      @MBUF0      ;TEST THE TST
1537
1538 004152 001402      BEQ      4$              ;BR IF TST SET "Z"
1539
1540 004154 000000      3$:  HALT                ;TST FAILED TO SET "Z"
1541 004156 000771      BR          1$          ;LOCK ON HARD ERROR
1542
1543 004160 020412      4$:  CMP      R4,(R2)      ;:[DEST] STILL = 000000
1544 004162 001402      BEQ      †ST22         ;:BR IF YES
1545
1546 004164 000000      5$:  HALT                ;TST ALTERED THE [DEST]
1547 004166 000765      BR          1$          ;LOCK ON HARD ERROR
1548
1549      ;*****
1550      ;*TEST 22          BASIC "BIT #N,@#A" WITH BIT SET IN "A"
1551      ;*****
1552 004170
1553 004170 012700 000022  †ST22:  MOV      #22,R0        ;:LOAD R0 WITH TEST NUMBER
1554 004174 012702 063236  MOV      @MBUF0,R2      ;:DEST ADDR = MBUF0
1555 004200 012704 040000  MOV      #40000,R4     ;:RESULT S / B = 40000
    
```

```

1556 004204 010412 1$: MOV R4,(R2) ;MAKE [DEST] = 40000
1557 004206 000277 SCC ;SCOPE SYNC - Z=1
1558
1559 004210 032737 040000 063236 2$: BIT #40000,@#MBOFO ;TEST THE BIT
1560
1561 004216 001002 BNE TST23 ;;BR IF Z=0 - IT SHOULD BE
1562
1563 004220 000000 3$: HALT ;BIT FAILED TO CLEAR "Z"
1564 004222 000770 BR 1$ ;LOCK ON HARD ERROR
1565
1566 ;*****
1567 ;*TEST 23 BASIC "BIT #N,@#A" WITH BIT CLEAR IN "A"
1568 ;*****
1569 TST23:
1570 004224 012700 000023 MOV #23,R0 ;:LOAD R0 WITH TEST NUMBER
1571 004230 012702 063236 MOV #MBOFO,R2 ;:DEST ADDR = MBOFO
1572 004234 005012 1$: CLR (R2) ;:MAKE [DEST] = 000000
1573 004236 000257 CCC ;:SCOPE SYNC - Z=0
1574
1575 004240 032737 040000 063236 2$: BIT #40000,@#MBOFO ;TEST THE BIT
1576
1577 004246 001402 BEQ 4$ ;BR IF Z=1 - IT SHOULD BE
1578
1579 004250 000000 3$: HALT ;BIT FAILED TO SET "Z"
1580 004252 000770 BR 1$ ;LOCK ON HARD ERROR
1581
1582 004254 005712 4$: TST (R2) ;DID BIT DELIVER A RESULT
1583 004256 001402 BEQ TST24 ;;BR IF NOT
1584
1585 004260 000000 5$: HALT ;BIT DISTURBED THE [DEST]
1586 004262 000764 BR 1$ ;LOCK ON HARD ERROR
1587
1588 ;*****
1589 ;*TEST 24 BASIC "TST (R)+ TEST
1590 ;*****
1591 TST24:
1592 004264 012700 000024 MOV #24,R0 ;:LOAD R0 WITH TEST NUMBER
1593 .SBTTL USER CONTROLLED BREAKPOINT -- BIT0
1594 004270 032737 000001 063160 BIT #BIT0,@#BPTLOC ;:BREAKPOINT HALT SET ??
1595 004276 001401 BEQ .+4 ;:BR IF NOT
1596 004300 000000 HALT ;:BREAK - DEPRESS CONTINUE TO RESTART
1597 004302 012702 063236 MOV #MBOFO,R2 ;:INITIAL DEST ADDR = MBOFO
1598 004306 005012 1$: CLR (R2) ;:MAKE [DEST] = 000000
1599 004310 000257 CCC ;:SCOPE SYNC
1600
1601 004312 005722 2$: TST (R2)+ ;TEST THE TST
1602
1603 004314 001402 BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE
1604
1605 004316 000000 3$: HALT ;TST FAILED TO SET "Z"
1606 004320 000772 BR 1$ ;LOCK ON HARD ERROR
1607
1608 004322 022702 063240 4$: CMP #MBOFO+2,R2 ;DID REG. GET AUTO-INCREMENTED ?
1609 004326 001402 BEQ TST25 ;;BR IF YES
1610
1611 004330 000000 5$: HALT ;TST FAILED TO UPDATE REGISTER
    
```



```

1612 004332 000765          BR      1S          ;LOCK ON HARD ERROR
1613
1614
1615
1616
1617 004334
1618 004334 012700 000025
1619 004340 012702 063254
1620 004344 012704 000377
1621 004350 012705 063256
1622 004354 000270
1623
1624 004356 005745
1625
1626 004360 100002
1627
1628 004362 000000
1629 004364 000771
1630
1631 004366 020502
1632 004370 001402
1633
1634 004372 000000
1635 004374 000765
1636
1637 004376 020412
1638 004400 001403
1639
1640 004402 000000
1641 004404 010412
1642 004406 000760
1643
1644
1645
1646
1647 004410
1648 004410 012700 000026
1649 004414 012702 063236
1650 004420 005004
1651 004422 005104
1652 004424 005012
1653 004426 000257
1654
1655 004430 005137 063236
1656
1657 004434 020412
1658 004436 001402
1659
1660 004440 000000
1661 004442 000770
1662
1663
1664
1665
1666 004444
1667 004444 012700 000027

```

```

;*****
;*TEST 25      BASIC "TST -(R)" TEST
;*****
TST25:
      MOV      #25,RO          ;;LOAD RO WITH TEST NUMBER
      MOV      #DWT+6,R2      ;;DEST ADDR = DWT+6
      MOV      #377,R4        ;;RESULT S / B = 377
1S:   MOV      #DWT+10,R5     ;;BASE DEST ADDR = DWT+10
      SEN
      TST      -(R5)          ;TEST THE TST
      BPL      4S            ;BR IF "N" CLEAR
3S:   HALT
      BR      1S            ;TST FAILED TO CLEAR "N"
                               ;LOCK ON HARD ERROR
4S:   CMP      R5,R2          ;DID DEST REG GET DECREMENTED?
      BEQ      6S            ;BR IF YES
5S:   HALT
      BR      1S            ;ERROR - TST FAILED TO UPDATE DEST REG
                               ;LOCK ON HARD ERROR
6S:   CMP      R4,(R2)        ;DID TST ALTER [DEST]?
      BEQ      TST26         ;;BR IF NOT
7S:   HALT
      MOV      R4,(R2)        ;TST ALTERED [DEST]
      BR      1S            ;RESTORE [DEST]
                               ;LOCK ON HARD ERROR
;*****
;*TEST 26      BASIC "COM @#A" TEST
;*****
TST26:
      MOV      #26,RO          ;;LOAD RO WITH TEST NUMBER
      MOV      #M#BUFO,R2     ;;DEST ADDR = M#BUFO
      CLR      R4              ;;RESULT S / B = 177777
1S:   COM      R4              ;MAKE [DEST] = 000000
      CLR      (R2)           ;SCOPE SYNC
      CCC
2S:   COM      @#M#BUFO ;TEST THE COM
      CMP      R4,(R2)        ;RESULT = 177777 ??
      BEQ      TST27         ;;BR IF YES
3S:   HALT
      BR      1S            ;COM DELIVERED THE WRONG RESULT
;*****
;*TEST 27      BASIC "INC @#A" TEST
;*****
TST27:
      MOV      #27,RO          ;;LOAD RO WITH TEST NUMBER

```

E05

```

1668 004450 012702 063236      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
1669 004454 012704 000100      MOV      #100,R4       ;RESULT S / B = 100
1670 004460 012712 000077      1$:     MOV      #77,(R2) ;[DEST] = 77
1671 004464 000257                CCC                    ;SCOPE SYNC
1672
1673 004466 005237 063236      2$:     INC      @#MBUFO ;TEST THE INC
1674
1675 004472 020412                CMP      R4,(R2)       ;DID RESULT = 100 ??
1676 004474 001402                BEQ      T$T30         ;;BR IF YES
1677
1678 004476 000000      3$:     HALT                    ;INC DELIVERED WRONG RESULT
1679 004500 000767                BR       1$            ;LOCK ON HARD ERROR
1680

```

```

;*****
;#TEST 30      BASIC "DEC RN" TEST
;*****

```

```

1683 T$T30:
1684 004502                MOV      #30,R0        ;:LOAD R0 WITH TEST NUMBER
1685 004502 012700 000030      1$:     MOV      #1,R3         ;[DEST] = +1
1686 004506 012703 000001      CCC                    ;SCOPE SYNC
1687 004512 000257
1688
1689 004514 005303      2$:     DEC      R3         ;TEST THE DEC
1690
1691 004516 005703                TST      R3            ;RESULT = 000000 ??
1692 004520 001402                BEQ      T$T31         ;;BR IF YES
1693
1694 004522 000000      3$:     HALT                    ;DEC DELIVERED THE WRONG RESULT
1695 004524 000770                BR       1$            ;LOCK ON HARD ERROR
1696

```

```

;*****
;#TEST 31      BASIC "DEC @#A" TEST
;*****

```

```

1697 T$T31:
1698 004526                MOV      #31,R0        ;:LOAD R0 WITH TEST NUMBER
1699 004526 012700 000031      1$:     MOV      #-1,R4       ;RESULT S / B = 177777
1700 004532 012704 177777      MOV      #MBUFO,R2    ;DEST ADDR = MBUFO
1701 004536 012702 063236      CLR      (R2)         ;MAKE [DEST] = 000000
1702 004542 005012                CCC                    ;SCOPE SYNC
1703 004544 000257
1704
1705 004546 005337 063236      2$:     DEC      @#MBUFO ;TEST THE DEC
1706
1707 004552 020412                CMP      R4,(R2)       ;DID RESULT = 177777 ??
1708 004554 001402                BEQ      T$T32         ;;BR IF YES
1709
1710 004556 000000      3$:     HALT                    ;DEC DELIVERED WRONG RESULT
1711 004560 000770                BR       1$            ;LOCK ON HARD ERROR
1712

```

```

;*****
;#TEST 32      BASIC "CLR X(R)" TESTS
;*****

```

```

1713 T$T32:
1714 004562                MOV      #32,R0        ;:LOAD R0 WITH TEST NUMBER
1715 004562 012700 000032      1$:     MOV      #MBUFO+2,R2 ;DEST ADDR = MBUFO+2
1716 004566 012702 063240      CLR      R4           ;RESULT S / B = 000000
1717 004572 005004                MOV      #MBUFO,R5    ;BASE DEST ADDR = MBUFO
1718 004574 012705 063236      MOV      #-1,(R2)     ;[DEST] = 177777
1719 004600 012712 177777

```

F05

```

1724 004604 000257          CCC          ;SCOPE SYNC
1725
1726 004606 005065 000002    2$: CLR      2(R5)      ;TEST THE CLR
1727
1728 004612 020412          CMP      R4.(R2)      ;RESULT = 0?
1729 004614 001402          BEQ      TST33        ;;BR IF YES
1730
1731 004616 000000    3$: HALT          ;CLR FAILED TO ZERO [DEST]
1732 004620 000765          BR       1$          ;LOCK ON HARD ERROR.
1733
1734          ;:*****
1735          ;*TEST 33      BASIC "ASL RN" TEST WITH [DEST]=125252 AND C(0)
1736          ;:*****
1737          TST33:
1738 004622 012700 000033    1$: MOV      #33,R0      ;:LOAD R0 WITH TEST NUMBER
1739 004626 012703 125252    MOV      #125252,R3    ;MAKE [DEST] = 125252
1740 004632 000257          CCC          ;MAKE C=0
1741
1742 004634 006303    2$: ASL      R3          ;TEST THE ASL - IT SHOULD SET "C"
1743
1744 004636 103402          BCS      4$          ;BR IF "C" GOT SET
1745
1746 004640 000000    3$: HALT          ;ASL FAILED TO SET "C" BIT
1747 004642 000771          BR       1$          ;LOCK ON HRD ERROR
1748
1749 004644 022703 052524    4$: CMP      #52524,R3  ;WAS RESULT = 52524 ??
1750 004650 001402          BEQ      TST34        ;;BR IF YES
1751
1752 004652 000000    5$: HALT          ;ASL DELIVERED THE WRONG RESULT
1753 004654 000764          BR       1$          ;LOCK ON HARD ERROR
1754
1755          ;:*****
1756          ;*TEST 34      BASIC "ASL RN" TEST WITH [DEST]=052525 AND C(1)
1757          ;:*****
1758          TST34:
1759 004656 012700 000034    1$: MOV      #34,R0      ;:LOAD R0 WITH TEST NUMBER
1760 004662 012703 052525    MOV      #052525,R3    ;MAKE [DEST] = 052525
1761 004666 000261          SEC          ;MAKE C=1
1762
1763 004670 006303    2$: ASL      R3          ;TEST THE ASL - IT SHOULD CLR "C"
1764
1765 004672 103002          BCC      4$          ;BR IF "C" GOT CLEARED
1766
1767 004674 000000    3$: HALT          ;ASL FAILED TO CLEAR "C"
1768 004676 000771          BR       1$          ;LOCK ON HARD ERROR
1769
1770 004700 022703 125252    4$: CMP      #125252,R3 ;RESULT = 125252 ??
1771 004704 001402          BEQ      TST35        ;;BR IF YES
1772
1773 004706 000000    5$: HALT          ;ASL DELIVERED WRONG REULT
1774 004710 000764          BR       1$          ;LOCK ON HARD ERROR
1775
1776          ;:*****
1777          ;*TEST 35      BASIC "ROL RN" TEST WITH [DEST]=125252 AND C(0)
1778          ;:*****
1779 004712 012700 000035    TST35: MOV      #35,R0      ;;LOAD R0 WITH TEST NUMBER
    
```

G05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 35
 BASIC "ROL RN" TEST WITH [DEST]=125252 AND C(0)

```

1780 004716 012703 125252 1$: MOV #125252,R3 ;MAKE [DEST] = 125252
1781 004722 000257 CCC ;MAKE C=0
1782
1783 004724 006103 2$: ROL R3 ;TEST THE ROL - IT SHOULD SET C
1784
1785 004726 103402 BCS 4$ ;BR IF "C" GOT SET
1786
1787 004730 000000 3$: HALT ;ROL FAILED TO SET "C"
1788 004732 000771 BR 1$ ;LOCK ON HARD ERROR
1789
1790 004734 022703 052524 4$: CMP #052524,R3 ;RESULT = 052524 ??
1791 004740 001402 BEQ TST36 ;;BR IF YES
1792
1793 004742 000000 5$: HALT ;ROL DELIVERED WRONG RESULT
1794 004744 000764 BR 1$ ;LOCK ON HARD ERROR
1795
1796 ;:*****
1797 ;*TEST 36 BASIC "ROL RN" TEST WITH [DEST]=052524 AND C(1)
1798 ;:*****
1799 TST36:
1800 004746 012700 000036 1$: MOV #36,R0 ;:LOAD R0 WITH TEST NUMBER
1801 004752 012703 -052524 MOV #052524,R3 ;MAKE [DEST] = 052524
1802 004756 000261 SEC ;MAKE C=1
1803
1804 004760 006103 2$: ROL R3 ;TEST THE ROL - IT SHOULD CLEAR C
1805
1806 004762 103002 BCC 4$ ;BR IF "C" IS CLEAR
1807
1808 004764 000000 3$: HALT ;ROL FAILED TO CLEAR "C"
1809 004766 000771 BR 1$ ;LOCK ON HARD ERROR
1810
1811 004770 022703 125251 4$: CMP #125251,R3 ;RESULT = 125251 ??
1812 004774 001402 BEQ TST37 ;;BR IF YES
1813
1814 004776 000000 5$: HALT ;ROL DELIVERED WRONG RESULT
1815 005000 000764 BR 1$ ;LOCK ON HARD ERROR
1816
1817 ;:*****
1818 ;*TEST 37 BASIS "TSTB (R)" TEST - EVEN ADDRESS
1819 ;:*****
1820 TST37:
1821 005002 012700 000037 MOV #37,R0 ;:LOAD R0 WITH TEST NUMBER
1822 005006 012702 063254 MOV #DWT+6,R2 ;DEST ADDR = DWT+6
1823 005012 012704 000377 MOV #377,R4 ;RESULT S / B = 377
1824 005016 000257 1$: CCC ;SCOPE SYNC
1825
1826 005020 105712 2$: TSTB (R2) ;TEST THE TSTB
1827
1828 005022 100402 BMI 4$ ;BR IF "N" SET - IT SHOULD BE
1829
1830 005024 000000 3$: HALT ;TSTB FAILED TO SET "N"
1831 005026 000773 BR 1$ ;LOCK ON HARD ERROR
1832
1833 005030 020412 4$: CMP R4,(R2) ;DID TSTB DISTURB [DEST]
1834 005032 001403 BEQ TST40 ;;BR IF NOT
1835
  
```

H05

```

1836 005034 000000
1837 005036 010412
1838 005040 000766
1839
1840
1841
1842 005042
1843 005042 012700 000040
1844 005046 012702 063764
1845 005052 012704 177401
1846 005056 012703 063765
1847 005062 000257
1848
1849 005064 105713
1850
1851 005066 100402
1852
1853 005070 000000
1854 005072 000773
1855
1856 005074 020412
1857 005076 001403
1858
1859 005100 000000
1860 005102 010412
1861 005104 000766
1862
1863
1864
1865
1866 005106
1867 005106 012700 000041
1868 005112 012702 063252
1869 005116 012704 177400
1870 005122 000257
1871
1872 005124 105737 063252
1873
1874 005130 001402
1875
1876 005132 000000
1877 005134 000772
1878
1879 005136 020412
1880 005140 001403
1881
1882 005142 000000
1883 005144 010412
1884 005146 000765
1885
1886
1887
1888
1889 005150
1890 005150 012700 000042
1891 005154 012702 063254

5$: HALT ;TSTB ALTERED (DEST)
MOV R4,(R2) ;RESTORE (DEST)
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 40 BASIS "TSTB (R)" TEST - ODD ADDRESS
;*****
TST40:
MOV #40,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #DWTB+6,R2 ;:DEST ADDR = DWTB+6
MOV #177401,R4 ;:RESULT S / B = 177401
MOV #DWTB+7,R3 ;:DEST ADDR USED = DWTB+7
1$: CCC ;SCOPE SYNC
2$: TSTB (R3) ;TEST THE TSTB
BMI 4$ ;BR IF "N" SET - IT SHOULD BE
3$: HALT ;TSTB FAILED TO SET "N"
BR 1$ ;LOCK ON HARD ERROR
4$: CMP R4,(R2) ;DID TSTB DISTURB (DEST)
BEQ TST41 ;:BR IF NOT
5$: HALT ;TSTB ALTERED (DEST)
MOV R4,(R2) ;RESTORE (DEST)
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 41 BASIC "TSTB @#A" TEST - EVEN ADDRESS
;*****
TST41:
MOV #41,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #DWTA+4,R2 ;:DEST ADDR = DWTA+4
MOV #177400,R4 ;:RESULT S / B = 177400
1$: CCC ;SCOPE SYNC
2$: TSTB @#DWTA+4 ;TEST THE TSTB
BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE
3$: HALT ;TSTB FAILED TO SET "Z"
BR 1$ ;LOCK ON HARD ERROR
4$: CMP R4,(R2) ;DID TSTB DISTURB (DEST)?
BEQ TST42 ;:BR IF NOT
5$: HALT ;TSTB ALTERED (DEST)
MOV R4,(R2) ;RESTORE (DEST)
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 42 BASIC "TSTB @#A" TEST - ODD ADDRESS
;*****
TST42:
MOV #42,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #DWTA+6,R2 ;:DEST ADDR = DWTA+6

```

```

1892 005160 012704 000377
1893 005164 000257
1894
1895 005166 105737 063255
1896
1897 005172 001402
1898
1899 005174 000000
1900 005176 000772
1901
1902 005200 020412
1903 005202 001403
1904
1905 005204 000000
1906 005206 010412
1907 005210 000765
1908
1909
1910
1911
1912 005212
1913 005212 012700 000043
1914 005216 010605
1915 005220 012704 177400
1916 005224 010506
1917 005226 005046
1918 005230 000257
1919
1920 005232 105366 000001
1921
1922 005236 020416
1923 005240 001402
1924
1925 005242 000000
1926 005244 000767
1927
1928 005246 010506
1929
1930
1931
1932
1933 005250
1934 005250 012700 000044
1935 005254 005003
1936 005256 000257
1937
1938 005260 013703 063222
1939
1940 005264 022703 063246
1941 005270 001402
1942
1943 005272 000000
1944 005274 000767
1945
1946
1947

```

```

MOV #377,R4 ;RESULT S / B = 377
CCC ;SCOPE SYNC
TSTB @#DWTA+7 ;TEST THE TSTB
BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE
HALT ;TSTB FAILED TO SET "Z"
BR 1$ ;LOCK ON HARD ERROR
CMP R4,(R2) ;DID TSTB DISTURB [DEST]?
BEQ TST43 ;;BR IF NOT
HALT ;TSTB ALTERED [DEST]
MOV R4,(R2) ;RESTORE [DEST]
BR 1$ ;LOCK ON HARD ERROR

```

```

*****
;TEST 43 BASIC "DECB 1(SP)"
*****
TST43:
MOV #43,R0 ;;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;;SAVE SP
MOV #177400,R4 ;RESULT S / B = 177400
1$: MOV R5,SP
CLR -(SP) ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: DECB 1(SP) ;TEST THE DECB
CMP R4,(SP) ;RESULT = 177400?
BEQ 4$ ;BR IF YES
3$: HALT ;ERROR - DECB FAILED
BR 1$ ;LOCK ON HARD ERROR
4$: MOV R5,SP ;RESET THE SP

```

```

*****
;TEST 44 BASIC "MOV @#A,R" TEST
*****
TST44:
MOV #44,R0 ;;LOAD R0 WITH TEST NUMBER
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: MOV @#ATA,R3 ;TEST THE MOV
CMP #DWTA,R3 ;RESULT = DWTA?
BEQ TST45 ;;BR IF YES
3$: HALT ;MOV FAILED TO DELIVER CORRECT RESULT
BR 1$ ;LOCK ON HARD ERROR

```

```

*****
;TEST 45 BASIC "MOV #N,X(R)" TEST

```

```

1948
1949 005276
1950 005276 012700 000045
1951 005302 012702 063240
1952 005306 012704 125252
1953 005312 012703 063236
1954 005316 005012
1955 005320 000257
1956
1957 005322 012763 125252 000002 2$: MOV #45,R0 ;:LOAD R0 WITH TEST NUMBER
1958
1959 005330 020412 CMP R4,(R2) ;:RESULT OK?
1960 005332 001402 BEQ TST46 ;:BR IF YES
1961
1962 005334 000000 3$: HALT ;:MOV DELIVERED WRONG RESULT
1963 005336 000765 BR 1$ ;:LOCK ON HARD ERROR
1964
1965
1966
1967
1968 005340
1969 005340 012700 000046
1970 005344 012703 063236
1971 005350 012704 125252
1972 005354 005013
1973 005356 000257
1974
1975 005360 012713 125252 2$: MOV #46,R0 ;:LOAD R0 WITH TEST NUMBER
1976
1977 005364 020413 CMP R4,(R3) ;:RESULT OK?
1978 005366 001402 BEQ TST47 ;:BR IF YES
1979
1980 005370 000000 3$: HALT ;:MOV DELIVERED WRONG RESULT
1981 005372 000770 BR 1$ ;:LOCK ON HARD ERROR
1982
1983
1984
1985
1986 005374
1987 005374 012700 000047
1988 005400 012705 063222
1989 005404 005003
1990 005406 000257
1991
1992 005410 012503 2$: MOV #47,R0 ;:LOAD R0 WITH TEST NUMBER
1993
1994 005412 022703 063246 CMP #ATA,R5 ;:SRC ADDR = ATA
1995 005416 000402 BR 4$ ;:[DEST] = 000000
1996
1997 005420 000000 3$: HALT ;:SCOPE SYNC
1998 005422 000766 BR 1$ ;:TEST THE MOV
1999
2000 005424 022705 063224 4$: CMP #DWA,R3 ;:RESULT OK?
2001 005430 001402 BEQ TST50 ;:BR IF YES
2002
2003 005432 000000 5$: HALT ;:MOV DELIVERED WRONG RESULT
;:LOCK ON HARD ERROR
;:DID SRC REG GET INCREMENTED?
;:MOV FAILED TO UPDATE SRC. REG.

```

K05

```

2004 005434 000761          BR      1$          ;LOCK ON HARD ERROR
2005
2006
2007
2008
2009 005436
2010 005436 012700 000050
2011 005442 012702 063242
2012 005446 012704 063246
2013 005452 005012
2014 005454 000257
2015
2016 005456 013737 063222 063242 2$:  MOV    @DATA,@MBUF1  ;TEST THE MOV
2017 005464 020412          CMP    R4,(R2)        ;DID RESULT = #DWTA ?
2018 005466 001402          BEQ    T$T51         ;;BR IF YES
2019
2020 005470 000000          3$:  HALT              ;MOV DELIVERED THE WRONG RESULT
2021 005472 000767          BR      1$          ;LOCK ON HARD ERROR
2022
2023
2024
2025
2026 005474
2027 005474 012700 000051
2028 005500 012705 005506
2029 005504 000257
2030
2031 005506 016507 000010          2$:  MOV    4$-2$(R5),PC  ;TEST THE MOV - GO TO NEXT TEST VIA 4$
2032
2033 005512 000000          3$:  HALT              ;MOV FAILED TO LOAD THE PC
2034 005514 000771          BR      1$          ;LOCK ON HARD ERROR
2035
2036 005516 005520          4$:  .+2              ;POINTER TO NEXT TEST
2037
2038
2039
2040
2041 005520
2042 005520 012700 000052
2043 005524 012704 063246
2044 005530 012702 063236
2045 005534 005012
2046 005536 000257
2047
2048 005540 013712 063222          2$:  MOV    @DATA,(R2)  ;TEST THE MOV
2049
2050 005544 020412          CMP    R4,(R2)        ;DID RESULT = #DWTA ??
2051 005546 001402          BEQ    T$T53         ;;BR IF YES
2052
2053 005550 000000          3$:  HALT              ;MOV DELIVERED WRONG RESULT
2054 005552 000770          BR      1$          ;LOCK ON HARD ERROR
2055
2056
2057
2058
2059 005554

```


L05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T53

MACY11 27(1006) 08-FEB-77 16:23 PAGE 40
 BASIC "MOV X(RA),RB" TEST

```

2060 005554 012700 000053      MOV      #53,R0      ;;LOAD R0 WITH TEST NUMBER
2061 005560 012705 063222      MOV      #ATA,R5    ;;[R5] = BASE ADDR FOR SOURCE (ATA)
2062 005564 005003              1$:     CLR      R3      ;;MAKE [DEST] = 000000
2063 005566 000257              CCC                      ;;SCOPE SYNC
2064
2065 005570 016503 000004      2$:     MOV      4(R5),R3  ;;TEST THE MOV
2066
2067 005574 022703 064554      CMP      #DBTA,R3   ;;RESULT = #DBTA ??
2068 005600 001402              BEQ      TST54      ;;BR IF YES
2069
2070 005602 000000      3$:     HALT                    ;;MOV DELIVERED WRONG RESULT
2071 005604 000767              BR       1$         ;;LOCK ON HARD ERROR
2072
2073      ;;*****
2074      ;;*TEST 54      BASIC "MOV RA,-(RB)" TEST
2075      ;;*****
2076 005606              TST54:
2077 005606 012700 000054      MOV      #54,R0      ;;LOAD R0 WITH TEST NUMBER
2078 005612 012702 063236      MOV      #MBUFO,R2   ;;FINAL DEST ADDR = MBUFO
2079 005616 012704 125252      MOV      #125252,R4  ;;RESULT S / B = 125252
2080 005622 012705 063240      1$:     MOV      #MBUFO+2,R5  ;;INITIAL DEST ADDR = TEMP2 + 2
2081 005626 005012              CLR      (R2)        ;;MAKE [DEST] = 000000
2082 005630 000257              CCC                      ;;SCOPE SYNC
2083
2084 005632 010445      2$:     MOV      R4,-(R5)  ;;TEST THE MOV
2085
2086 005634 020412      CMP      R4,(R2)    ;;RESULT = 125252
2087 005636 001402              BEQ      4$         ;;BR IF YES
2088
2089 005640 000000      3$:     HALT                    ;;MOV DELIVERED THE WRONG ESULT
2090 005642 000767              BR       1$         ;;LOCK ON HARD ERROR
2091
2092 005644 020205      4$:     CMP      R2,R5    ;;DID REGISTER GET DECREMENTED ?
2093 005646 001402              BEQ      TST55      ;;BR IF YES
2094
2095 005650 000000      5$:     HALT                    ;;MOV FAILED TO UPDATE REGISTER
2096 005652 000763              BR       1$         ;;LOCK ON HARD ERROR
2097
2098      ;;*****
2099      ;;*TEST 55      BASIC "MOV #A,-(R)" TEST
2100      ;;*****
2101 005654              TST55:
2102 005654 012700 000055      MOV      #55,R0      ;;LOAD R0 WITH TEST NUMBER
2103 005660 012704 063246      MOV      #DWTA,R4    ;;RESULT S / B = #DWTA
2104 005664 012702 063236      MOV      #MBUFO,R2   ;;DEST ADDR = MBUFO
2105 005670 012705 063240      1$:     MOV      #MBUFO+2,R5  ;;INITIAL DEST ADDR = MBUFO+2
2106 005674 005012              CLR      (R2)        ;;MAKE [DEST] = 000000
2107 005676 000257              CCC                      ;;SCOPE SYNC
2108
2109 005700 013745 063222      2$:     MOV      #ATA,-(R5)  ;;TEST THE MOV
2110
2111 005704 020412      CMP      R4,(R2)    ;;RESULT = 000000
2112 005706 001402              BEQ      4$         ;;BR IF YES
2113
2114 005710 000000      3$:     HALT                    ;;MOV DELIVERED THE WRONG RESULT
2115 005712 000766              BR       1$         ;;LOCK ON HARD ERROR
  
```

M05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 41
 BASIC "MOV @#A,-(R)" TEST

```

2116
2117 005714 020502
2118 005716 001402
2119
2120 005720 000000
2121 005722 000762
2122
2123
2124
2125
2126 005724
2127 005724 012700 000056
2128 005730 012702 063236
2129 005734 012704 063246
2130 005740 012705 063222
2131 005744 005012
2132 005746 000257
2133
2134 005750 011537 063236
2135
2136 005754 020412
2137 005756 001402
2138
2139 005760 000000
2140 005762 000770
2141
2142
2143
2144
2145 005764
2146 005764 012700 000057
2147 005770 012702 063236
2148 005774 012704 063246
2149 006000 012705 063224
2150 006004 005012
2151 006006 000257
2152
2153 006010 014537 063236
2154
2155 006014 020412
2156 006016 001402
2157
2158 006020 000000
2159 006022 000766
2160
2161 006024 022705 063222
2162 006030 001402
2163
2164 006032 000000
2165 006034 000761
2166
2167
2168
2169 006036
2170 006036 012700 000060
2171 006042 012705 063222
    
```

```

4S:  CMP      R5,R2      ;DID DEST REG GET DECREMENTED ??
      BEQ      TST56     ;;BR IF YES

5S:  HALT
      BR       1$        ;MOV FAILED TO UPDATE REGISTER
                        ;LOCK ON HARD ERROR

*****
;*TEST 56      BASIC "MOV (R),@#A" TEST
*****
TST56:
      MOV      #56,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      #MBUFO,R2 ;DEST ADDR = MBUFO
      MOV      #DWTA,R4  ;RESULT S / B = #DWTA
      MOV      #ATA,R5   ;SOURCE ADDR = ATA
1$:   CLR      (R2)       ;MAKE [DEST] = 000000
      CCC
                        ;SCOPE SYNC

2$:   MOV      (R5),@#MBUFO ;TEST THE MOV

      CMP      R4,(R2)   ;RESULT = #DWTA ??
      BEQ      TST57     ;;BR IF YES

3$:   HALT
      BR       1$        ;MOV DELIVERED THE WRONG RESULT
                        ;LOCK ON HARD ERROR

*****
;*TEST 57      BASIC "MOV -(R),@#A" TEST
*****
TST57:
      MOV      #57,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      #MBUFO,R2 ;DEST ADDR = MBUFO
      MOV      #DWTA,R4  ;RESULT S / B = #DWTA
1$:   MOV      #ATA+2,R5  ;INITIAL SOURCE ADDR = ATA+2
      CLR      (R2)       ;MAKE [DEST] = 000000
      CCC
                        ;SCOPE SYNC

2$:   MOV      -(R5),@#MBUFO ;TEST THE MOV

      CMP      R4,(R2)   ;RESULT = #DWTA ?
      BEQ      4$        ;BR IF YES

3$:   HALT
      BR       1$        ;MOV DELIVERED THE WRONG RESULT
                        ;LOCK ON HARD ERROR

4$:   CMP      #ATA,R5   ;DID THE SRC REG GET DECREMENTED ?
      BEQ      TST60     ;;BR IF YES

5$:   HALT
      BR       1$        ;MOV FAILED TO UPDATE SOURCE REG
                        ;LOCK ON HARD ERROR

*****
;*TEST 60      BASIC "MOV (RA),RB" TEST
*****
TST60:
1$:   MOV      #60,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      #ATA,R5   ;INITIAL SOURCE ADDR = ATA
    
```

N05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 42
 DQKDA.A.P11 08-FEB-77 16:17 T60 BASIC "MOV (RA),RB" TEST

```

2172 006046 005003          CLR      R3          ;MAKE [DEST] = 000000
2173 006050 000257          CCC
2174
2175 006052 012503          2$:     MOV      (R5)+,R3      ;TEST THE MOV
2176
2177 006054 022703 063246        CMP      #DWTB,R3      ;RESULT = #DWTB ?
2178 006060 001402          BEQ      4$            ;BR IF YES
2179
2180 006062 000000          3$:     HALT
2181 006064 000766          BR      1$            ;MOV DELIVERED WRONG RESULT
2182
2183 006066 022705 063224        4$:     CMP      #ATA+2,R5      ;DID SOURCE REG GET INCREMENTED
2184 006072 001402          BEQ      TST61        ;;BR IF YES
2185
2186 006074 000000          5$:     HALT
2187 006076 000761          BR      1$            ;MOV FAILED TO UPDATE SOURCE REGISTER
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
  
```

```

;*****
;#TEST 61      BASIC "MOV X(RA),RB" TEST
;*****
TST61:
1$:     MOV      #61,R0          ;;LOAD R0 WITH TEST NUMBER
        MOV      #ATA,R5        ;BASE SOURCE ADDR = ATA
        CLR      R3            ;MAKE [DEST] = 000000
        CCC
2$:     MOV      2(R5),R3      ;TEST THE MOV
        CMP      #DWTB,R3      ;RESULT = #DWTB ?
        BEQ      TST62        ;;BR IF YES
3$:     HALT
        BR      1$            ;MOV FAILED TO DELIVER CORRECT RESULT
        ;LOCK ON HARD ERROR
;*****
;#TEST 62      BASIC "MOV 2X(RA),RB" TEST
;*****
TST62:
1$:     MOV      #62,R0          ;;LOAD R0 WITH TEST NUMBER
        MOV      #DWTB+2,2#M#BUFO+2 ;SET UP ADDRESS TABLE M#BUFO
        MOV      #M#BUFO,R5     ;BASE ADDRESS IN R5
        CLR      R3            ;MAKE [DEST] = 000000
        CCC
2$:     MOV      22(R5),R3     ;TEST THE MOV
        CMP      #-1,R3        ;RESULT = 177777
        BEQ      TST63        ;;BR IF YES
3$:     HALT
        BR      1$            ;MOV DELIVERED THE WRONG RESULT
        ;LOCK ON HARD ERROR
;*****
;#TEST 63      BASIC "MOV (R)+,X(R)" TEST
;*****
TST63:
        MOV      #63,R0          ;;LOAD R0 WITH TEST NUMBER
  
```

```

2228 006176 012704 125252      MOV      #125252,R4      ;RESULT S / B = 125252
2229 006202 012702 063244      MOV      #MBUF1+2,R2    ;FINAL DEST ADDR = MBUF1+2
2230 006206 010437 063236      MOV      R4,#MBUFO      ;SOURCE OPERAND = 125252
2231 006212 012705 063236      1$:     MOV      #MBUFO,R5 ;[R5] = INITIAL SRC ADDR = MBUFO
2232 006216 005012          CLR      (R2)           ;MAKE [DEST] = 000000
2233 006220 000257          CCC                    ;SCOPE SYNC
2234
2235 006222 012565 000004      2$:     MOV      (R5)+,4(R5) ;TEST THE MOV
2236
2237 006226 020412          CMP      R4,(R2)        ;RESULT = 125252 ?
2238 006230 001402          BEQ      4$             ;BR IF YES
2239
2240 006232 000000      3$:     HALT                    ;MOV DELIVERED WRONG RESULT
2241 006234 000766          BR      1$             ;LOCK ON HARD ERROR
2242
2243 006236 022705 063240      4$:     CMP      #MBUFO+2,R5 ;DID REGISTER GET INCREMENTED ?
2244 006242 001402          BEQ      TST64         ;;BR IF YES
2245
2246 006244 000000      5$:     HALT                    ;MOV FAILED TO UPDATE REGISTER
2247 006246 000761          BR      1$             ;LOCK ON HARD ERROR
2248
2249
2250
2251
2252
2253
2254
2255 006250          ;*****
2256 006250 012700 000064      ;*TEST 64 BASIC "CMP R,#A" TEST WITH [R] = [A]
2257
2258
2259
2260
2261
2262
2263
2264 006250          ;*****
2265 006250 012700 000064      TST64:  MOV      #64,R0        ;LOAD R0 WITH TEST NUMBER
2266 006254 032737 000002 063160 .SBTTL  USER CONTROLLED BREAKPOINT -- BIT1
2267 006262 001401          BIT      #BIT1,#BPTLOC ;BREAKPOINT HALT SET ??
2268 006264 000000          BEQ      .+4           ;BR IF NOT
2269 006266 012702 063236          HALT                    ;BREAK - DEPRESS CONTINUE TO RESTART
2270 006272 012704 125252          MOV      #MBUFO,R2     ;DEST ADDR = MBUFO
2271 006276 010405          MOV      #125252,R4    ;RESULT S / B = 125252
2272 006300 010412      1$:     MOV      R4,R5         ;[R5] = SOURCE OP = 125252
2273 006302 000257          MOV      R4,(R2)       ;MAKE [DEST] = 125252
2274
2275
2276
2277
2278
2279
2280
2281 006304 020537 063236      2$:     CMP      R5,#MBUFO   ;TEST THE CMP
2282
2283
2284
2285
2286
2287
2288
2289 006310 001402          BEQ      4$             ;BR IF "Z" WAS SET - IT SHOULD BE
2290
2291
2292
2293
2294
2295
2296
2297 006312 000000      3$:     HALT                    ;CMP FAILED TO SET "Z"
2298 006314 000770          BR      1$             ;LOCK ON HARD ERROR
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399

```

C06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T65

MACY11 27(1006) 08-FEB-77 16:23 PAGE 44
 BASIC "CMP R,2#A" WITH [R] NOT EQUAL TO [A]

```

2284 006342 005005      1S:  CLR      R5          ;[R5] = SOURCE OP = 000000
2285 006344 010412      MOV      R4,(R2)       ;MAKE [DEST] = 125252
2286 006346 000277      SCC          ;SCOPE SYNC - MAKE Z=1
2287
2288 006350 020537 063236  2S:  CMP      R5,2#MBUFO ;TEST THE CMP
2289
2290 006354 001002      BNE     TST66          ;;BR IF Z=0 - IT SHOULD BE
2291
2292 006356 000000      3S:  HALT          ;CMP FAILED TO CLEAR "Z"
2293 006360 000770      BR      1S           ;LOCK ON HARD ERROR
2294
2295      ;*****
2296      ;*TEST 66      BASIC "BIS #N,2#A" TEST - N=177777,[A]=000000
2297      ;*****
2298      †T66:
2299 006362 012700 000066      MOV      #66,R0        ;:LOAD R0 WITH TEST NUMBER
2300 006366 012702 063236      MOV      #MBUFO,R2     ;:DEST ADDR = MBUFO
2301 006372 012704 177777      MOV      #-1,R4        ;:RESULT S / B = 177777
2302 006376 005012      1S:  CLR      (R2)       ;:[DEST] = 000000
2303 006400 000257      CCC          ;SCOPE SYNC
2304
2305 006402 052737 177777 063236 2S:  BIS      #-1,2#MBUFO ;TEST THE BIS
2306
2307 006410 020412      CMP      R4,(R2)       ;:RESULT OK?
2308 006412 001402      BEQ     TST67          ;;BR IF YES
2309
2310 006414 000000      3S:  HALT          ;BIS FAILED TO SET ALL BITS IN BITFLG
2311 006416 000767      BR      1S           ;LOCK ON HARD ERROR
2312
2313      ;*****
2314      ;*TEST 67      BASIC "BIC #N,2#A" TEST
2315      ;*****
2316      †T67:
2317 006420 012700 000067      MOV      #67,R0        ;:LOAD R0 WITH TEST NUMBER
2318 006424 012702 063236      MOV      #MBUFO,R2     ;:DEST ADDR = MBUFO
2319 006430 012704 000077      MOV      #77,R4        ;:RESULT S / B = 77
2320 006434 012712 177777      1S:  MOV      #-1,(R2)  ;:MAKE [DEST] = 177777
2321 006440 000257      CCC          ;SCOPE SYNC
2322
2323 006442 042737 177700 063236 2S:  BIC      #177700,2#MBUFO ;TEST THE BIC
2324
2325 006450 020412      CMP      R4,(R2)       ;:DID RESULT = 77 ?
2326 006452 001402      BEQ     TST70          ;;BR IF YES
2327
2328 006454 000000      3S:  HALT          ;BIC DELIVERED THE WRONG RESULT
2329 006456 000766      BR      1S           ;LOCK ON HARD ERROR
2330
2331      ;*****
2332      ;*TEST 70      BASIC "BIC #N,R" TEST
2333      ;*****
2334      †T70:
2335 006460 012700 000070      MOV      #70,R0        ;:LOAD R0 WITH TEST NUMBER
2336 006464 005003      1S:  CLR      R3          ;:[DEST] = 177777
2337 006466 005103      COM     R3
2338 006470 000257      CCC          ;SCOPE SYNC
2339
    
```

```

2340 006472 042703 177400      2$:  BIC      #177400,R3      ;TEST THE BIC
2341
2342 006476 022703 000377      CMP      #377,R3      ;RESULT OK?
2343 006502 001402                BEQ      TST71        ;;BR IF YES
2344
2345 006504 000000      3$:  HALT                    ;BIC FAILED TO CLEAR HI-BYTE
2346 006506 000766                BR      1$           ;LOCK ON HARD ERROR
2347
2348
2349      ;*****
2350      ;*TEST 71      BASIC "BIC #N,2(SP)" TEST
2351      ;*****
2351 006510                TST71:
2352 006510 012700 000071      MOV      #71,R0      ;;LOAD R0 WITH TEST NUMBER
2353 006514 012704 000357      MOV      #357,R4     ;RESULT S / B = 357
2354 006520 010605                MOV      SP,R5      ;SAVE SP
2355 006522 010506      1$:  MOV      R5,SP     ;RESET SP FOR ERROR LOOP
2356 006524 012746 000377      MOV      #377,-(SP) ;[DEST] = 377 PUT ON STACK
2357 006530 005746                TST      -(SP)      ;DECREMENT SP
2358 006532 000257                CCC                    ;SCOPE SYNC
2359
2360 006534 042766 000020 000002 2$:  BIC      #20,2(SP)   ;TEST THE BIC - CLEAR BIT 4
2361
2362 006542 010602                MOV      SP,R2      ;[R2] = DEST ADDR
2363 006544 005722                TST      (R2)+
2364 006546 020412                CMP      R4,(R2)    ;RESULT = 357?
2365 006550 001402                BEQ      4$         ;BR IF YES
2366
2367 006552 000000      3$:  HALT                    ;BIC FAILED TO CLR BIT2 OF DEST
2368 006554 000762                BR      1$           ;LOCK ON HARD ERROR
2369
2370 006556 010506      4$:  MOV      R5,SP
2371
2372      ;*****
2373      ;*TEST 72      BASIC "ADD #N,RN" TEST
2374      ;*****
2375 006560                TST72:
2376 006560 012700 000072      MOV      #72,R0     ;;LOAD R0 WITH TEST NUMBER
2377 006564 012703 000002      1$:  MOV      #2,R3      ;MAKE [DEST] = 2
2378 006570 000257                CCC                    ;SCOPE SYNC
2379
2380 006572 062703 000002      2$:  ADD      #2,R3      ;TEST THE ADD
2381
2382 006576 022703 000004      CMP      #4,R3      ;RESULT = 4 ?
2383 006602 001402                BEQ      TST73      ;;BR IF YES
2384
2385 006604 000000      3$:  HALT                    ;ADD DELIVERED THE WRONG RESULT
2386 006606 000766                BR      1$           ;LOCK ON HARD ERROR
2387
2388      ;*****
2389      ;*TEST 73      BASIC "ADD #N,(R)" TEST
2390      ;*****
2391 006610                TST73:
2392 006610 012700 000073      MOV      #73,R0     ;;LOAD R0 WITH TEST NUMBER
2393 006614 012702 063236      MOV      #MBUFO,R2  ;DEST ADDR = MBUFO
2394 006620 012704 000004      MOV      #4,R4      ;RESULT S / B = 4
2395 006624 012712 000002      1$:  MOV      #2,(R2)   ;MAKE [DEST] = 2

```

E06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T73

MACY11 27(1006) 08-FEB-77 16:23 PAGE 46
BASIC "ADD #N,(R)" TEST

```

2396 006630 000257          CCC          ;SCOPE SYNC
2397
2398 006632 062712 000002    2$:  ADD      #2,(R2) ;TEST THE ADD
2399
2400 006636 020412          CMP      R4,(R2) ;RESULT = 4 ?
2401 006640 001402          BEQ      TST74    ;;BR IF YES
2402
2403 006642 000000    3$:  HALT          ;ADD DELIVERED THE WRONG RESULT
2404 006644 000767          BR       1$      ;LOCK ON HARD ERROR
2405
2406          ;:*****
2407          ;:TEST 74      BASIC "ADD #N,X(R)" TEST
2408          ;:*****
2409          †TST74:
2410 006646 012700 000074    MOV      #74,R0    ;:LOAD R0 WITH TEST NUMBER
2411 006652 012704 000002    MOV      #2,R4     ;:RESULT S / B = 2
2412 006656 012702 063240    MOV      #MBUFO+2,R2 ;:DEST ADDR = MBUFO + 2
2413 006662 012705 063236    1$:  MOV      #MBUFO,R5 ;:BASE DEST ADDR = MBUFO
2414 006666 005012          CLR      (R2)     ;:MAKE [DEST] = 000000
2415 006670 000257          CCC          ;:SCOPE SYNC
2416
2417 006672 062765 000002 000002 2$:  ADD      #2,2(R5) ;:TEST THE ADD
2418
2419 006700 020412          CMP      R4,(R2) ;:RESULT = 2 ?
2420 006702 001402          BEQ      TST75    ;;BR IF YES
2421
2422 006704 000000    3$:  HALT          ;:ADD DELIVERED THE WRONG RESULT
2423 006706 000765          BR       1$      ;:LOOP ON HARD ERROR
2424
2425          ;:*****
2426          ;:TEST 75      BASIC "CMPB #N,(SP)+" TEST
2427          ;:*****
2428          †TST75:
2429 006710 012700 000075    MOV      #75,R0    ;:LOAD R0 WITH TEST NUMBER
2430 006714 012704 177400    MOV      #177400,R4 ;:RESULT S / B = 177400
2431 006720 010605          MOV      SP,R5     ;:SAVE SP
2432 006722 010602          MOV      SP,R2     ;:SET UP DEST ADDR
2433 006724 005742          TST     -(R2)     ;:R2 CONTAINS DEST ADDR
2434 006726 010506    1$:  MOV      R5,SP    ;:RESET SP FOR ERROR LOOP
2435 006730 010446          MOV      R4,-(SP) ;:MAKE [DEST] = 177400
2436 006732 000257          CCC          ;:SCOPE SYNC - "Z" = 0
2437
2438 006734 122726 000000    2$:  CMPB     #0,(SP)+ ;:TEST THE CMPB
2439
2440 006740 001402          BEQ      4$      ;:BR IF "Z" SET - IT SHOULD BE
2441
2442 006742 000000    3$:  HALT          ;:CMPB FAILED TO SET "Z"
2443 006744 000770          BR       1$      ;:LOCK ON HARD ERROR
2444
2445 006746 020506    4$:  CMP      R5,SP    ;:DID SP GET UPDATED BY 2?
2446 006750 001402          BEQ      6$      ;:BR IF YES
2447
2448 006752 000000    5$:  HALT          ;:CMPB FAILED TO UPDATE SP PROPERLY
2449 006754 000764          BR       1$      ;:LOCK ON HARD ERROR
2450
2451 006756 020412    6$:  CMP      R4,(R2) ;:[DEST] ALTERED?

```

F06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T75

MACY11 27(1006) 08-FEB-77 16:23 PAGE 47
 BASIC "CMPB #N,(SP)+" TEST

```

2452 006760 001402          BEQ      TST76          ;;BR IF NOT
2453
2454 006762 000000          7$:    HALT              ;CMPB MODIFIED [DEST]
2455 006764 000760          BR      1$              ;LOCK ON HARD ERROR.
2456
2457          ;;*****
2458          ;*TEST 76      BASIC "CMPB (RA)+,(RB)+" - SRC AND DEST EVEN
2459          ;*****
2460          †TST76:
2461 006766 012700 000076      MOV      #76,R0          ;;LOAD R0 WITH TEST NUMBER
2462 006772 012704 177777      MOV      #-1,R4         ;RESULT S / B = 177777
2463 006776 012702 063250      MOV      #DWTA+2,R2     ;DEST ADDR = DWTA+2
2464 007002 012705 063254      1$:    MOV      #DWTA+6,R5  ;SRC ADDR = DWTA+6
2465 007006 010203          MOV      R2,R3          ;R3 GETS DEST ADDR
2466 007010 000257          CCC                    ;SCOPE SYNC
2467
2468 007012 122523          2$:    CMPB     (R5)+,(R3)+ ;TEST THE CMPB
2469
2470 007014 001402          BEQ      4$              ;BR IF "Z" = 1 - IT SHOULD BE
2471
2472 007016 000000          3$:    HALT              ;CMPB FAILED TO SET "Z"
2473 007020 000770          BR      1$              ;LOCK ON HARD ERROR
2474
2475 007022 022703 063251          4$:    CMP      #DWTA+3,R3 ;DID DEST REG GET UPDATED?
2476 007026 001402          BEQ      6$              ;BR IF YES
2477
2478 007030 000000          5$:    HALT              ;CMPB FAILED TO UPDATE DEST REG
2479 007032 000763          BR      1$              ;LOCK ON HARD ERROR
2480
2481 007034 022705 063255          6$:    CMP      #DWTA+7,R5  ;DID SRC REG GET UPDATED?
2482 007040 001402          BEQ      8$              ;BR IF YES
2483
2484 007042 000000          7$:    HALT              ;CMPB FAILED TO UPDATE SRC REG
2485 007044 000756          BR      1$              ;LOCK ON HARD ERROR
2486
2487 007046 020412          8$:    CMP      R4,(R2)    ;DID [DEST] GET ALTERED?
2488 007050 001403          BEQ      TST77          ;;BR IF NOT
2489
2490 007052 000000          9$:    HALT              ;CMPB DELIVERED A RESULT
2491 007054 010412          MOV      R4,(R2)        ;RESTORE [DEST]
2492 007056 000751          BR      1$              ;LOCK ON HARD ERROR
2493
2494          ;;*****
2495          ;*TEST 77      BASIC "CMPB (RA)+,(RB)+" - SRC AND DEST ODD
2496          ;*****
2497          †TST77:
2498 007060 012700 000077      MOV      #77,R0          ;;LOAD R0 WITH TEST NUMBER
2499 007064 012704 177777      MOV      #-1,R4         ;RESULT S / B = 177777
2500 007070 012702 063250      MOV      #DWTA+2,R2     ;DEST ADDR = DWTA+2
2501 007074 012705 063253      1$:    MOV      #DWTA+5,R5  ;SRC ADDR = DWTA+5
2502 007100 012703 063251      MOV      #DWTA+3,R3     ;R3 GETS DEST ADDR+1
2503 007104 000257          CCC                    ;SCOPE SYNC
2504
2505 007106 122523          2$:    CMPB     (R5)+,(R3)+ ;TEST THE CMPB
2506
2507 007110 001402          BEQ      4$              ;BR IF "Z" = 1 - IT SHOULD BE

```


G06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 48
BASIC "CMPB (RA)+,(RB)+ - SRC AND DEST ODD

```

2508
2509 007112 000000      3$:  HALT
2510 007114 000767      BR      1$      ;CMPB FAILED TO SET "Z"
2511                                     ;LOCK ON HARD ERROR
2512 007116 022703 063252  4$:  CMP      #DWT+4,R3      ;DID DEST REG GET UPDATED?
2513 007122 001402      BEQ      6$      ;BR IF YES
2514
2515 007124 000000      5$:  HALT
2516 007126 000762      BR      1$      ;CMPB FAILED TO UPDATE DEST REG
2517                                     ;LOCK ON HARD ERROR
2518 007130 022705 063254  6$:  CMP      #DWT+6,R5      ;DID SRC REG GET UPDATED?
2519 007134 001402      BEQ      8$      ;BR IF YES
2520
2521 007136 000000      7$:  HALT
2522 007140 000755      BR      1$      ;CMPB FAILED TO UPDATE SRC REG
2523                                     ;LOCK ON HARD ERROR
2524 007142 020412      8$:  CMP      R4,(R2)      ;DID [DEST] GET ALTERED?
2525 007144 001403      BEQ      T$T100      ;;BR IF NOT
2526
2527 007146 000000      9$:  HALT
2528 007150 010412      MOV      R4,(R2)      ;CMPB DELIVERED A RESULT
2529 007152 000750      BR      1$      ;RESTORE [DEST]
2530                                     ;LOCK ON HARD ERROR
2531
2532 ;*****
2533 ;*TEST 100 BASIC "CMPB (RA)+,(RB)+ - SRC / EVEN,DEST / ODD
2534 ;*****
2535 007154 012700 000100  T$T100:  MOV      #100,R0      ;:LOAD R0 WITH TEST NUMBER
2536 007154 012704 177400  MOV      #177400,R4   ;:RESULT S / B = 177400
2537 007164 012702 063252  MOV      #DWT+4,R2   ;:DEST ADDR = DWT+4
2538 007170 012705 063254  1$:  MOV      #DWT+6,R5   ;:SRC ADDR = DWT+6
2539 007174 012703 063253  MOV      #DWT+5,R3   ;:R3 GETS DEST ADDR
2540 007200 000257      CCC                                     ;SCOPE SYNC
2541
2542 007202 122523      2$:  CMPB     (R5)+,(R3)+  ;TEST THE CMPB
2543
2544 007204 001402      BEQ      4$      ;BR IF "Z" = 1 - IT SHOULD BE
2545
2546 007206 000000      3$:  HALT
2547 007210 000767      BR      1$      ;CMPB FAILED TO SET "Z"
2548                                     ;LOCK ON HARD ERROR
2549 007212 022703 063254  4$:  CMP      #DWT+6,R3      ;DID DEST REG GET UPDATED?
2550 007216 001402      BEQ      6$      ;BR IF YES
2551
2552 007220 000000      5$:  HALT
2553 007222 000762      BR      1$      ;CMPB FAILED TO UPDATE DEST REG
2554                                     ;LOCK ON HARD ERROR
2555 007224 022705 063255  6$:  CMP      #DWT+7,R5      ;DID SRC REG GET UPDATED?
2556 007230 001402      BEQ      8$      ;BR IF YES
2557
2558 007232 000000      7$:  HALT
2559 007234 000755      BR      1$      ;CMPB FAILED TO UPDATE SRC REG
2560                                     ;LOCK ON HARD ERROR
2561 007236 020412      8$:  CMP      R4,(R2)      ;DID [DEST] GET ALTERED?
2562 007240 001403      BEQ      T$T101      ;;BR IF NOT
2563

```

H06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 49
 DQKDA.P11 08-FEB-77 16:17 T100 BASIC "CMPB (RA)+,(RB)+" - SRC / EVEN,DEST / ODD

```

2564 007242 000000          9$:  HALT                    ;CMPB DELIVERED A RESULT
2565 007244 010412          MOV      R4,(R2)          ;RESTORE [DEST]
2566 007246 000750          BR       1$              ;LOCK ON HARD ERROR
2567
2568
2569
2570
2571 007250
2572 007250 012700 000101      ;:*****
2573 007254 012704 177777      ;:TEST 101 BASIC "CMPB (RA)+,(RB)+" - SRC / ODD,DEST / EVEN
2574 007260 012702 063250      ;:*****
2575 007264 012705 063253      †ST101:
2576 007270 010203          MOV      #101,R0          ;:LOAD R0 WITH TEST NUMBER
2577 007272 000257          MOV      #-1,R4          ;:RESULT S / B = 177777
2578
2579 007274 122523          1$:  MOV      #DWT+2,R2      ;:DEST ADDR = DWT+2
2580
2581 007276 001402          MOV      #DWT+5,R5      ;:SRC ADDR = DWT+5
2582
2583 007300 000000          MOV      R2,R3          ;:R3 GETS DEST ADDR
2584 007302 000770          CCC                    ;:SCOPE SYNC
2585
2586 007304 022703 063251      2$:  CMPB     (R5)+,(R3)+   ;:TEST THE CMPB
2587 007310 001402          BEQ     4$              ;:BR IF "Z" = 1 - IT SHOULD BE
2588
2589 007312 000000          3$:  HALT                    ;:CMPB FAILED TO SET "Z"
2590 007314 000763          BR       1$              ;:LOCK ON HARD ERROR
2591
2592 007316 022705 063254      4$:  CMP      #DWT+3,R3    ;:DID DEST REG GET UPDATED?
2593 007322 001402          BEQ     6$              ;:BR IF YES
2594
2595 007324 000000          5$:  HALT                    ;:CMPB FAILED TO UPDATE DEST REG
2596 007326 000756          BR       1$              ;:LOCK ON HARD ERROR
2597
2598 007330 020412          6$:  CMP      #DWT+6,R5    ;:DID SRC REG GET UPDATED?
2599 007332 001403          BEQ     8$              ;:BR IF YES
2600
2601 007334 000000          7$:  HALT                    ;:CMPB FAILED TO UPDATE SRC REG
2602 007336 010412          BR       1$              ;:LOCK ON HARD ERROR
2603 007340 000751          8$:  CMP      R4,(R2)      ;:DID [DEST] GET ALTERED?
2604
2605
2606
2607
2608
2609 007342
2610 007342 012700 000102      9$:  HALT                    ;:CMPB DELIVERED A RESULT
2611 007346 012702 063242      MOV      R4,(R2)          ;:RESTORE [DEST]
2612 007352 012703 063236      BR       1$              ;:LOCK ON HARD ERROR
2613 007356 012704 177400
2614 007362 012705 064554      ;:*****
2615 007372 000257          ;:TEST 102 BASIC "MOVB (RA)+,X(RB) - SRC EVEN / DEST EVEN
2616
2617 007374 112563 000004      ;:*****
2618
2619 007400 020412          †ST102:
2619 007400 020412          MOV      #102,R0          ;:LOAD R0 WITH TEST NUMBER
2619 007400 020412          MOV      #MBUF1,R2       ;:DEST ADDR = MBUF1
2619 007400 020412          MOV      #MBUF0,R3       ;:BASE DEST ADDR = MBUF0
2619 007400 020412          MOV      #177400,R4      ;:RESULT S / B = 177400
2619 007400 020412          1$:  MOV      #DBTA,R5     ;:SRC ADDR = DBTA
2619 007400 020412          MOV      #-1,(R2)        ;:[DEST] = 177777
2619 007400 020412          CCC                    ;:SCOPE SYNC
2619 007400 020412          2$:  MOVB    (R5)+,4(R3)   ;:TEST THE MOVB
2619 007400 020412          CMP     R4,(R2)          ;:RESULT OK?
  
```

```

2620 007402 001402          BEQ      4S          ;BR IF YES
2621
2622 007404 000000          3S:    HALT          ;MOV DELIVERED WRONG RESULT
2623 007406 000765          BR      1S          ;LOCK ON HARD ERROR
2624
2625 007410 022705 064555          4S:    CMP      #DBTA+1,R5 ;DID SRC REG GET INCREMENTED BY +1
2626 007414 001402          BEQ      TST103      ;;BR IF YES
2627
2628 007416 000000          5S:    HALT          ;MOVB FAILED TO UPDATE SRC REG
2629 007420 000760          BR      1S          ;LOCK ON HARD ERROR
2630
2631          ;*****
2632          ;*TEST 103 BASIC "MOVB (RA)+,X(RB) - SRC ODD / DEST ODD
2633          ;*****
2634          †TST103:
2635          MOV      #103,R0          ;;LOAD R0 WITH TEST NUMBER
2636          MOV      #MBUF1,R2       ;DEST ADDR = MBUF1
2637          MOV      #MBUFO,R3      ;BASE DEST ADDR = MBUFO
2638          MOV      #777,R4        ;RESULT S / B = 777
2639          1S:    MOV      #DBTB+1,R5 ;SRC ADDR = DBTB+1
2640          MOV      #-1,(R2)       ;[DEST] = 177777
2641          CCC
2642          ;SCOPE SYNC
2643
2644 007422 012700 000103          2S:    MOVVB   (R5)+,5(R3)      ;TEST THE MOVVB
2645 007426 012702 063242          CMP      R4,(R2)          ;RESULT OK?
2646 007432 012703 063236          BEQ      4S          ;BR IF YES
2647 007436 012704 000777          3S:    HALT          ;MOV DELIVERED WRONG RESULT
2648 007442 012705 064561          BR      1S          ;LOCK ON HARD ERROR
2649 007446 012712 177777          4S:    CMP      #DBTB+2,R5      ;DID SRC REG GET INCREMENTED BY +1
2650 007452 000257          BEQ      TST104      ;;BR IF YES
2651
2652 007476 000000          5S:    HALT          ;MOVB FAILED TO UPDATE SRC REG
2653 007500 000760          BR      1S          ;LOCK ON HARD ERROR
2654
2655          ;*****
2656          ;*TEST 104 BASIC "MOVB (RA)+,X(RB) - SRC EVEN / DEST ODD
2657          ;*****
2658          †TST104:
2659          MOV      #104,R0          ;;LOAD R0 WITH TEST NUMBER
2660          MOV      #MBUF1,R2       ;DEST ADDR = MBUF1
2661          MOV      #MBUFO,R3      ;BASE DEST ADDR = MBUFO
2662          MOV      #377,R4        ;RESULT S / B = 377
2663          1S:    MOV      #DBTA,R5  ;SRC ADDR = DBTA
2664          MOV      #-1,(R2)       ;[DEST] = 177777
2665          CCC
2666          ;SCOPE SYNC
2667
2668 007502 012700 000104          2S:    MOVVB   (R5)+,5(R3)      ;TEST THE MOVVB
2669 007506 012702 063242          CMP      R4,(R2)          ;RESULT OK?
2670 007512 012703 063236          BEQ      4S          ;BR IF YES
2671 007516 012704 000377          3S:    HALT          ;MOV DELIVERED WRONG RESULT
2672 007522 012705 064554          BR      1S          ;LOCK ON HARD ERROR
2673 007526 012712 177777          4S:    CMP      #DBTA+1,R5      ;DID SRC REG GET INCREMENTED BY +1
2674 007532 000257
2675 007534 112563 000005

```

J06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T104

MACY11 27(1006) 08-FEB-77 16:23 PAGE 51
 BASIC "MOVB (RA)+,X(RB) - SRC EVEN / DEST ODD

```

2676 007554 001402          BEQ     TST105          ;;BR IF YES
2677
2678 007556 000000          5S:    HALT             ;MOVB FAILED TO UPDATE SRC REG
2679 007560 000760          BR      1S              ;LOCK ON HARD ERROR
2680
2681
2682
2683
2684 007562 012700 000105          ;:*****
2685 007566 012702 063242          ;*TEST 105 BASIC "MOVB (RA)+,X(RB) - SRC ODD / DEST EVEN
2686 007572 012703 063236          ;:*****
2687 007576 012704 177401          †TST105:
2688 007602 012705 064561          MOV     #105,R0         ;;LOAD R0 WITH TEST NUMBER
2689 007606 012712 177777          MOV     #MBUF1,R2       ;DEST ADDR = MBUF1
2690 007612 000257          MOV     #MBUF0,R3       ;BASE DEST ADDR = MBUF0
2691
2692 007614 112563 000004          1S:    MOV     #177401,R4 ;RESULT S / B = 177401
2693
2694 007620 020412          MOV     #DBTB+1,R5      ;SRC ADDR = DBTB+1
2695 007622 001402          MOV     #-1,(R2)        ;[DEST] = 177777
2696
2697 007624 000000          CCC
2698 007626 000765          ;SCOPE SYNC
2699
2700 007630 022705 064562          2S:    MOVB    (R5)+,4(R3) ;TEST THE MOVB
2701 007634 001402          CMP     R4,(R2)         ;RESULT OK?
2702
2703 007636 000000          BEQ     4S              ;BR IF YES
2704 007640 000760          3S:    HALT             ;MOV DELIVERED WRONG RESULT
2705
2706
2707
2708
2709 007642 012700 000106          BR      1S              ;LOCK ON HARD ERROR
2710 007642 012702 063236          4S:    CMP     #DBTB+2,R5 ;DID SRC REG GET INCREMENTED BY +1
2711 007646 012704 177401          BEQ     TST106          ;:BR IF YES
2712 007652 012705 063756          5S:    HALT             ;MOVB FAILED TO UPDATE SRC REG
2713 007656 012705 063756          BR      1S              ;LOCK ON HARD ERROR
2714 007662 010203          ;:*****
2715 007664 012713 177777          ;*TEST 106 BASIC "MOVB 2(RA),(RB)+" TEST - SRC EVEN / DEST EVEN
2716 007670 000257          ;:*****
2717
2718 007672 116523 000002          †TST106:
2719
2720 007676 020412          MOV     #106,R0         ;;LOAD R0 WITH TEST NUMBER
2721 007700 001402          MOV     #MBUF0,R2       ;DEST ADDR = MBUF0
2722
2723 007702 000000          MOV     #177401,R4      ;RESULT S / B = 177401
2724 007704 000766          MOV     #DWTB,R5        ;SRC ADDR = DWTB
2725
2726 007706 022703 063237          1S:    MOV     R2,R3        ;R3 GETS DEST ADDR
2727 007712 001402          MOV     #-1,(R3)        ;[DEST] = 177400
2728
2729 007714 000000          CCC
2730 007716 000761          ;SCOPE SYNC
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
    
```

K06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 52
 BASIC "MOVB 2(RA),(RB)+" TEST - SRC ODD / DEST EVEN

```

2732
2733
2734
2735 007720
2736 007720 012700 000107
2737 007724 012702 063236
2738 007730 012704 177401
2739 007734 012705 064560
2740 007740 010203
2741 007742 012713 177777
2742 007746 000257
2743
2744 007750 116523 000001
2745
2746 007754 020412
2747 007756 001402
2748
2749 007760 000000
2750 007762 000766
2751
2752 007764 022703 063237
2753 007770 001402
2754
2755 007772 000000
2756 007774 000761
2757
2758
2759
2760
2761 007776
2762 007776 012700 000110
2763 010002 012702 063236
2764 010006 012704 000777
2765 010012 012705 063756
2766 010016 012703 063237
2767 010022 012712 177777
2768 010026 000257
2769
2770 010030 116523 000002
2771
2772 010034 020412
2773 010036 001402
2774
2775 010040 000000
2776 010042 000765
2777
2778 010044 022703 063240
2779 010050 001402
2780
2781 010052 000000
2782 010054 000760
2783
2784
2785
2786
2787 010056

;*****
;TEST 107 BASIC "MOVB 2(RA),(RB)+" TEST - SRC ODD / DEST EVEN
;*****
†TST107:
MOV #107,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #MBUFD,R2 ;:DEST ADDR = MBUFD
MOV #177401,R4 ;:RESULT S / B = 177401
MOV #DBTB,R5 ;:SRC ADDR = DBTB
1$: MOV R2,R3 ;:R3 GETS DEST ADDR
MOV #-1,(R3) ;:[DEST] = 177777
CCC ;:SCOPE SYNC

2$: MOVB 1(R5),(R3)+ ;:TEST THE MOVB

CMP R4,(R2) ;:RESULT OK?
BEQ 4$ ;:BR IF YES

3$: HALT ;:MOVB DELIVERED WRONG RESULT
BR 1$ ;:LOCK ON HARD ERROR

4$: CMP #MBUFD+1,R3 ;:DID DEST REG GET INCREMENTED?
BEQ TST110 ;:BR IF YES

5$: HALT ;:MOVB FAILED TO AUTO INCREMENT DEST REG
BR 1$ ;:LOCK ON HARD ERROR

;*****
;TEST 110 BASIC "MOVB 2(RA),(RB)+" TEST - SRC EVEN / DEST ODD
;*****
†TST110:
MOV #110,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #MBUFD,R2 ;:DEST ADDR = MBUFD
MOV #777,R4 ;:RESULT S / B = 777
MOV #DWTB,R5 ;:SRC ADDR = DWTB
1$: MOV #MBUFD+1,R3 ;:R3 GETS DEST ADDR
MOV #-1,(R2) ;:[DEST] = 177777
CCC ;:SCOPE SYNC

2$: MOVB 2(R5),(R3)+ ;:TEST THE MOVB

CMP R4,(R2) ;:RESULT OK?
BEQ 4$ ;:BR IF YES

3$: HALT ;:MOVB DELIVERED WRONG RESULT
BR 1$ ;:LOCK ON HARD ERROR

4$: CMP #MBUFD+2,R3 ;:DID DEST REG GET INCREMENTED?
BEQ TST111 ;:BR IF YES

5$: HALT ;:MOVB FAILED TO AUTO INCREMENT DEST REG
BR 1$ ;:LOCK ON HARD ERROR

;*****
;TEST 111 BASIC "MOVB 2(RA),(RB)+" TEST - SRC ODD / DEST ODD
;*****
†TST111:

```

```

2788 010056 012700 000111      MOV      #111,R0      ;;LOAD R0 WITH TEST NUMBER
2789 010062 012702 063236      MOV      #MBUF0,R2   ;;DEST ADDR = MBUF0
2790 010066 012704 000777      MOV      #777,R4     ;;RESULT S / B = 777
2791 010072 012705 064560      MOV      #DBTB,R5    ;;SRC ADDR = DBTB
2792 010076 012703 063237      1S:    MOV      #MBUF0+1,R3 ;;R3 GETS DEST ADDR = MBUF0+1
2793 010102 012712 177777      MOV      #-1,(R2)    ;;[DEST] = 177777
2794 010106 000257      CCC                               ;;SCOPE SYNC
2795
2796 010110 116523 000001      2S:    MOVB     1(R5),(R3)+ ;;TEST THE MOVB
2797
2798 010114 020412      CMP      R4,(R2)     ;;RESULT OK?
2799 010116 001402      BEQ      4S         ;;BR IF YES
2800
2801 010120 000000      3S:    HALT                               ;;MOVB DELIVERED WRONG RESULT
2802 010122 000765      BR      1S         ;;LOCK ON HARD ERROR
2803
2804 010124 022703 063240      4S:    CMP      #MBUF0+2,R3 ;;DID DEST REG GET INCREMENTED?
2805 010130 001402      BEQ      TST112    ;;BR IF YES
2806
2807 010132 000000      5S:    HALT                               ;;MOVB FAILED TO AUTO INCREMENT DEST REG
2808 010134 000760      BR      1S         ;;LOCK ON HARD ERROR
2809

```

```

*****
; *TEST 112      BASIC "MOVB -(RA),RB" TEST - SRC EVEN ADDR
*****

```

```

2810
2811
2812
2813 010136
2814 010136 012700 000112      TST112: MOV      #112,R0     ;;LOAD R0 WITH TEST NUMBER
2815 010142 012705 063255      1S:    MOV      #DWT+7,R5  ;;SRC ADDR = DWT+7
2816 010146 005003      CLR      R3         ;;[DEST] = 000000
2817 010150 000257      CCC                               ;;SCOPE SYNC
2818
2819 010152 114503      2S:    MOVB     -(R5),R3  ;;TEST THE MOVB
2820
2821 010154 022703 177777      CMP      #-1,R3     ;;RESULT OK?
2822 010160 001402      BEQ      4S         ;;BR IF YES
2823
2824 010162 000000      3S:    HALT                               ;;MOVB FAILED - WRONG RESULT
2825 010164 000766      BR      1S         ;;LOCK ON HARD ERROR
2826
2827 010166 022705 063254      4S:    CMP      #DWT+6,R5  ;;SRC REG GET DECREMENTED?
2828 010172 001402      BEQ      TST113    ;;BR IF YES
2829
2830 010174 000000      5S:    HALT                               ;;MOVB FAILED TO UPDATE SRC REG
2831 010176 000761      BR      1S         ;;LOCK ON HARD ERROR
2832

```

```

*****
; *TEST 113      BASIC "MOVB -(RA),RB" TEST - SRC ODD ADDR
*****

```

```

2833
2834
2835 010200
2836 010200 012700 000113      TST113: MOV      #113,R0     ;;LOAD R0 WITH TEST NUMBER
2837 010204 012705 063254      1S:    MOV      #DWT+6,R5  ;;SRC ADDR = DWT+6
2838 010210 005003      CLR      R3         ;;[DEST] = 000000
2839 010212 000257      CCC                               ;;SCOPE SYNC
2840
2841 010214 114503      2S:    MOVB     -(R5),R3  ;;TEST THE MOVB
2842
2843 010216 022703 177777      CMP      #-1,R3     ;;RESULT OK?

```

M06

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T113

MACY11 27(1006) 08-FEB-77 16:23 PAGE 54
BASIC "MOVB -(RA),R8" TEST - SRC ODD ADDR

```

2844 010222 001402          BEQ      4S          ;BR IF YES
2845
2846 010224 000000          3S:    HALT          ;MOVB FAILED - WRONG RESULT
2847 010226 000766          BR      1S          ;LOCK ON HARD ERROR
2848
2849 010230 022705 063253    4S:    CMP      #DWT+5,R5 ;SRC REG GET DECREMENTED?
2850 010234 001402          BEQ      TST114      ;;BR IF YES
2851
2852 010236 000000          5S:    HALT          ;MOVB FAILED TO UPDATE SRC REG
2853 010240 000761          BR      1S          ;LOCK ON HARD ERROR
2854
2855 ;*****
2856 ;*TEST 114 BASIC "MOVB (RA)+,-(SP)" TEST - SRC ADDR EVEN
2857 ;*****
2858
2859 010242          †TST114:
2860 010242 012700 000114      MOV      #114,R0      ;:LOAD R0 WITH TEST NUMBER
2861 010246 010605          MOV      SP,R5        ;:SAVE SP
2862 010250 012704 177400      MOV      #177400,R4   ;:RESULT S / B = 177400
2863 010254 010506          1S:    MOV      R5,SP       ;:RESET SP FOR ERROR LOOP
2864 010256 012703 064554      MOV      #DBTA,R3     ;:SRC ADDR = DBTA
2865 010262 012746 177777      MOV      #-1,-(SP)    ;:(DEST) = 177777
2866 010266 010602          MOV      SP,R2        ;:R2 GETS DEST ADDR
2867 010270 005726          TST      (SP)+        ;:RESET SP
2868 010272 000257          CCC          ;:SCOPE SYNC
2869
2870 010274 112346          2S:    MOVB     (R3)+,-(SP) ;TEST THE MOVB
2871 010276 022703 064555      CMP      #DBTA+1,R3   ;:DID MOVB INCREMENT SRC REG?
2872 010302 001402          BEQ      4S          ;BR IF YES
2873
2874 010304 000000          3S:    HALT          ;MOVB FAILED TO UPDATE SRC REG
2875 010306 000762          BR      1S          ;LOCK ON HARD ERROR
2876
2877 010310 020412          4S:    CMP      R4,(R2)  ;:RESULT OK?
2878 010312 001402          BEQ      6S          ;BR IF YES
2879
2880 010314 000000          HALT          ;MOVB FAILED TO DELIVER CORRECT RESULT
2881 010316 000756          BR      1S          ;LOCK ON HARD ERROR
2882
2883 010320 020206          6S:    CMP      R2,SP     ;:DID SP GET PUSHED BY 2 ?
2884 010322 001402          BEQ      8S          ;BR IF YES
2885
2886 010324 000000          HALT          ;MOVB FAILED TO PUSH SP PROPERLY
2887 010326 000752          BR      1S          ;LOCK ON HARD ERROR
2888
2889 010330 010506          8S:    MOV      R5,SP     ;:RESET SP IN CASE OF ERROR
2890
2891 ;*****
2892 ;*TEST 115 BASIC "MOVB (RA)+,-(SP)" TEST - SRC ADDR ODD
2893 ;*****
2894
2895 010332          †TST115:
2896 010332 012700 000115      MOV      #115,R0      ;:LOAD R0 WITH TEST NUMBER
2897 010336 010605          MOV      SP,R5        ;:SAVE SP
2898 010340 012704 177400      MOV      #177400,R4   ;:RESULT S / B = 177400
2899 010344 010506          1S:    MOV      R5,SP       ;:RESET SP FOR ERROR LOOP
2900 010346 012703 063761      MOV      #DWTB+3,R3   ;:SRC ADDR = DWTB+3

```

```

2900 010352 012746 177777      MOV      #-1, -(SP)      ;[DEST] = 177777
2901 010356 010602      MOV      SP, R2        ;R2 GETS DEST ADDR
2902 010360 005726      TST      (SP)+         ;RESET SP
2903 010362 000257      CCC                      ;SCOPE SYNC
2904
2905 010364 112346      2$:      MOVB      (R3)+,-(SP) ;TEST THE MOVB
2906
2907 010366 022703 063762      CMP      #DWTB+4,R3    ;DID MOVB INCREMENT SRC REG?
2908 010372 001402      BEQ      4$            ;BR IF YES
2909
2910 010374 000000      3$:      HALT                      ;MOVB FAILED TO UPDATE SRC REG
2911 010376 000762      BR                          ;LOCK ON HARD ERROR
2912
2913 010400 020412      4$:      CMP      R4,(R2)        ;RESULT OK?
2914 010402 001402      BEQ      6$            ;BR IF YES
2915
2916 010404 000000      HALT                      ;MOVB FAILED TO DELIVER CORRECT RESULT
2917 010406 000756      BR                          ;LOCK ON HARD ERROR
2918
2919 010410 020206      6$:      CMP      R2,SP          ;DID SP GET PUSHED BY 2
2920 010412 001402      BEQ      8$            ;BR IF YES
2921
2922 010414 000000      HALT                      ;MOVB FAILED TO PUSH SP
2923 010416 000752      BR                          ;LOCK ON HARD ERROR
2924
2925 010420 010506      8$:      MOV      R5,SP          ;RESET SP IN CASE OF ERROR
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955

```

; *TEST 116 BASIC "MOVB X(R),@#A" TEST - SRC EVEN / DEST EVEN

†ST116:

```

2931 010422 012700 000116      MOV      #116,R0        ;:LOAD R0 WITH TEST NUMBER
2932 010426 012702 063236      MOV      #MBUFO,R2      ;:DEST ADDR = MBUFO
2933 010432 012704 000001      MOV      #1,R4          ;:RESULT S / B = 1
2934 010436 012705 063756      MOV      #DWTB,R5      ;:BASE SRC ADDR = DWTB
2935 010442 005012      1$:      CLR      (R2)          ;:[DEST] = 000000
2936 010444 000257      CCC                      ;:SCOPE SYNC
2937
2938 010446 116537 000006 063236 2$:      MOVB      6(R5),@#MBUFO ;TEST THE MOVB
2939
2940 010454 020412      CMP      R4,(R2)        ;:RESULT OK?
2941 010456 001402      BEQ      TST117         ;:BR IF YES
2942
2943 010460 000000      3$:      HALT                      ;:MOVB DELIVERED WRONG RESULT
2944 010462 000767      BR                          ;:LOCK ON HARD ERROR
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955

```

; *TEST 117 BASIC "MOVB X(R),@#A" TEST - SRC ODD / DEST EVEN

†ST117:

```

2949 010464 012700 000117      MOV      #117,R0        ;:LOAD R0 WITH TEST NUMBER
2950 010470 012702 063236      MOV      #MBUFO,R2      ;:DEST ADDR = MBUFO
2951 010474 012704 000001      MOV      #1,R4          ;:RESULT S / B = 1
2952 010500 012705 064560      MOV      #DBTB,R5      ;:BASE SRC ADDR = DBTB
2953 010504 005012      1$:      CLR      (R2)          ;:[DEST] = 000000
2954 010506 000257      CCC                      ;:SCOPE SYNC
2955

```



```

2956 010510 116537 000001 063236 2S:  MOVB  1(R5),D#MBUFO ;TEST THE MOVB
2957
2958 010516 020412          CMP   R4,(R2) ;RESULT OK?
2959 010520 001402          BEQ   TST120  ;;BR IF YES
2960
2961 010522 000000          3S:  HALT                ;MOVB DELIVERED WRONG RESULT
2962 010524 000767          BR   1S             ;LOCK ON HARD ERROR
2963
2964          ;;*****
2965          ;;*TEST 120 BASIC "MOVB X(R),D#A" TEST - SRC EVEN / DEST ODD
2966          ;;*****
2967          TST120:
2968 010526 012700 000120      MOV   #120,R0      ;;LOAD R0 WITH TEST NUMBER
2969 010532 012702 063236      MOV   #MBUFO,R2   ;DEST ADDR = MBUFO
2970 010536 012704 000400      MOV   #400,R4     ;RESULT S / B = 400
2971 010542 012705 063756      MOV   #DWTB,R5    ;BASE SRC ADDR = DWTB
2972 010550 000257          1S:  CLR   (R2)        ;[DEST] = 000000
2973          CCC                          ;SCOPE SYNC
2974 010552 116537 000006 063237 2S:  MOVB  6(R5),D#MBUFO+1 ;TEST THE MOVB
2975
2976 010560 020412          CMP   R4,(R2) ;RESULT OK?
2977 010562 001402          BEQ   TST121  ;;BR IF YES
2978
2979 010564 000000          3S:  HALT                ;MOVB DELIVERED WRONG RESULT
2980 010566 000767          BR   1S             ;LOCK ON HARD ERROR
2981
2982          ;;*****
2983          ;;*TEST 121 BASIC "MOVB X(R),D#A" TEST - SRC ODD / DEST ODD
2984          ;;*****
2985          TST121:
2986 010570 012700 000121      MOV   #121,R0     ;;LOAD R0 WITH TEST NUMBER
2987 010574 032737 000004 063160 .SBTTL USER CONTROLLED BREAKPOINT -- BIT2
2988 010602 001401          BIT   #BIT2,D#BPTLOC ;BREAKPOINT HALT SET ??
2989 010604 000000          BEQ   .+4         ;BR IF NOT
2990 010606 012702 063236      HALT                ;BREAK - DEPRESS CONTINUE TO RESTART
2991 010612 012704 000400      MOV   #MBUFO,R2   ;DEST ADDR = MBUFO
2992 010616 012705 064560      MOV   #400,R4     ;RESULT S / B = 400
2993 010622 005012          MOV   #DBTB,R5    ;BASE SRC ADDR = DBTB
2994 010624 000257          1S:  CLR   (R2)        ;[DEST] = 000000
2995          CCC                          ;SCOPE SYNC
2996 010626 116537 000001 063237 2S:  MOVB  1(R5),D#MBUFO+1 ;TEST THE MOVB
2997
2998 010634 020412          CMP   R4,(R2) ;RESULT OK?
2999 010636 001402          BEQ   TST122  ;;BR IF YES
3000
3001 010640 000000          3S:  HALT                ;MOVB DELIVERED WRONG RESULT
3002 010642 000767          BR   1S             ;LOCK ON HARD ERROR
3003
3004          ;;*****
3005          ;;*TEST 122 BASIC QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLG=0
3006          ;;*****
3007          TST122:
3008 010644 012700 000122      MOV   #122,R0     ;;LOAD R0 WITH TEST NUMBER
3009 010650 000257          1S:  CCC                          ;CLEAR ALL FLAGS
3010
3011 010652 001404          2S:  BEQ   3S             ;NO BR SHOULD OCCUR-FLAG=0
    
```

```

3012 010654 100403      BMI      3$      ;NO BR SHOULD OCCUR-FLAG=0
3013 010656 102402      BVS      3$      ;NO BR SHOULD OCCUR-FLAG=0
3014 010660 103401      BCS      3$      ;NO BR SHOULD OCCUR-FLAG=0
3015 010662 000402      BR       TST123  ;;GO TO NEXT TEST
3016
3017 010664 000000      3$:  HALT      ;ONE OF ABOVE BR'S FAILED
3018 010666 000770      BR       1$      ;ERROR LOOP RETURN
3019

```

```

3020
3021 ;:*****
3022 ;*TEST 123 BASIC QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLAG=1
3023 ;:*****

```

```

3023 010670          TST123:
3024 010670 012700 000123      MOV      #123,R0      ;;LOAD R0 WITH TEST NUMBER
3025 010674 000277      1$:  SCC          ;MAKE N:C = 1111
3026
3027 010676 001402      21$:  BEQ      22$      ;TEST THE BEQ-IT SHOULD BR
3028
3029 010700 000000      3$:  HALT      ;BEQ FAILED
3030 010702 000774      BR       1$      ;ERROR LOOP RETURN
3031
3032 010704 100402      22$:  BMI      23$      ;TEST THE BMI-IT SHOULD BR
3033
3034 010706 000000      5$:  HALT      ;BMI FAILED
3035 010710 000771      BR       1$      ;ERROR LOOP RETURN
3036
3037 010712 102402      23$:  BVS      24$      ;TEST THE BVS-IT SHOULD BR
3038
3039 010714 000000      7$:  HALT      ;BVS FAILED
3040 010716 000766      BR       1$      ;ERROR LOOP RETURN
3041
3042 010720          24$:
3043 010720 103402      BCS      TST124      ;;TEST THE BCS-IT SHOULD BR
3044
3045 010722 000000      9$:  HALT      ;BCS FAILED
3046 010724 000763      BR       1$      ;ERROR LOOP RETURN
3047

```

```

3048 ;:*****
3049 ;*TEST 124 BASIC BVC TEST WITH V=1
3050 ;:*****

```

```

3051 010726          TST124:
3052 010726 012700 000124      MOV      #124,R0      ;;LOAD R0 WITH TEST NUMBER
3053
3054 010732 000262      1$:  SEV          ;MAKE V=1
3055
3056 010734 102001      2$:  BVC      3$      ;TEST THE BVC-IT SHOULDN'T BR
3057 010736 000402      BR       TST125      ;;GO TO NEXT TEST
3058
3059 010740 000000      3$:  HALT      ;BVC FAILED
3060 010742 000773      BR       1$      ;ERROR LOOP RETURN
3061

```

```

3062 ;:*****
3063 ;*TEST 125 BASIC BVC TEST WITH V=0
3064 ;:*****

```

```

3065 010744          TST125:
3066 010744 012700 000125      MOV      #125,R0      ;;LOAD R0 WITH TEST NUMBER
3067

```

```

3068 010750 000242      1$: CLV                      ;MAKE V=0
3069
3070 010752
3071 010752 102002      2$: BVC      TST126          ;;TEST THE BVC-IT SHOULD BR
3072
3073 010754 000000      3$: HALT                      ;BVC FAILED
3074 010756 000774      BR      1$                    ;ERROR LOOP RETURN
3075
3076      ;:*****
3077      ;:TEST 126      BASIC BGE TEST WITH N,V = 00
3078      ;:*****
3079 010760
3080 010760 012700 000126  †ST126: MOV      #126,RO          ;;LOAD RO WITH TEST NUMBER
3081
3082 010764 000257      1$: CCC                      ;MAKE N:C = 0000
3083
3084 010766
3085 010766 002002      2$: BGE      TST127          ;;TEST THE BGE-IT SHOULD BR
3086
3087 010770 000000      3$: HALT                      ;BGE FAILED
3088 010772 000774      BR      1$                    ;ERROR LOOP RETURN
3089
3090      ;:*****
3091      ;:TEST 127      BASIC BGE TEST WITH N,V = 01
3092      ;:*****
3093 010774
3094 010774 012700 000127  †ST127: MOV      #127,RO          ;;LOAD RO WITH TEST NUMBER
3095
3096 011000 000257      1$: CCC                      ;CLEAR FLAGS
3097 011002 000262      SEV                      ;MAKE N,V = 01
3098
3099 011004 002001      2$: BGE      3$                    ;TEST THE BGE-IT SHOULDN'T BR
3100 011006 000402      BR      TST130          ;;GO TO NEXT TEST
3101
3102 011010 000000      3$: HALT                      ;BGE FAILED
3103 011012 000772      BR      1$                    ;ERROR LOOP RETURN
3104
3105      ;:*****
3106      ;:TEST 130      BASIC BGE TEST WITH N,V = 10
3107      ;:*****
3108 011014
3109 011014 012700 000130  †ST130: MOV      #130,RO          ;;LOAD RO WITH TEST NUMBER
3110
3111 011020 000257      1$: CCC                      ;CLEAR FLAGS
3112 011022 000270      SEN                      ;MAKE N,V = 10
3113
3114 011024 002001      2$: BGE      3$                    ;TEST THE BGE-IT SHOULDN'T BR
3115 011026 000402      BR      TST131          ;;GO TO NEXT TEST
3116
3117 011030 000000      3$: HALT                      ;BGE FAILED
3118 011032 000772      BR      1$                    ;ERROR LOOP RETURN
3119
3120      ;:*****
3121      ;:TEST 131      BASIC BGE TEST WITH N,V = 11
3122      ;:*****
3123 011034  †ST131:

```

E07

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T131

MACY11 27(1006) 08-FEB-77 16:23 PAGE 59
BASIC BGE TEST WITH N,V = 11

```

3124 011034 012700 000131      MOV      #131,RO      ;;LOAD RO WITH TEST NUMBER
3125
3126 011040 000257      1$:      CCC          ;CLEAR FLAGS
3127 011042 000272          272          ;MAKE N,V = 11
3128
3129 011044          2$:
3130 011044 002002      BGE      TST132      ;;TEST THE BGE-IT SHOULD BR
3131
3132 011046 000000      3$:      HALT         ;BGE FAILED
3133 011050 000773      BR      1$          ;ERROR LOOP RETURN
3134
3135      ;;*****
3136      ;;*TEST 132      BASIC BLT TEST WITH N,V = 00
3137      ;;*****
3138 011052          †TST132:
3139 011052 012700 000132      MOV      #132,RO      ;;LOAD RO WITH TEST NUMBER
3140
3141 011056 000257      1$:      CCC          ;CLEAR FLAGS
3142
3143 011060 002401      2$:      BLT      3$          ;TEST THE BLT-IT SHOULDN'T BR
3144 011062 000402      BR      TST133      ;;GO TO NEXT TEST
3145
3146 011064 000000      3$:      HALT         ;BLT FAILED
3147 011066 000773      BR      1$          ;ERROR LOOP RETURN
3148
3149      ;;*****
3150      ;;*TEST 133      BASIC BLT TEST WITH N,V = 01
3151      ;;*****
3152 011070          †TST133:
3153 011070 012700 000133      MOV      #133,RO      ;;LOAD RO WITH TEST NUMBER
3154
3155 011074 000257      1$:      CCC          ;CLEAR FLAGS
3156 011076 000262      SEV          ;MAKE N,V = 01
3157
3158 011100          2$:
3159 011100 002402      BLT      TST134      ;;TEST THE BLT-IT SHOULD BR
3160
3161 011102 000000      3$:      HALT         ;BLT FAILED
3162 011104 000773      BR      1$          ;ERROR LOOP RETURN
3163
3164      ;;*****
3165      ;;*TEST 134      BASIC BLT TEST WITH N,V = 10
3166      ;;*****
3167 011106          †TST134:
3168 011106 012700 000134      MOV      #134,RO      ;;LOAD RO WITH TEST NUMBER
3169
3170 011112 000257      1$:      CCC          ;CLEAR FLAGS
3171 011114 000270      SEN          ;SET N - N,V = 10
3172
3173 011116          2$:
3174 011116 002402      BLT      TST135      ;;TEST THE BLT-IT SHOULD BR
3175
3176 011120 000000      3$:      HALT         ;BLT FAILED
3177 011122 000773      BR      1$          ;ERROR LOOP RETURN
3178
3179      ;;*****

```

F07

```

3180 ;*TEST 135 BASIC BLT TEST WITH N,V = 11
3181 ;*****
3182 011124 012700 000135 †ST135: MOV #135,RO ;;LOAD RO WITH TEST NUMBER
3183 011124 012700 000135 1$: CCC ;;CLEAR FLAGS
3184 011130 000257 272 ;;MAKE N,V = 11
3185 011130 000257 2$: BLT 3$ ;;TEST THE BLT-IT SHOULDN'T BR
3186 011132 000272 BR TST136 ;;GO TO NEXT TEST
3187 011134 002401 3$: HALT ;;BLT FAILED
3188 011136 000402 BR 1$ ;;ERROR LOOP RETURN
3189 011136 000402
3190 011140 000000
3191 011140 000000
3192 011142 000772
3193
3194 ;*****
3195 ;*TEST 136 BASIC BGT TEST WITH Z = 1 AND N,V = 01
3196 ;*****
3197 011144 012700 000136 †ST136: MOV #136,RO ;;LOAD RO WITH TEST NUMBER
3198 011144 012700 000136 1$: CCC ;;CLEAR FLAGS
3199 011150 000257 266 ;;SET Z AND V
3200 011150 000257 2$: BGT 3$ ;;TEST THE BGT-IT SHOULDN'T BR
3201 011152 000266 BR TST137 ;;GO TO NEXT TEST
3202 011154 003001 3$: HALT ;;BGT FAILED
3203 011156 000402 BR 1$ ;;ERROR LOOP RETURN
3204 011156 000402
3205 011160 000000
3206 011160 000000
3207 011162 000772
3208 011162 000772
3209
3210 ;*****
3211 ;*TEST 137 BASIC BGT TEST WITH Z = 0 AND N,V = 01
3212 ;*****
3213 011164 012700 000137 †ST137: MOV #137,RO ;;LOAD RO WITH TEST NUMBER
3214 011164 012700 000137 1$: CCC ;;CLEAR FLAGS
3215 011170 000257 SEV ;;SET V
3216 011172 000262 2$: BGT 3$ ;;TEST THE BGT-IT SHOULD NOT BR
3217 011174 003001 BR TST140 ;;GO TO SCOPE LOOP EXIT
3218 011176 000402 3$: HALT ;;BGT FAILED
3219 011176 000402 BR 1$ ;;ERROR LOOP RETURN
3220 011200 000000
3221 011200 000000
3222 011202 000772
3223 011202 000772
3224
3225 ;*****
3226 ;*TEST 140 BASIC BGT TEST WITH Z = 1 AND N,V = 00
3227 ;*****
3228 011204 012700 000140 †ST140: MOV #140,RO ;;LOAD RO WITH TEST NUMBER
3229 011204 012700 000140 1$: CCC ;;CLEAR FLAGS
3230 011210 000257 SEZ ;;SET Z
3231 011212 000264 2$: BGT 3$ ;;TEST THE BGT-IT SHOULD NOT BR
3232 011214 003001 BR TST141 ;;GO TO SCOPE LOOP EXIT
3233 011216 000402
3234 011216 000402
3235 011216 000402
  
```

```

3236 011220 000000 3S: HALT ;BGT FAILED
3237 011222 000772 BR 1S ;ERROR LOOP RETURN
3238
3239 ;*****
3240 ;*TEST 141 BASIC BGT TEST WITH Z = 0 AND N,V = 00
3241 ;*****
3242 011224 TST141:
3243 011224 012700 000141 MOV #141,RO ;;LOAD RO WITH TEST NUMBER
3244
3245 011230 000257 1S: CCC ;CLEAR FLAGS
3246
3247 011232 2S: BGT TST142 ;;TEST THE BGT - IT SHOULD BR
3248 011232 003002
3249
3250 011234 000000 3S: HALT ;BGT FAILED
3251 011236 000774 BR 1S ;ERROR LOOP RETURN
3252
3253 ;*****
3254 ;*TEST 142 BASIC BGT TEST WITH Z = 1 AND N,V = 01
3255 ;*****
3256 011240 TST142:
3257 011240 012700 000142 MOV #142,RO ;;LOAD RO WITH TEST NUMBER
3258
3259 011244 1S: CCC ;CLEAR FLAGS
3260 011246 000266 266 ;MAKE N,V = 01 AND Z = 1
3261
3262 011250 2S: BGT 3S ;TEST THE BGT-IT SHOULDN'T BR
3263 011252 000402 BR TST143 ;;GO TO NEXT TEST
3264
3265 011254 3S: HALT ;BGT FAILED
3266 011256 000772 BR 1S ;ERROR LOOP RETURN
3267
3268 ;*****
3269 ;*TEST 143 BASIC BGT TEST WITH Z = 1 AND N,V = 10
3270 ;*****
3271 011260 TST143:
3272 011260 012700 000143 MOV #143,RO ;;LOAD RO WITH TEST NUMBER
3273
3274 011264 1S: CCC ;CLEAR FLAGS
3275 011266 000274 274 ;MAKE Z = 1 AND N,V = 10
3276
3277 011270 2S: BGT 3S ;TEST THE BLT-IT SHOULDN'T BR
3278 011272 000402 BR TST144 ;;GO TO NEXT TEST
3279
3280 011274 3S: HALT ;BLT FAILED
3281 011276 000772 BR 1S ;ERROR LOOP RETURN
3282
3283 ;*****
3284 ;*TEST 144 BASIC BGT TEST WITH Z = 1 AND N,V = 11
3285 ;*****
3286 011300 TST144:
3287 011300 012700 000144 MOV #144,RO ;;LOAD RO WITH TEST NUMBER
3288
3289 011304 1S: CCC ;CLEAR FLAGS
3290 011306 000276 276 ;MAKE Z = 1 AND N,V = 11
3291

```

3292 011310 003001
 3293 011312 000402
 3294
 3295 011314 000000
 3296 011316 000772
 3297
 3298
 3299
 3300
 3301 011320
 3302 011320 012700 000145
 3303
 3304 011324 000257
 3305 011326 000272
 3306
 3307 011330
 3308 011330 003002
 3309
 3310 011332 000000
 3311 011334 000773
 3312
 3313
 3314
 3315
 3316 011336
 3317 011336 012700 000146
 3318
 3319 011342 000257
 3320
 3321 011344
 3322 011344 101002
 3323
 3324 011346 000000
 3325 011350 000774
 3326
 3327
 3328
 3329
 3330 011352
 3331 011352 012700 000147
 3332
 3333 011356 000257
 3334 011360 000261
 3335
 3336 011362 101001
 3337 011364 000402
 3338
 3339 011366 000000
 3340 011370 000772
 3341
 3342
 3343
 3344
 3345 011372
 3346 011372 012700 000150
 3347

```

2$:  BGT      3$          ;TEST THE BGT-IT SHOULD NOT BR
    BR      TST145      ;;GO TO NEXT TEST

3$:  HALT     1$          ;BLT FAILED
    BR      1$          ;ERROR LOOP RETURN

*****
;*TEST 145      BASIC BGT TEST WITH Z=0 AND N,V=11
*****
TST145:
    MOV      #145,RO      ;;LOAD RO WITH TEST NUMBER

1$:  CCC      272        ;CLEAR FLAGS
    SEC                      ;MAKE N:C=1010

2$:  BGT      TST146      ;;TEST THE BGT - IT SHOULD BR

3$:  HALT     1$          ;BGT FAILED
    BR      1$          ;ERROR LOOP RETURN

*****
;*TEST 146      BASIC BHI TEST WITH Z,C = 00
*****
TST146:
    MOV      #146,RO      ;;LOAD RO WITH TEST NUMBER

1$:  CCC      272        ;MAKE Z,C = 00

2$:  BHI      TST147      ;;TEST THE BHI-IT SHOULD BR

3$:  HALT     1$          ;BHI FAILED
    BR      1$          ;ERROR LOOP RETURN

*****
;*TEST 147      BASIC BHI TEST WITH Z,C = 01
*****
TST147:
    MOV      #147,RO      ;;LOAD RO WITH TEST NUMBER

1$:  CCC      SEC        ;CLEAR FLAGS
    SEC                      ;MAKE Z,C = 01

2$:  BHI      3$          ;TEST THE BHI-IT SHOULD NOT BR
    BR      TST150      ;;GO TO NEXT TEST

3$:  HALT     1$          ;BHI FAILED
    BR      1$          ;ERROR LOOP RETURN

*****
;*TEST 150      BASIC BHI TEST WITH Z,C = 10
*****
TST150:
    MOV      #150,RO      ;;LOAD RO WITH TEST NUMBER
  
```

```

3348 011376 000257      1$:   CCC           ;CLEAR FLAGS
3349 011400 000264      SEZ           ;MAKE Z,C = 10
3350
3351 011402 101001      2$:   BHI           3$           ;TEST THE BHI-IT SHOULD NOT BR
3352 011404 000402      BR           TST151           ;;GO TO NEXT TEST
3353
3354 011406 000000      3$:   HALT          ;BHI FAILED
3355 011410 000772      BR           1$           ;ERROR LOOP RETURN
3356
3357      ;*****
3358      ;*TEST 151      BASIC BHI TEST WITH Z,C = 11
3359      ;*****
3360      †TST151:
3361 011412 012700 000151      MOV          #151,R0           ;;LOAD R0 WITH TEST NUMBER
3362
3363 011416 000257      1$:   CCC           ;CLEAR FLAGS
3364 011420 000265      265          ;MAKE Z,C = 11
3365
3366 011422 101001      2$:   BHI           3$           ;TEST THE BHI-IT SHOULDN'T BR
3367 011424 000402      BR           TST152           ;;GO TO NEXT TEST
3368
3369 011426 000000      3$:   HALT          ;BHI FAILED
3370 011430 000772      BR           1$           ;ERROR LOOP RETURN
3371
3372      ;*****
3373      ;*TEST 152      BASIC NEG MODE 0 TEST : (DEST) GT 0
3374      ;*****
3375      †TST152:
3376 011432 012700 000152      MOV          #152,R0           ;;LOAD R0 WITH TEST NUMBER
3377 011436 012704 177776      MOV          #-2,R4           ;RESULT S / B = 177776
3378 011442 012703 000002      1$:   MOV          #2,R3           ;INITIAL (DEST) = 2
3379 011446 000257      CCC           ;CLEAR FLAGS
3380 011450 000266      266          ;MAKE N:C = 0110
3381
3382 011452 005403      2$:   NEG          R3           ;TEST THE NEG
3383
3384 011454 100003      BPL          3$
3385 011456 001402      BEQ          3$           ;DID N:C = 1001?
3386 011460 102401      BVS          3$
3387 011462 103402      BCS          4$
3388
3389 011464 000000      3$:   HALT          ;NEGATE FAILED TO ALTER CODES PROPERLY
3390 011466 000765      BR           1$           ;ERROR LOOP RETURN
3391
3392 011470 020304      4$:   CMP          R3,R4           ;CORRECT RESULT?
3393 011472 001402      BEQ          TST153           ;;BR IF YES
3394
3395 011474 000000      5$:   HALT          ;NEG DELIVERED WRONG RESULT
3396 011476 000761      BR           1$           ;ERROR LOOP RETURN
3397
3398      ;*****
3399      ;*TEST 153      BASIC "SUB #,2#" TEST
3400      ;*****
3401      †TST153:
3402 011500 012700 000153      MOV          #153,R0           ;;LOAD R0 WITH TEST NUMBER
3403 011504 012704 000002      MOV          #2,R4           ;RESULT S / B = 2

```



```

3404 011510 012702 063236          MOV    #MBUFO,R2      ;R2 POINTS TO DEST
3405 011514 012712 000004          MOV    #4,(R2)       ;INITIAL (DEST) = 4
3406 011520 000257                   CCC                   ;CLEAR FLAGS
3407
3408 011522 162737 000002 063236 2$:  SUB    #2,2#MBUFO    ;TEST THE SUB
3409
3410 011530 020412                   CMP    R4,(R2)       ;RESULT=2?
3411 011532 001403                   BEQ    TST154        ;BR IF YES
3412 011534 011203                   MOV    (R2),R3       ;GET WAS DATA
3413 011536 000000          3$:  HALT                   ;SUB DELIVERED WRONG RESULT
3414 011540 000765                   BR     1$            ;ERROR LOOP RETURN
3415
3416                                     ;*****
3417                                     ;*TEST 154 BASIC "SUB 2#,RN" TEST
3418                                     ;*****
3419 TST154:
3420 011542 012700 000154          MOV    #154,R0       ;:LOAD R0 WITH TEST NUMBER
3421 011546 012737 000002 063236 1$:  MOV    #2,2#MBUFO    ;SRC = 2
3422 011554 012703 000004          MOV    #4,R3         ;INITIAL (DEST) = 4
3423 011560 000257                   CCC                   ;SCOPE SYNC
3424
3425 011562 163703 063236 2$:  SUB    2#MBUFO,R3    ;TEST THE SUB
3426
3427 011566 020403                   CMP    R4,R3         ;RESULT=2?
3428 011570 001402                   BEQ    TST155        ;:BR IF YES
3429
3430 011572 000000          3$:  HALT                   ;SUB DELIVERED WRONG RESULT
3431 011574 000767                   BR     1$            ;ERROR LOOP RETURN
3432
3433                                     ;*****
3434                                     ;*TEST 155 BASIC "RTS RN" TEST - <N:C> = 1111
3435                                     ;*****
3436 TST155:
3437 011576 012700 000155          MOV    #155,R0       ;:LOAD R0 WITH TEST NUMBER
3438 011602 010605                   MOV    SP,R5         ;SAVE SP
3439 011604 010506          1$:  MOV    R5,SP         ;RESET SP FOR ERROR LOOP
3440 011606 012703 011626          MOV    #4$,R3        ;RTS SHOULD LOAD PC FROM (R3)
3441 011612 012746 177777          MOV    #-1,-(SP)     ;RTS SHOULD LOAD R3 FROM STACK
3442 011616 000277                   SCC                   ;N:C = 1111
3443
3444 011620 000203          2$:  RTS    R3          ;TEST THE RTS - GO TO 4$
3445
3446 011622 000000          3$:  HALT                   ;RTS FAILED TO LOAD THE PC
3447 011624 000767                   BR     1$            ;LOCK ON ERROR
3448
3449 011626 100003          4$:  BPL    5$            ;N:C = 1111 ?
3450 011630 001002                   BNE    5$
3451 011632 102001                   BVC    5$
3452 011634 103402                   BCS    6$
3453
3454 011636 000000          5$:  HALT                   ;RTS ALTERED CODES - CLEARED ONE
3455 011640 000761                   BR     1$            ;LOCK ON ERROR
3456
3457 011642 020327 177777          6$:  CMP    R3,#-1     ;DID R3 GET LOADED FROM STACK ?
3458 011646 001402                   BEQ    8$            ;BR IF YES
3459
    
```

```

3460 011650 000000      7$:  HALT          ;RTS FAILED TO LOAD REG
3461 011652 000754      BR          1$      ;LOCK ON ERROR
3462
3463 011654 020506      8$:  CMP          R5,SP  ;DID RTS POP THE STACK POINTER ?
3464 011656 001402      BEQ          T$T156  ;;BR IF YES
3465
3466 011660 000000      9$:  HALT          ;RTS FAILED TO POP SP
3467 011662 000750      BR          1$      ;LOCK ON ERROR
3468

```

```

;*****
;#TEST 156 BASIC "RTS PC" TEST
;*****

```

```

3472 011664          T$T156:
3473 011664 012700 000156      MOV          #156,R0  ;;LOAD R0 WITH TEST NUMBER
3474 011670 010605          MOV          SP,R5   ;SAVE THE ORIGINAL SP
3475 011672 010506      1$:  MOV          R5,SP   ;RESET SP FOR ERROR LOOP
3476 011674 012746 011710      MOV          #4$,-(SP) ;PUSH NEW PC ON STACK
3477 011700 000257          CCC          ;SCOPE SYNC
3478
3479 011702 000207      2$:  RTS          PC    ;TEST THE RTS - GO TO 4$
3480
3481 011704 000000      3$:  HALT          ;RTS FAILED TO LOAD PC
3482 011706 000771      BR          1$      ;LOCK ON HARD ERROR
3483
3484 011710 020605      4$:  CMP          SP,R5   ;DID SP GET POPPED ?
3485 011712 001402      BEQ          T$T157  ;;BR IF YES
3486
3487 011714 000000      5$:  HALT          ;RTS FAILED TO UPDATE SP
3488 011716 000765      BR          1$      ;LOCK ON HARD ERROR
3489

```

```

;*****
;#TEST 157 BASIC "JSR PC,2#A" TEST
;*****

```

```

3493 011720          T$T157:
3494 011720 012700 000157      MOV          #157,R0  ;;LOAD R0 WITH TEST NUMBER
3495          .SBTTL USER CONTROLLED BREAKPOINT -- BIT3
3496 011724 032737 000010 063160  BIT          #BIT3,2#BPTLOC ;BREAKPOINT HALT SET ??
3497 011732 001401          BEQ          .+4      ;BR IF NOT
3498 011734 000000          HALT          ;BREAK - DEPRESS CONTINUE TO RESTART
3499 011736 010605          MOV          SP,R5   ;SAVE ORIGINAL SP
3500 011740 010506      1$:  MOV          R5,SP   ;RESET SP FOR ERROR LOOP
3501 011742 000257          CCC          ;SCOPE SYNC
3502
3503 011744 004737 011754      2$:  JSR          PC,2#4$ ;TEST THE JSR - GO TO 4$
3504
3505 011750 000000      3$:  HALT          ;JSR FAILED TO LOAD PC
3506 011752 000772      BR          1$      ;LOCK ON HARD ERROR
3507
3508 011754 022726 011750      4$:  CMP          #3$, (SP)+ ;DID JSR SAVE OLD PC ON STACK ?
3509 011760 001402      BEQ          T$T160  ;;BR IF YES
3510
3511 011762 000000      5$:  HALT          ;JSR FAILED TO SAVE OLD PC
3512 011764 000765      BR          1$      ;LOCK ON HARD ERROR
3513

```

```

;*****
;#TEST 160 BASIC "RTI" TEST - N:C=0000
;*****

```

```

3514
3515

```

3516
3517 011766
3518 011766 012700 000160
3519 011772 010605
3520 011774 010506
3521 011776 012746 000357
3522 012002 012746 012022
3523 012006 005037 177776
3524 012012 000257
3525
3526 012014 000002
3527
3528 012016 000000
3529 012020 000765
3530
3531 012022 013702 177776
3532 012026 022702 000357
3533 012032 001404
3534
3535 012034 010237 177776
3536 012040 000000
3537 012042 000754
3538
3539 012044 020605
3540 012046 001402
3541
3542 012050 000000
3543 012052 000750
3544
3545
3546
3547
3548 012054
3549 012054 012700 000161
3550 012060 010605
3551 012062 010506
3552 012064 005046
3553 012066 012746 012104
3554 012072 012737 000357 177776
3555 012100 000240
3556
3557 012102 000002
3558
3559 012104 013702 177776
3560 012110 022702 000000
3561 012114 001404
3562
3563 012116 010237 177776
3564 012122 000000
3565 012124 000756
3566
3567
3568
3569
3570 012126
3571 012126 012700 000162

```
*****
TST160:
MOV #160,R0 ;:LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;:SAVE THE SP
1$: MOV R5,SP ;:RESET THE SP FOR ERROR LOOP
MOV #357,-(SP) ;:NEW PSW = 357
MOV #4$,-(SP) ;:NEW PC = 4$
CLR @#PSW ;:MAKE [PSW] = 000
CCC ;:MAKE N:C=0000

2$: RTI ;:TEST THE RTI - GO TO 4$

3$: HALT ;:RTI FAILED TO LOAD PC
BR 1$ ;:LOOP ON HARD ERROR

4$: MOV @#PSW,R2 ;:SAVE THE [PSW] IN R2
CMP #357,R2 ;:WAS [PSW] = 357 ?
BEQ 6$ ;:BR IF YES

5$: MOV R2,@#PSW ;:RESTORE THE ERROR PSW
HALT ;:RTI FAILED TO LOAD PSW
BR 1$ ;:LOCK ON HARD ERROR

6$: CMP SP,R5 ;:DID SP GET UPDATED OK ?
BEQ TST161 ;:BR IF YES

7$: HALT ;:RTI FAILED TO UPDATE THE SP
BR 1$ ;:LOCK ON HARD ERROR

*****
;:TEST 161 BASIC "RTI" TEST WITH N:C=1111
*****
TST161:
MOV #161,R0 ;:LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;:SAVE THE SP IN R5
1$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
CLR -(SP) ;:NEW PSW = 000000
MOV #4$,-(SP) ;:NEW PC = 4$
MOV #357,@#PSW ;:MAKE OLD PSW = 357
NOP ;:SCOPE SYNC

2$: RTI ;:TEST THE RTI - GO TO 4$

4$: MOV @#PSW,R2 ;:GET THE PSW
CMP #0,R2 ;:WAS [PSW]=000
BEQ TST162 ;:BR IF YES

3$: MOV R2,@#PSW ;:RESTORE ERROR PSW
HALT ;:RTI FAILED TO CLEAR PSW
BR 1$ ;:LOCK ON HARD ERROR

*****
;:TEST 162 BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357
*****
TST162:
MOV #162,R0 ;:LOAD R0 WITH TEST NUMBER
```

M07

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T162

MACY11 27(1006) 08-FEB-77 16:23 PAGE 67
 BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357

```

3572 012132 010605          MOV      SP,R5          ;SAVE THE SP
3573 012134 010506          MOV      R5,SP         ;RESET SP FOR ERROR LOOP
3574 012136 012737 012174 000020 1S:  MOV      #4,@#20       ;SET UP IOT VECTOR
3575 012144 012737 000357 000022  MOV      #357,@#22     ;
3576 012152 012766 177777 177776  MOV      #-1,-2(SP)    ;IOT SHOULD CHANGE -1 TO 0
3577 012160 005037 177776  CLR      @#PSW         ;MAKE [PSW] = 000
3578 012164 000257          CCC                    ;SCOPE SYNC
3579
3580 012166 000004          2S:  IOT                ;TEST THE IOT
3581
3582 012170 000000          3S:  HALT               ;IOT FAILED TO LOAD PC
3583 012172 000760          BR      1S             ;LOCK ON HARD ERROR
3584
3585 012174 013702 177776 4S:  MOV      @#PSW,R2     ;GET THE PSW
3586 012200 022702 000357  CMP      #357,R2       ;DID IOT LOAD A 357 ?
3587 012204 001404          BEQ     6S             ;BR IF YES
3588
3589 012206 010237 177776 5S:  MOV      R2,@#PSW     ;RESTORE ERROR PSW
3590 012212 000000          HALT               ;IOT FAILED TO LOAD PSW
3591 012214 000747          BR      1S             ;LOCK ON HARD ERROR
3592
3593 012216 022726 012170 6S:  CMP      #35,(SP)+    ;DID IOT SAVE OLD PC ?
3594 012222 001404          BEQ     8S             ;BR IF YES
3595
3596 012224 010237 177776 7S:  MOV      R2,@#PSW     ;RESTORE ERROR PSW
3597 012230 000000          HALT               ;IOT FAILED TO SAVE OLD PC
3598 012232 000740          BR      1S             ;LOCK ON HARD ERROR
3599
3600 012234 005726 8S:  TST      (SP)+        ;DID IOT SAVE OLD PSW ?
3601 012236 001404          BEQ     TST163        ;;BR IF YES
3602
3603 012240 010237 177776 9S:  MOV      R2,@#PSW     ;RESTORE ERROR PSW
3604 012244 000000          HALT               ;IOT FAILED TO SAVE OLD PSW
3605 012246 000732          BR      1S             ;LOCK ON HARD ERROR
3606
3607 ;*****
3608 ;*TEST 163 BASIC "IOT" TEST - VERIFY LINKAGE TO SCOPE SERVICE
3609 ;*****
3610 012250          TST163:
3611 012250 012700 000163 1S:  MOV      #163,R0       ;LOAD R0 WITH TEST NUMBER
3612 012254 010605          MOV      SP,R5         ;SAVE SP
3613 012256 010506          MOV      R5,SP         ;RESET SP FOR ERROR LOOP
3614 012260 005037 063170  CLR      @#SCOFLG      ;TRAP SERVICE WILL COM "SCOFLG"
3615 012264 012737 061536 000020  MOV      #SCOPEA,@#20  ;SET UP IOT VECTOR
3616 012272 005037 000022  CLR      @#22
3617 012276 000257          CCC                    ;SCOPE SYNC
3618
3619 012300 000004          2S:  SCOPE                ;TEST THE IOT
3620
3621 012302 005137 063170  COM      @#SCOFLG      ;SCOFLG SHOULD BECOME 000000
3622 012306 001402          BEQ     4S             ;BR IF IT DID
3623
3624 012310 000000          3S:  HALT               ;IOT FAILED TO LINK TO SCOPE SERVICE
3625 012312 000761          BR      1S             ;LOCK ON HARD ERROR
3626
3627 012314 010506          4S:  MOV      R5,SP         ;RESET SP IN CASE OF ERROR
    
```

```

3628
3629
3630
3631 012316
3632 012316 012700 000164
3633 012322 010605
3634 012324 010506
3635 012326 012737 012364 000020
3636 012334 012737 000357 000022
3637 012342 012766 177777 177776
3638 012350 005037 177776
3639 012354 000257
3640
3641 012356 000004
3642
3643 012360 000000
3644 012362 000760
3645
3646 012364 013702 177776
3647 012370 022702 000357
3648 012374 001404
3649
3650 012376 010237 177776
3651 012402 000000
3652 012404 000747
3653
3654 012406 022726 012360
3655 012412 001404
3656
3657 012414 010237 177776
3658 012420 000000
3659 012422 000740
3660
3661 012424 005726
3662 012426 001404
3663
3664 012430 010237 177776
3665 012434 000000
3666 012436 000732
3667
3668
3669
3670
3671 012440
3672 012440 012700 000165
3673 012444 010605
3674 012446 010506
3675 012450 012737 012474 000020
3676 012456 005037 000022
3677 012462 012737 000340 177776
3678 012470 000277
3679
3680 012472 000004
3681
3682 012474 013702 177776
3683 012500 001404

```

```

*****
;TEST 164 BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357
*****
†ST164:
MOV #164,R0 ;:LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;:SAVE THE SP
1$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
MOV #4$,J#20 ;:SET UP IOT VECTOR
MOV #357,J#22
MOV #-1,-2(SP) ;:IOT SHOULD CHANGE -1 TO 0
CLR J#PSW ;:MAKE [PSW] = 000
CCC ;:SCOPE SYNC

2$: IOT ;:TEST THE IOT

3$: HALT ;:IOT FAILED TO LOAD PC
BR 1$ ;:LOCK ON HARD ERROR

4$: MOV J#PSW,R2 ;:GET THE PSW
CMP #357,R2 ;:DID IOT LOAD A 357 ?
BEQ 6$ ;:BR IF YES

5$: MOV R2,J#PSW ;:RESTORE ERROR PSW
HALT ;:IOT FAILED TO LOAD PSW
BR 1$ ;:LOCK ON HARD ERROR

6$: CMP #3$, (SP)+ ;:DID IOT SAVE OLD PC ?
BEQ 8$ ;:BR IF YES

7$: MOV R2,J#PSW ;:RESTORE ERROR PSW
HALT ;:IOT FAILED TO SAVE OLD PC
BR 1$ ;:LOCK ON HARD ERROR

8$: TST (SP)+ ;:DID IOT SAVE OLD PSW ?
BEQ TST165 ;:BR IF YES

9$: MOV R2,J#PSW ;:RESTORE ERROR PSW
HALT ;:IOT FAILED TO SAVE OLD PSW
BR 1$ ;:LOCK ON HARD ERROR

*****
;TEST 165 BASIC IOT TEST - VERIFY LOADING PSW WITH 000
*****
†ST165:
MOV #165,R0 ;:LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;:SAVE THE SP
1$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
MOV #4$,J#20 ;:SET UP IOT VECTOR
CLR J#22
MOV #340,J#PSW ;:MAKE [PSW] = 340
SCC ;:MAKE N:C=1111

2$: IOT ;:TEST THE IOT

4$: MOV J#PSW,R2 ;:GET THE [PSW]
BEQ 6$ ;:BR IF [PSW] = 000

```

```

3684
3685 012502 010237 177776
3686 012506 000000
3687 012510 000756
3688
3689 012512 010506
3690
3691
3692
3693
3694 012514
3695 012514 012700 000166
3696 012520 010605
3697 012522 010506
3698 012524 012737 062126 000034
3699 012532 005037 000036
3700 012536 005037 063162
3701 012542 000257
3702
3703 012544 104401
3704
3705 012546 005137 063162
3706 012552 001402
3707
3708 012554 000000
3709 012556 000761
3710
3711
3712
3713
3714 012560
3715 012560 012700 000167
3716 012564 010605
3717 012566 010506
3718 012570 012737 061764 000030
3719 012576 005037 000032
3720 012602 005037 063164
3721 012606 000257
3722
3723 012610 104000
3724
3725 012612 005137 063164
3726 012616 001402
3727
3728 012620 000000
3729 012622 000761
3730
3731
3732
3733 012624
3734 012624 012700 000170
3735 012630 010605
3736 012632 012737 061040 000010
3737 012640 012737 000340 000012
3738 012646 010506
3739 012650 005037 063172

3S:  MOV      R2, @#PSW      ;RESTORE THE ERROR PSW
      HALT
      BR       1S            ;IOT FAILED TO CLEAR THE PSW
                                ;LOCK ON HARD ERROR

6S:  MOV      RS, SP        ;RESET THE SP BEFORE CONTINUING

;*****
;*TEST 166      BASIC "TRAP" TEST - LINKAGE TO PRINT ROUTINE
;*****
TST166:
      MOV      @#166, R0     ;:LOAD R0 WITH TEST NUMBER
      MOV      SP, RS       ;:SAVE THE SP
      MOV      RS, SP       ;:RESET SP FOR ERROR LOOP
      MOV      @#PRINA, @#34 ;:SET UP THE "TRAP" VECTOR
      CLR      @#36
      CLR      @#PRIFLG     ;:INITIALIZE TEST FLAG
      CCC
                                ;:SCOPE SYNC

2S:  TYPE
                                ;:TEST THE TRAP

      COM      @#PRIFLG     ;:SHOULD MAKE [PRIFLG] = 000000
      BEQ      TST167      ;:BR IF IT DID

3S:  HALT
      BR       1S            ;:TRAP FAILED TO LINK TO PRINT SERV.
                                ;:LOCK ON HARD ERROR

;*****
;*TEST 167      BASIC "EMT" TEST - LINKAGE TO ERROR SERVICE
;*****
TST167:
      MOV      @#167, R0     ;:LOAD R0 WITH TEST NUMBER
      MOV      SP, RS       ;:SAVE THE SP
      MOV      RS, SP       ;:RESET SP FOR ERROR LOOP
      MOV      @#ERRA, @#30 ;:SET UP THE EMT VECTOR
      CLR      @#32
      CLR      @#ERRFLG     ;:EMT SERVICE WILL COM [ERRFLG]
      CCC
                                ;:SCOPE SYNC

2S:  ERROR
                                ;:TEST THE EMT

      COM      @#ERRFLG     ;:DID EMT SERV. COM ERRFLG?
      BEQ      TST170      ;:BR IF YES

3S:  HALT
      BR       1S            ;:EMT DID NOT LINK PROPERLY
                                ;:LOCK ON HARD ERROR

;*****
;*TEST 170      BASIC TEST OF RSVD INSTR. TRAP LINKAGE
;*****
TST170:
      MOV      @#170, R0     ;:LOAD R0 WITH TEST NUMBER
      MOV      SP, RS       ;:SAVE THE SP
      MOV      @#RSVTST, @#10 ;:SET UP RSVD INSTR. TRAP VECTOR
      MOV      @#340, @#12
      MOV      RS, SP       ;:RESET SP FOR ERROR LOOP
      CLR      @#RSVFLG     ;:INITIALIZE TEST FLAG THAT WILL GET

```

```

3740                                     ;COMPLEMENTED BY TRAP SERVICE
3741 012654 000257                       CCC                       ;SCOPE SYNC
3742                                     ;
3743 012656 000007                       2$: 000007                 ;FORCE RSVD INSTR. TRAP
3744                                     ;
3745 012660 005137 063172                 COM      @#RSVFLG         ;TEST FLAG SHOULD GO TO 000000
3746 012664 001402                       BEQ      4$              ;BR IF TRAP SPRUNG
3747                                     ;
3748 012666 000000                       3$: HALT                  ;RSVD INSTR. TRAP FAILED
3749 012670 000766                       BR       1$              ;LOCK ON HARD ERROR
3750                                     ;
3751 012672 012737 061046 000010         4$: MOV      @#RSERR,@#10 ;SET UP RSVD INSTR TRAP VECTOR TO POINT
3752 012700 012737 000340 000012         MOV      @340,@#12      ;TO ERROR SERVICE ROUTINE
3753                                     ;
3754                                     ;*****
3755                                     ;*TEST 171 BASIC TEST OF BUS TIMEOUT TRAP LINKAGE
3756                                     ;*****
3757 012706                                     †ST171:
3758 012706 012700 000171                 MOV      @171,R0        ;:LOAD R0 WITH TEST NUMBER
3759 012712 010605                       MOV      SP,R5          ;:SAVE THE SP
3760 012714 012737 061136 000004         MOV      @#ETST,@#4     ;:SET UP THE BUS ERROR VECTOR
3761 012722 012737 000340 000006         MOV      @340,@#6
3762 012730 010506                       1$: MOV      R5,SP       ;:RESET SP FOR ERROR LOOP
3763 012732 005037 063174                 CLR      @#BERFLG      ;:INITIALIZE TEST FLAG THAT WILL GET
3764                                     ;:COMPLEMENTED BY TRAP SERVICE
3765 012736 000257                       CCC                       ;:SCOPE SYNC
3766                                     ;
3767 012740 005737 177700                 2$: TST      @#177700   ;:FORCE BUS TIMEOUT USING R0 ADDR.
3768                                     ;
3769 012744 005137 063174                 COM      @#BERFLG      ;:TEST FLAG SHOULD GO TO 000000
3770 012750 001402                       BEQ      TST172         ;:BR IF TRAP SPRUNG
3771                                     ;
3772 012752 000000                       3$: HALT                  ;:BUS ERROR FAILED TO SPRING TRAP
3773 012754 000765                       BR       1$              ;:LOCK ON HARD ERROR
3774                                     ;
3775                                     ;*****
3776                                     ;*TEST 172 BASIC TEST FOR ACCESSING DL11 REGISTERS
3777                                     ;*****
3778 012756                                     †ST172:
3779 012756 012700 000172                 MOV      @172,R0        ;:LOAD R0 WITH TEST NUMBER
3780 012762 005067 050250                 CLR      MBUFO          ;:INIT STALL COUNTER
3781 012766 005367 050244                 11$: DEC     MBUFO       ;:COUNT THE TIMER
3782 012772 001375                       BNE     11$             ;:BR IF NO-TIMEOUT
3783 012774 012737 013034 000004         MOV      @3$,@#4 ;SET UP ;BUS TIMEOUT VECTOR
3784 013002 012737 000340 000006         MOV      @340,@#6
3785 013010 010605                       MOV      SP,R5          ;:SAVE TH SP
3786 013012 010506                       1$: MOV      R5,SP       ;:RESET SP FOR ERROR LOOP
3787 013014 012702 177560                 MOV      @RCSR,R2      ;:[R2] = STARTING DL11 ADDR.
3788 013020 000257                       CCC                       ;:SCOPE SYNC
3789                                     ;
3790 013022 005722                       2$: TST      (R2)+       ;:REFERENCE DL11 - RCSR
3791 013024 005722                       TST      (R2)+         ;:REFERENCE DL11 - RDBR
3792 013026 005722                       TST      (R2)+         ;:REFERENCE DL11 - XCSR
3793 013030 005712                       TST      (R2)          ;:REFERENCE DL11 - XDBR
3794                                     ;
3795 013032 000403                       BR       4$              ;GO TO NEXT TEST

```

```

3796
3797 013034 005742      3S:  TST      -(R2)          ;BAD ADDRESS IN R2
3798 013036 000000      HALT
3799 013040 000764      BR        1S          ;ONE OF DL11 ADDR'S CAUSED TIME OUT
                          ;LOCK ON HARD ERROR
3800
3801 013042 012737 061144 000004 4S:  MOV      #BERR, R4      ;SET UP BUS ERROR VECTOR TO POINT
3802 013050 012737 000340 000006  MOV      #340, R6      ;TO ERROR SERVICE ROUTINE
3803
3804
3805
3806 013056
3807 013056 012700 000173      ;*****
3808 013062 012702 177564      ;*TEST 173 BASIC TEST OF DL11 - XCSR - READY(1)
3809 013066 012704 000200      ;*****
3810 013072 005012      TST173:
3811 013074 005001      MOV      #173, R0      ;:LOAD R0 WITH TEST NUMBER
3812 013076 000257      MOV      #XCSR, R2     ;:DEST ADDR = XCSR
                          MOV      #200, R4      ;:RESULT S / B = 200
3813
3814 013100 020412      1S:  CLR      (R2)         ;:CLEAR (DEST)
3815
3816 013102 001405      CLR      R1           ;:SET UP TIMEOUT COUNTER
3817 013104 005301      CCC
3818 013106 001374      ;:SCOPE SYNC
3819
3820 013110 011203      2S:  CMP      R4, (R2)   ;:TEST READY BIT - IT SHOULD BE SET
3821 013112 000000      BEQ      TST174       ;:BR IF IT WAS
3822 013114 000766      DEC      R1           ;:TICK-TOCK GOES THE TIMER
                          BNE      2S          ;:BR IF NOT A TIMEOUT
3823
3824
3825
3826
3827 013116
3828 013116 012700 000174      3S:  MOV      (R2), R3   ;:GET THE WAS DATA
3829 013122 012702 177564      HALT
3830 013126 012704 000200      BR        1S          ;:READY BIT IN XCSR FAILED ON A (0)
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846 013116
3847 013116 012700 000174      ;*****
3848 013122 012702 177564      ;*TEST 174 BASIC TEST OF DL11 - XCSR - MAINT BIT (0)
3849 013126 012704 000200      ;*****
3850 013132 005012      TST174:
3851 013134 000257      MOV      #174, R0      ;:LOAD R0 WITH TEST NUMBER
                          MOV      #XCSR, R2     ;:DEST ADDR = XCSR
                          MOV      #200, R4      ;:RESULT S / B = 200
3852
3853 013136 020412      1S:  CLR      (R2)         ;:CLEAR MAINT. BIT
3854
3855 013140 001403      CCC
3856
3857 013142 011203      2S:  CMP      R4, (R2)   ;:TEST MAINT(0)
3858 013144 000000      BEQ      TST175       ;:BR IF MAINT BIT CLEAR
3859 013146 000771      MOV      (R2), R3     ;:GET THE WAS DATA
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000

```


E08

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.A.P11 08-FEB-77 16:17 T175

MACY11 27(1006) 08-FEB-77 16:23 PAGE 72
 BASIC TEST OF DL11 XCSR - MAINT BIT = 1

3852 013172 020412
 3853
 3854 013174 001403
 3855
 3856 013176 011203
 3857 013200 000000
 3858 013202 000770
 3859
 3860
 3861
 3862
 3863
 3864
 3865
 3866
 3867
 3868
 3869
 3870

```

2S:  CMP      R4,(R2)      ;TEST MAINT.(1)
      BEQ      TST176      ;;BR IF IT WAS
3S:  MOV      (R2),R3      ;GET THE WAS DATA
      HALT                      ;CAN'T SET MAINT BIT IN XCSR
      BR       1S           ;LOCK 0 HARD ERROR
  
```

```

*****
;TEST 176 BASIC DL11 OUT / IN ECHO TEST (MAINT MODE)
;THIS ROUTINE USES THE MAINTENANCE MODE FEATURE OF THE DL11 TO
;TURN AROUND A STRING OF 8 CHARACTERS TO THE DL11. THIS STRING CONSISTS
;OF ALTERNATING NULL / DELETE CHARS WHICH ARE NON PRINTING. THE 8 CHARS
;ARE OUTPUT THEN READ BACK INTO A CORE BUFFER AND THEN THE INPUT AND
;OUTPUT CORE BUFFERS ARE CHECKED FOR EQUIVALENCE. IF AN ERROR IS DET-
;ECTED DURING THE COMPARISON THE ROUTINE HALTS WITH THE WAS AND S / B
;DATA IN R3 AND R4 RESPECTFULLY. A TIMER IS EMPLOYED TO PREVENT THE
;TEST FROM HANGING IF RECEIVER DONE DOES NOT RESPOND.
*****
  
```

3871 013204
 3872 013204 012700 000176
 3873 013210 012702 177560
 3874 013214 105762 000002
 3875 013220 105762 000002
 3876 013224 012703 063212
 3877 013230 012704 063202
 3878 013234 012705 000010
 3879 013240 012762 000004 000004
 3880

```

†TST176:
6S:  MOV      #176,R0      ;:LOAD R0 WITH TEST NUMBER
      MOV      #RCSR,R2   ;:R2 POINTS TO DL11 - START ADDR
      TSTB    2(R2)       ;:REFERENCE DL11 INPUT DATA BUFFER TWICE
      TSTB    2(R2)       ;:TO FLUSH RCVR "DONE" BIT
      MOV      #IBUF,R3   ;:R3 POINTS TO CORE INPUT BUFFER
      MOV      #OBUF,R4   ;:R4 POINTS TO CORE OUTPUT BUFFER
      MOV      #10,R5     ;:R5 WILL COUNT 8 CHARS OUTPUT
      MOV      #4,4(R2)   ;:TURN ON MAINT MODE
  
```

3881 013246 005001
 3882 013250 112462 000006
 3883 013254 105712
 3884 013256 100404
 3885 013260 005301
 3886 013262 001374
 3887

```

1S:  CLR      R1           ;:R1 USED AS TIMEOUT COUNTER
      MOVB    (R4)+,6(R2) ;:LOAD OUTPUT BUFFER IN DL11
2S:  TSTB    (R2)         ;:RECEIVER DONE SET ?
      BMI     3S          ;:BR IF YES
      DEC    R1           ;:COUNT THE TIMER
      BNE    2S          ;:BR IF NO TIMEOUT
  
```

3888 013264 000000
 3889 013266 000750
 3890

```

      HALT                      ;:DL11 FAILED TO RESPOND IN TIME
      BR       6S           ;:LOCK ON HARD ERROR
  
```

3891 013270 116223 000002
 3892 013274 005305
 3893 013276 001363
 3894

```

3S:  MOVB    2(R2),(R3)+   ;:READ THE DL11 INPUT BUFFER INTO CORE
      DEC    R5           ;:COUNT ONE CHAR
      BNE    1S          ;:BR IF NOT DONE 8 CHARS
  
```

3895 013300 005062 000004
 3896 013304 012705 000010
 3897 013310 012703 063212
 3898 013314 012704 063202
 3899

```

      CLR    4(R2)          ;:TURN OFF MAINT. MODE
      MOV    #10,R5        ;:RESET CHAR COUNTER
      MOV    #IBUF,R3     ;:RESET INBUF POINTER
      MOV    #OBUF,R4     ;:RESET OUTBUF POINTER
  
```

3900 013320 122324
 3901 013322 001003
 3902 013324 005305
 3903 013326 001374
 3904 013330 000410
 3905

```

4S:  CMPB    (R3)+,(R4)+   ;:INPUT = OUTPUT ??
      BNE    5S          ;:BR IF NOT
      DEC    R5          ;:COUNT ONE CHECKED
      BNE    4S          ;:BR UNTIL 8 DONE
      BR     CITST       ;:GO TO NEXT TEST
  
```

3906 013332 114303
 3907 013334 114404

```

5S:  MOVB    -(R3),R3     ;:WAS DATA IN R3 [BITS 7:0]
      MOVB    -(R4),R4     ;:S / B DATA IN R4 [BITS 7:0]
  
```

F08

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T176

MACY11 27(1006) 08-FEB-77 16:23 PAGE 73
BASIC DL11 OUT / IN ECHO TEST (MAINT MODE)

3908 013336 042703 177400
3909 013342 042704 177400
3910 013346 000000
3911 013350 000717

BIC #177400,R3
BIC #177400,R4
HALT
BR 65

;STRIP OFF BITS <15:08>
;RECEIVED DATA NOT EQUAL TO OUTPUT DATA
;LOCK ON HARD ERROR

```

3912 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3913 ;/////////////////COMPREHENSIVE INSTRUCTION TESTS/////////////////
3914 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
3915
3916 013352 012737 061204 000020 CITST: MOV    #SSCOPE, @#20    ;SET UP IOT VECTOR
3917 013360 005037 000022          CLR    @#22
3918 013364 012737 061544 000030          MOV    #SEERROR, @#30    ;SET UP EMT VECTOR
3919 013372 012737 000340 000032          MOV    #340, @#32
3920 013400 012737 063112 000034          MOV    #STRAP, @#34    ;SET UP TRAP VECTOR
3921 013406 012737 000340 000036          MOV    #340, @#36
3922 013414 012737 060610 000024          MOV    #SPWRDN, @#24    ;SET UP POWER FAIL VECTOR
3923 013422 012737 000340 000026          MOV    #340, @#26
3924 013430 105737 001141          TSTB  @#SENVN          ;DO NOT SIZE BIT SET?
3925 013434 100003          BPL   3$              ;BR IF NOT - USE HARDWARE SWITCH REG
3926 013436 012737 001142 001040          MOV    #SSWREG, @#SWR  ;USE APT SWITCH REG.
3927 013444 032777 010000 165366 3$: BIT    #SW12, @SWR    ;INHIBIT PRINTING INTRO. I.D. MESSAGE?
3928 013452 001007          BNE   1$              ;BR IF YES
3929 013454 005737 063200          TST   @#ONCE          ;FIRST TIME INTO "CIT" TESTS ?
3930 013460 001004          BNE   1$              ;BR IF NOT - PRINT ID ONLY ONCE
3931 013462 005137 063200          COM   @#ONCE          ;SET FLAG TO INHIBIT PRINTING AGAIN
3932 013466 104401          TYPE          ;IDENTIFY THIS PROGRAM
3933 013470 065065          IDENT1          ;ADDR OF THE ID MESSAGE
3934 013472 005037 177776          CLR    @#PSW          ;SET CPU PRIORITY TO LEVEL 000
3935 013476 012737 003272 001006 1$: MOV    #TST0, @#SLPADR ;INITIALIZE SCOPE LOOP RETURN
3936 013504 012737 000040 001110          MOV    #40, @#STIMES  ;ITERATE ON BIT SECTION 32 TIMES
3937 013512 010037 001124          MOV    RO, @#STESTN   ;PREVENT MISSED TEST ERROR ON
3938 ;FIRST SCOPE CALL
3939
3940 ;*****
3941 ;*TEST 177      BCC TEST WITH C=1
3942 ;*****
3943 013516          TST177:
3944 013516 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
3945 013520 012700 000177          MOV    #177, RO      ;LOAD RO WITH TEST NUMBER
3946 013524 013701 013532          MOV    @#2$, R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
3947 013530 000261          SEC              ;MAKE C=1
3948
3949 013532 103001 2$: BCC   3$          ;TEST THE BCC, IT SHOULDN'T BR
3950 013534 000401          BR    TST200        ;GO TO SCOPE EXIT
3951
3952 013536 104005 3$: ERROR 5          ;BCC FAILED
3953
3954 ;*****
3955 ;*TEST 200      BCC TEST WITH C=0
3956 ;*****
3957 013540          TST200:
3958 013540 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
3959 013542 012700 000200          MOV    #200, RO     ;LOAD RO WITH TEST NUMBER
3960 013546 013701 013554          MOV    @#2$, R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
3961 013552 000241          CLC              ;MAKE C=0
3962
3963 013554 2$: BCC   TST201        ;;TEST THE BCC-IT SHOULD BR
3964 013554 103001
3965
3966 013556 104005 3$: ERROR 5          ;BCC FAILED
3967

```

```

3968
3969
3970
3971 013560
3972 013560 000004
3973 013562 012700 000201
3974 013566 013701 013604
3975 013572 012704 000017
3976 013576 012702 177776
3977
3978 013602 000277
3979
3980 013604 103004
3981
3982 013606 013703 177776
3983 013612 020304
3984 013614 001401
3985
3986 013616 104001
3987
3988
3989
3990
3991 013620
3992 013620 000004
3993 013622 012700 000202
3994 013626 013701 013644
3995 013632 012704 000017
3996 013636 012702 177776
3997
3998 013642 000277
3999
4000 013644 000401
4001
4002 013646 104005
4003
4004 013650 013703 177776
4005 013654 020304
4006 013656 001401
4007
4008 013660 104001
4009
4010
4011
4012
4013 013662
4014 013662 000004
4015 013664 012700 000203
4016 013670 013701 013704
4017 013674 005004
4018 013676 012702 177776
4019
4020 013702 000257
4021
4022 013704 103404
4023

```

```

*****
;TEST 201 VERIFY NO BRANCH MICROROUTINE DOES NOT CLR FLAGS
*****
↑ST201:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #201,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #17,R4 ;S/B PSW
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL

SCC ;MAKE N:C = 1111

25: BCC 35 ;TEST THE BCC-IT SHOULDN'T BR

MOV @#PSW,R3 ;GET WAS FLAGS
CMP R3,R4 ;N:C = 1111?
BEQ TST202 ;BR IF YES

35: ERROR 1 ;NO BRANCH MICROROUTINE ALTERED CODES

*****
;TEST 202 VERIFY BRANCH MICROROUTINE DOES NOT CLR FLAGS
*****
↑ST202:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #202,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #17,R4 ;S/B PSW
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL

SCC ;MAKE N:C = 1111

25: BR 45 ;TEST THE BR

35: ERROR 5 ;JUST IN CASE THE BR DIDN'T WORK

45: MOV @#PSW,R3 ;GET THE FLAGS
CMP R3,R4 ;N:C = 1111?
BEQ TST203 ;BR IF YES

55: ERROR 1 ;BRANCH MICROROUTINE ALTERED CODES

*****
;TEST 203 VERIFY NO BRANCH MICROROUTINE DOES NOT SET FLAGS
*****
↑ST203:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #203,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;PSW S/B = 0
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL

CCC ;MAKE N:C = 0000

25: BCS 35 ;TEST THE BCS-IT SHOULDN'T BR

```

```

4024 013706 013703 177776      MOV      2#PSW,R3      ;GET FLAGS
4025 013712 005703              TST      R3            ;N:C = 0000
4026 013714 001401              BEQ      TST204        ;;BR IF YES
4027
4028 013716 104001      3$:      ERROR      1      ;NO BRANCH MICROROUTINE-ALTERED CODES
4029
4030
4031      ;*****
4032      ;*TEST 204      VERIFY BRANCH MICROROUTINE DOES NOT SET FLAGS
4033      ;*****
4034 013720              TST204:
4035 013720 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
4036 013722 012700 000204      MOV      #204,R0      ;:LOAD R0 WITH TEST NUMBER
4037 013726 013701 013742      MOV      2#2$,R1     ;:LOAD R1 WITH TEST INSTRUCTION WORD
4038 013732 005004              CLR      R4            ;PSW S/B = 0
4039 013734 012702 177776      MOV      #PSW,R2     ;DEST = PSW FOR ERROR CALL
4040
4041              CCC              ;MAKE N:C = 0000
4042 013742 000401      2$:      BR          4$      ;TEST THE BR
4043
4044 013744 104005      3$:      ERROR      5      ;JUST IN CASE THE BR DIDN'T WORK
4045
4046 013746 013703 177776      4$:      MOV      2#PSW,R3 ;GET FLAGS
4047 013752 005703              TST      R3            ;N:C = 0000
4048 013754 001401              BEQ      TST205        ;;BR IF YES
4049
4050 013756 104001      5$:      ERROR      1      ;BRANCH MICROROUTINE ALTERED CODES.
4051
4052      ;*****
4053      ;*TEST 205      BLE TEST WITH Z = 0, AND N,V = 00
4054      ;*****
4055 013760              TST205:
4056 013760 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
4057 013762 012700 000205      MOV      #205,R0      ;:LOAD R0 WITH TEST NUMBER
4058 013766 013701 013774      MOV      2#2$,R1     ;:LOAD R1 WITH TEST INSTRUCTION WORD
4059 013772 000257              CCC              ;CLEAR FLAGS
4060
4061 013774 003401      2$:      BLE      3$      ;TEST THE BLE-IT SHOULDN'T BR
4062 013776 000401              BR          TST206    ;;GO TO SCOPE EXIT
4063
4064 014000 104005      3$:      ERROR      5      ;BLE FAILED
4065
4066      ;*****
4067      ;*TEST 206      BLE TEST WITH Z = 1 AND N,V = 00
4068      ;*****
4069 014002              TST206:
4070 014002 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
4071 014004 012700 000206      MOV      #206,R0      ;:LOAD R0 WITH TEST NUMBER
4072 014010 013701 014020      MOV      2#2$,R1     ;:LOAD R1 WITH TEST INSTRUCTION WORD
4073 014014 000257              CCC              ;CLEAR FLAGS
4074 014016 000264              SEZ              ;SET Z = 1
4075
4076 014020      2$:      BLE      TST207    ;;TEST THE BLE-IT SHOULD BR
4077 014020 003401
4078
4079 014022 104005      3$:      ERROR      5      ;BLE FAILED

```

```

4080
4081
4082
4083
4084 014024
4085 014024 000004
4086 014026 012700 000207
4087 014032 013701 014042
4088 014036 000257
4089 014040 000262
4090
4091 014042
4092 014042 003401
4093
4094 014044 104005
4095
4096
4097
4098
4099 014046
4100 014046 000004
4101 014050 012700 000210
4102 014054 013701 014064
4103 014060 000257
4104 014062 000270
4105
4106 014064
4107 014064 003401
4108
4109 014066 104005
4110
4111
4112
4113
4114 014070
4115 014070 000004
4116 014072 012700 000211
4117 014076 013701 014106
4118 014102 000257
4119 014104 000272
4120
4121 014106 003401
4122 014110 000401
4123
4124 014112 104005
4125
4126
4127
4128
4129 014114
4130 014114 000004
4131 014116 012700 000212
4132 014122 013701 014130
4133 014126 000257
4134
4135 014130 101401

```

```

*****
;TEST 207 BLE TEST WITH Z = 0 AND N,V = 01
*****
†ST207:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #207,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
SEV ;MAKE Z = 0 AND N,V = 01

2$: BLE TST210 ;;TEST THE BLE-IT SHOULD BR

3$: ERROR 5 ;BLE FAILED

*****
;TEST 210 BLE TEST WITH Z = 0 AND N,V = 10
*****
†ST210:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #210,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
SEN ;MAKE Z = 0 AND N,V = 10

2$: BLE TST211 ;;TEST THE BLE-IT SHOULD BR

3$: ERROR 5 ;BLE FAILED

*****
;TEST 211 BLE TEST WITH Z = 0 AND N,V = 11
*****
†ST211:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #211,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
272 ;MAKE Z = 0 AND N,V = 11

2$: BLE 3$ ;TEST THE BLE-IT SHOULDN'T BR
BR TST212 ;;GO TO SCOPE EXIT

3$: ERROR 5 ;BLE FAILED

*****
;TEST 212 BLOS TEST WITH Z,C = 00
*****
†ST212:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #212,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;MAKE Z,C = 00

2$: BLOS 3$ ;TEST THE BLOS-IT SHOULDN'T BR

```

```

4136 014132 000401          BR      TST213          ;;GO TO SCOPE EXIT
4137
4138 014134 104005          3$:    ERROR    5          ;BLOS FAILED
4139
4140          ;;*****
4141          ;*TEST 213      BLOS TEST WITH Z,C = 01
4142          ;;*****
4143          †TST213:
4144 014136 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4145 014140 012700 000213    MOV      #213,R0  ;LOAD R0 WITH TEST NUMBER
4146 014144 013701 014154    MOV      @#2$,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD
4147 014150 000257          CCC          ;CLEAR FLAGS
4148 014152 000261          SEC          ;MAKE Z,C = 01
4149
4150 014154          2$:
4151 014154 101401          BLOS    TST214          ;;TEST THE BLOS-IT SHOULD BR
4152
4153 014156 104005          3$:    ERROR    5          ;BLOS FAILED
4154
4155          ;;*****
4156          ;*TEST 214      BLOS TEST WITH Z,C = 10
4157          ;;*****
4158          †TST214:
4159 014160 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4160 014162 012700 000214    MOV      #214,R0  ;LOAD R0 WITH TEST NUMBER
4161 014166 013701 014176    MOV      @#2$,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD
4162 014172 000257          CCC          ;CLEAR FLAGS
4163 014174 000264          SEZ          ;MAKE Z,C = 10
4164
4165 014176          2$:
4166 014176 101401          BLOS    TST215          ;;TEST THE BLOS-IT SHOULD BR
4167
4168 014200 104005          3$:    ERROR    5          ;BLOS FAILED
4169
4170          ;;*****
4171          ;*TEST 215      BLOS TEST WITH Z,C = 11
4172          ;;*****
4173          †TST215:
4174 014202 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4175 014204 012700 000215    MOV      #215,R0  ;LOAD R0 WITH TEST NUMBER
4176 014210 013701 014220    MOV      @#2$,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD
4177 014214 000257          CCC          ;CLEAR FLAGS
4178 014216 000265          265          ;MAKE Z,C = 11
4179
4180 014220          2$:
4181 014220 101401          BLOS    TST216          ;;TEST THE BLOS-IT SHOULD BR
4182
4183 014222 104005          3$:    ERROR    5          ;BLOS FAILED
4184
4185          ;;*****
4186          ;*TEST 216      SXT MODE 0 TEST WITH N = 0 AND C = 1
4187          ;;*****
4188          †TST216:
4189 014224 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4190 014226 012700 000216    MOV      #216,R0  ;LOAD R0 WITH TEST NUMBER
4191 014232 013701 014250    MOV      @#2$,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD

```

```

4192 014236 005004          CLR      R4          ;RESULT S / B = 0
4193 014240 012703 177777  MOV      #-1,R3     ;INITAL DEST. OP = 177777
4194 014244 000257          CCC          ;CLEAR CODES
4195 014246 000263          263         ;N:C = 0011
4196
4197 014250 006703          2$:      SXT      R3          ;TEST THE SXT
4198
4199 014252 100403          BMI      3$
4200 014254 001002          BNE      3$          ;DID SXT MAKE N:C = 0101?
4201 014256 102401          BVS      3$
4202 014260 103401          BCS      4$
4203
4204 014262 104002          3$:      ERROR    2          ;SXT FAILED TO ALTER CODES PROPERLY
4205
4206 014264 005703          4$:      TST      R3          ;DID RESULT = 0?
4207 014266 001401          BEQ      TST217     ;;BR IF IT DID
4208
4209 014270 104002          5$:      ERROR    2          ;SXT DELIVERED WRONG RESULT TO R3
4210
4211
4212
4213
4214 014272          ;*****
4215 014272 000004          ;*TEST 217      SXT MODE 0 TEST WITH N = 0 AND C = 0
4216 014274 012700 000217  ;*****
4217 014300 013701 014326  ;*TEST 217:
4218
4219 014304 032737 000020 063160 .SBTTL USER CONTROLLED BREAKPOINT -- BIT4
4220 014312 001401          BIT      #BIT4,#BPTLOC ;BREAKPOINT HALT SET ??
4221 014314 000000          BEQ      .+4         ;BR IF NOT
4222
4223 014316 005004          HALT
4224 014320 012703 177777  ;CALL THE SCOPE LOOP UTILITY
4225 014324 000257          MOV      #217,R0     ;LOAD R0 WITH TEST NUMBER
4226
4227 014326 006703          MOV      #2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
4228 014330 103001          .SBTTL USER CONTROLLED BREAKPOINT -- BIT4
4229
4230 014332 104002          BIT      #BIT4,#BPTLOC ;BREAKPOINT HALT SET ??
4231
4232
4233
4234
4235 014334          BEQ      .+4         ;BR IF NOT
4236 014334 000004          ;BREAK - DEPRESS CONTINUE TO RESTART
4237 014336 012700 000220  ;RESULT S / B = 0
4238 014342 013701 014356  ;INITIAL DEST OP = 177777
4239 014346 012704 177777  ;CLEAR N:C
4240 014352 005003          CLR      R4
4241 014354 000277          MOV      #-1,R3
4242
4243 014356 006703          2$:      SXT      R3          ;TEST THE SXT
4244
4245 014360 100003          BPL      3$
4246 014362 001402          BEQ      3$          ;N:C = 1001?
4247 014364 102401          BVS      3$

```


M08

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 80
 DQKDA.A.P11 08-FEB-77 16:17 T220 SXT MODE 0 TEST WITH N = 1 AND C = 1

```

4248 014366 103401          BCS      4$
4249
4250 014370 104002          3$:      ERROR      2          ;SXT FAILED TO ALTER CODES PROPERLY
4251
4252 014372 010305          4$:      MOV        R3,R5          ;GET RESULT
4253 014374 005105          COM      R5          ;COMPLEMENT IT-SHOULD GO TO 0
4254 014376 001401          BEQ     TST221        ;;BR IF RESULT OF SXT = 1
4255
4256 014400 104002          5$:      ERROR      2          ;SXT DELIVERED WRONG RESULT.
4257
4258
4259
4260
4261 014402
4262 014402 000004
4263 014404 012700 000221
4264 014410 013701 014426
4265 014414 012704 177777
4266 014420 005003
4267 014422 000257
4268 014424 000276
4269
4270 014426 006703          2$:      SXT        R3          ;TEST THE SXT
4271 014430 103001          BCC     TST222        ;;BR IF "C" UNAFFECTED
4272
4273 014432 104002          3$:      ERROR      2          ;SXT SET "C" BIT
4274
4275
4276
4277
4278 014434
4279 014434 000004
4280 014436 012700 000222
4281 014442 013701 014464
4282 014446 012702 063236
4283 014452 005004
4284 014454 012712 177777
4285 014460 000257
4286 014462 000263
4287
4288 014464 006712          2$:      SXT        (R2)        ;TEST THE SXT - DM1
4289
4290 014466 100403          BMI     3$
4291 014470 001002          BNE     3$          ;N:C = 0101
4292 014472 102401          BVS     3$
4293 014474 103401          BCS     4$
4294
4295 014476 104001          3$:      ERROR      1          ;SXT FAILED TO ALTER CODES PROPERLY
4296
4297 014500 005712          4$:      TST        (R2)        ;DID RESULT = 0?
4298 014502 001401          BEQ     11$         ;BR IF YES
4299
4300 014504 104001          5$:      ERROR      1          ;SXT SHOULD HAVE ZEROED (DEST)
4301
4302 014506 012702 063236          11$:     MOV        #MBUF0,R2        ;DEST ADDR = MBUF0
4303 014512 013701 014526          MOV     @#12$,R1      ;LOAD R1 WITH TEST INSTR WORD
    
```

```

;*****
;TEST 221      SXT MODE 0 TEST WITH N = 1 AND C = 0
;*****
TST221:
    
```

```

SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV     #221,R0 ;LOAD R0 WITH TEST NUMBER
MOV     @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV     #-1,R4  ;RESULT S / B = 177777
CLR     R3      ;INITIAL DEST OP = 0
CCC     276     ;CLEAR FLAGS
        ;MAKE N:C = 1110
    
```

```

2$:      SXT        R3          ;TEST THE SXT
        BCC     TST222        ;;BR IF "C" UNAFFECTED
    
```

```

3$:      ERROR      2          ;SXT SET "C" BIT
    
```

```

;*****
;TEST 222      SXT MODE 1 AND 2 TEST WITH N = 0 AND C = 1
;*****
TST222:
    
```

```

SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV     #222,R0 ;LOAD R0 WITH TEST NUMBER
MOV     @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV     #MBUF0,R2 ;R2 POINTS TO DEST OP
CLR     R4      ;RESULT S / B = 0
MOV     #-1,(R2) ;INITIAL (DEST) = 177777
CCC     263     ;CLEAR CODES
        ;MAKE N:C = 0011
    
```

```

2$:      SXT        (R2)        ;TEST THE SXT - DM1
    
```

```

BMI     3$
BNE     3$          ;N:C = 0101
BVS     3$
BCS     4$
    
```

```

3$:      ERROR      1          ;SXT FAILED TO ALTER CODES PROPERLY
    
```

```

4$:      TST        (R2)        ;DID RESULT = 0?
        BEQ     11$         ;BR IF YES
    
```

```

5$:      ERROR      1          ;SXT SHOULD HAVE ZEROED (DEST)
    
```

```

11$:     MOV        #MBUF0,R2        ;DEST ADDR = MBUF0
        MOV     @#12$,R1      ;LOAD R1 WITH TEST INSTR WORD
    
```

```

4304 014516 012712 177777      MOV      #-1,(R2)      ;INITIAL [DEST] = 177777
4305 014522 000257      CCC                      ;CLEAR CODES
4306 014524 000263      263                    ;MAKE N:C = 0011
4307
4308 014526 006722      12$:  SXT      (R2)+    ;TEST SXT - DM2
4309
4310 014530 100403      BMI      7$            ;N:C = 0101 ?
4311 014532 001002      BNE      7$
4312 014534 102401      BVS      7$
4313 014536 103401      BCS      6$
4314
4315 014540 104001      7$:   ERROR    1        ;SXT FAILED TO ALTER CODES PROPERLY
4316
4317 014542 005737 063236      6$:   TST      2#MBUFO  ;DID RESULT GET ZEROED ?
4318 014546 001401      BEQ      8$            ;BR IF YES
4319
4320 014550 104001      9$:   ERROR    1        ;SXT FAILED TO ZERO [DEST]
4321
4322 014552 020227 063240      8$:   CMP      R2,#MBUFO+2 ;WAS IT REALLY MODE 2 ?
4323 014556 001401      BEQ      TST223        ;;BR IF YES
4324
4325 014560 104001      ERROR    1            ;SXT FAILED TO AUTO INCREMENT
4326
4327
4328 ;:*****
4329 ;:TEST 223      SXT MODE 1 TEST WITH N = 0 AND C = 0
4330 ;:*****
4331 ;:TST223:
4331 014562 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
4332 014564 012700 000223      MOV      #223,R0      ;LOAD R0 WITH TEST NUMBER
4333 014570 013701 014610      MOV      2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
4334 014574 005004      CLR      R4           ;RESULT S / B = 0
4335 014576 012702 063236      MOV      #MBUFO,R2    ;R2 POINTS TO DEST OP
4336 014602 012712 177777      MOV      #-1,(R2)     ;INITIAL [DEST] = 177777
4337 014606 000257      CCC                      ;CLEAR "C" BIT
4338
4339 014610 006712      2$:   SXT      (R2)     ;TEST THE SXT
4340 014612 103001      BCC      TST224        ;;BR IF "C" UNDISTURBED
4341
4342 014614 104001      3$:   ERROR    1        ;SXT SET THE "C" BIT
4343
4344 ;:*****
4345 ;:TEST 224      SXT MODE 1 TEST WITH N = 1 AND C = 1
4346 ;:*****
4347 ;:TST224:
4348 014616 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
4349 014620 012700 000224      MOV      #224,R0      ;LOAD R0 WITH TEST NUMBER
4350 014624 013701 014644      MOV      2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
4351 014630 012704 177777      MOV      #-1,R4       ;RESULT S / B = 177777
4352 014634 012702 063236      MOV      #MBUFO,R2    ;R2 POINTS TO DEST OP
4353 014640 005012      CLR      (R2)         ;INITIAL [DEST] = 0
4354 014642 000277      SCC                      ;MAKE N:C = 1111
4355
4356 014644 006712      2$:   SXT      (R2)     ;TEST THE SXT
4357
4358 014646 100003      BPL      3$
4359 014650 001402      BEQ      3$            ;N:C = 1001?

```

4360 014652 102401
4361 014654 103401
4362
4363 014656 104001
4364
4365 014660 021204
4366 014662 001401
4367
4368 014664 104001
4369
4370
4371
4372
4373 014666
4374 014666 000004
4375 014670 012700 000225
4376 014674 013701 014716
4377 014700 012704 177777
4378 014704 012702 063236
4379 014710 005012
4380 014712 000257
4381 014714 000276
4382
4383 014716 006712
4384 014720 103001
4385
4386 014722 104001
4387
4388
4389
4390
4391 014724
4392 014724 000004
4393 014726 012700 000226
4394 014732 013701 014752
4395 014736 012704 177400
4396 014742 012703 000377
4397 014746 000257
4398 014750 000273
4399
4400 014752 000303
4401
4402 014754 100403
4403 014756 001002
4404 014760 102401
4405 014762 103001
4406
4407 014764 104002
4408
4409 014766 020403
4410 014770 001401
4411
4412 014772 104002
4413
4414
4415

BVS 3\$
BCS 4\$
3\$: ERROR 1 ;SXT FAILED TO ALTER CODES PROPERLY
4\$: CMP (R2),R4 ;RESULT = 177777?
BEQ TST225 ;;BR IF YES
5\$: ERROR 1 ;SXT DELIVERED WRONG RESULT
;*****
;*TEST 225 SXT MODE 1 TEST WITH N = 1 AND C = 0
;*****
TST225:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #225,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
MOV #MBUFD,R2 ;R2 POINTS TO DEST OP
CLR (R2) ;INITIAL [DEST] = 0
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
2\$: SXT (R2) ;TEST THE SXT
BCC TST226 ;;BR IF "C" UNAFFECTED
3\$: ERROR 1 ;SXT SET THE "C" BIT
;*****
;*TEST 226 SWAB MODE 0 TEST WITH POS. RESULT
;*****
TST226:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #226,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 177400
MOV #377,R3 ;INITIAL DEST OP = 377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2\$: SWAB R3 ;TEST THE SWAB
BMI 3\$
BNE 3\$;N:C = 0100
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SWAB FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT?
BEQ TST227 ;;BR IF YES
5\$: ERROR 2 ;SWAB DELIVERED WRONG RESULT
;*****
;*TEST 227 SWAB MODE 0 TEST WITH NEG. RESULT

```

4416
4417 014774
4418 014774 000004
4419 014776 012700 000227
4420 015002 013701 015022
4421 015006 012704 000377
4422 015012 012703 177400
4423 015016 000257
4424 015020 000267
4425
4426 015022 000303
4427
4428 015024 100003
4429 015026 001402
4430 015030 102401
4431 015032 103001
4432
4433 015034 104002
4434
4435 015036 020403
4436 015040 001401
4437
4438 015042 104002
4439
4440
4441
4442
4443 015044
4444 015044 000004
4445 015046 012700 000230
4446 015052 013701 015076
4447 015056 012704 177400
4448 015062 012702 063236
4449 015066 012712 000377
4450 015072 000257
4451 015074 000273
4452
4453 015076 000312
4454
4455 015100 100403
4456 015102 001002
4457 015104 102401
4458 015106 103001
4459
4460 015110 104001
4461
4462 015112 020412
4463 015114 001401
4464
4465 015116 104001
4466
4467 015120 013701 015140
4468 015124 012702 063236
4469 015130 012712 000377
4470 015134 000257
4471 015136 000273

```

```

*****
TST227:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #227,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #377,R4 ;RESULT S / B = 377
MOV #177400,R3 ;INITIAL DEST OP = 177400
CCC ;CLEAR FLAGS
267 ;MAKE N:C = 0111

25: SWAB R3 ;TEST THE SWAB

BPL 35
BEQ 35 ;DID SWAB MAKE N:C = 1000
BVS 35
BCC 45

35: ERROR 2 ;SWAB FAILED TO ALTER CODES PROPERLY

45: CMP R4,R3 ;DID SWAB DELIVER CORRECT RESULT?
BEQ TST230 ;BR IF OK

55: ERROR 2 ;SWAB DELIVERED WRONG RESULT

```

```

*****
TST230:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #230,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 177400
MOV #MBUFD,R2 ;R2 POINTS TO DEST OP
MOV #377,(R2) ;SET UP DEST OP = 377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011

25: SWAB (R2) ;TEST THE SWAB - DM1

BMI 35
BNE 35 ;N:C = 0100
BVS 35
BCC 45

35: ERROR 1 ;SWAB FAILED TO ALTER CODES PROPERLY

45: CMP R4,(R2) ;CORRECT RESULT?
BEQ 55 ;BR IF OK

55: ERROR 1 ;SWAB DELIVERED WRONG RESULT

55: MOV #205,R1 ;LOAD R1 WITH TEST INSTR. WORD
MOV #MBUFD,R2 ;R2 POINTS TO DEST OP
MOV #377,(R2) ;[DEST] = 000377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011

```

```

4472
4473 015140 000322      20$:  SWAB      (R2)+      ;TEST THE SWAB - DM2
4474
4475 015142 100403      BMI       7$             ;N:C = 0100
4476 015144 001002      BNE       7$
4477 015146 102401      BVS       7$
4478 015150 103001      BCC       6$
4479
4480 015152 104001      7$:  ERROR      1             ;SWAB FAILED TO SET CODES PROPERLY
4481
4482 015154 020437 063236      6$:  CMP        R4,2#MBUFD   ;CORRECT RESULT ?
4483 015160 001401      BEQ       8$             ;BR IF YES
4484
4485 015162 104001      9$:  ERROR      1             ;SWAB DELIVERED THE WRONG RESULT
4486
4487 015164 020227 063240      8$:  CMP        R2,#MBUFD+2  ;DID AUTO INCREMENT OCCUR ?
4488 015170 001401      BEQ       TST231        ;;BR IF YES
4489
4490 015172 104001      ERROR     1             ;SWAB FAILED TO AUTO INC REG.
4491
4492
4493
4494
4495
4496 015174
4497 015174 000004      TST231: SCOPE           ;CALL THE SCOPE LOOP UTILITY
4498 015176 012700 000231      MOV       #231,R0        ;LOAD R0 WITH TEST NUMBER
4499 015202 013701 015226      MOV       2#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
4500 015206 012704 000377      MOV       #377,R4       ;RESULT S / B = 377
4501 015212 012702 063236      MOV       #MBUFD,R2     ;R2 POINTS TO DEST OP
4502 015216 012712 177400      MOV       #177400,(R2)  ;SET UP DEST. OP = 177400
4503 015222 000257      CCC
4504 015224 000267      267                    ;CLEAR FLAGS
4505 015226 000312      2$:  SWAB      (R2)        ;TEST THE SWAB
4506
4507 015230 100003      BPL       3$
4508 015232 001402      BEQ       3$            ;N:C = 1000?
4509 015234 102401      BVS       3$
4510 015236 103001      BCC       4$
4511
4512 015240 104001      3$:  ERROR      1             ;SWAB FAILED TO ALTER CODES PROPERLY
4513
4514 015242 020412      4$:  CMP        R4,(R2)     ;CORRECT RESULT?
4515 015244 001401      BEQ       TST232        ;;BR IF YES
4516
4517 015246 104001      5$:  ERROR      1             ;SWAB DELIVERED WRONG RESULT
4518
4519
4520
4521
4522
4523 015250
4524 015250 000004      TST232: SCOPE           ;CALL THE SCOPE LOOP UTILITY
4525 015252 012700 000232      MOV       #232,R0        ;LOAD R0 WITH TEST NUMBER
4526 015256 013701 015272      MOV       2#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
4527 015262 005004      CLR       R4             ;RESULT S / B = 0
4527 015264 005003      CLR       R3             ;INITIAL (DEST) = 0

```

```

4528 015266 000257          CCC          ;CLEAR FLAGS
4529 015270 000273          273          ;MAKE N:C = 1011
4530
4531 015272 005403          2S:  NEG      R3          ;TEST THE NEG
4532
4533 015274 100403          BMI      3S
4534 015276 001002          BNE      3S          ;N:C = 0100 ONLY "Z" SET?
4535 015300 102401          BVS      3S
4536 015302 103001          BCC      4S
4537
4538 015304 104002          3S:  ERROR  2          ;NEG FAILED TO ALTER CODES PROPERLY
4539
4540 015306 020304          4S:  CMP      R3,R4          ;WAS RESULT = 0
4541 015310 001401          BEQ      TST233          ;;BR IF YES
4542
4543 015312 104002          5S:  ERROR  2          ;NEG DELIVERED WRONG RESULT
4544

```

```

*****
;TEST 233      NEG MODE 0 TEST : [DEST] LT 0
*****
TST233:

```

```

4548 015314
4549 015314 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4550 015316 012700 000233  MOV      #233,R0          ;LOAD R0 WITH TEST NUMBER
4551 015322 013701 015342  MOV      @#2S,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
4552 015326 012704 000002  MOV      #2,R4          ;RESULT S / B = 2
4553 015332 012703 177776  MOV      #-2,R3          ;INITIAL [DEST] = 177776
4554 015336 000257          CCC          ;CLEAR FLAGS
4555 015340 000276          276          ;MAKE N:C = 1110
4556
4557 015342 005403          2S:  NEG      R3          ;TEST THE NEG
4558
4559 015344 100403          BMI      3S
4560 015346 001402          BEQ      3S          ;N:C = 0001?
4561 015350 102401          BVS      3S
4562 015352 103401          BCS      4S
4563
4564 015354 104002          3S:  ERROR  2          ;NEG FAILED TO ALTER CODES PROPERLY
4565
4566 015356 020304          4S:  CMP      R3,R4          ;RESULT = 2?
4567 015360 001401          BEQ      TST234          ;;BR IF YES
4568
4569 015362 104002          5S:  ERROR  2          ;NEG DELIVERED WRONG RESULT
4570

```

```

*****
;TEST 234      NEG MODE 0 TEST : [DEST] = 100000 (8)
*****
TST234:

```

```

4574 015364
4575 015364 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4576 015366 012700 000234  MOV      #234,R0          ;LOAD R0 WITH TEST NUMBER
4577 015372 013701 015410  MOV      @#2S,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
4578 015376 012704 100000  MOV      #100000,R4          ;RESULT S / B = 100000
4579 015402 010403          MOV      R4,R3          ;INITIAL [DEST] = 100000
4580 015404 000257          CCC          ;CLEAR FLAGS
4581 015406 000264          SEZ          ;MAKE N:C = 01000
4582
4583 015410 005403          2S:  NEG      R3          ;TEST THE NEG

```

F09

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.A.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 86
NEG MODE 0 TEST : [DEST] = 100000 (8)

```

4584
4585 015412 100003          BPL      3$
4586 015414 001402          BEQ      3$          ;N:C = 1011?
4587 015416 102001          BVC      3$
4588 015420 103401          BCS      4$
4589
4590 015422 104002          3$:      ERROR    2          ;NEG FAILED TO ALTER CODES PROPERLY
4591
4592 015424 020304          4$:      CMP      R3,R4          ;RESULT STILL 100000?
4593 015426 001401          BEQ      TST235          ;;BR IF YES
4594
4595 015430 104002          5$:      ERROR    2          ;NEG DELIVERED WRONG RESULT
4596
4597
4598 ;:*****
4599 ;*TEST 235      NEG MODE 1 TEST : [DEST] = 0
4600 ;:*****
4601 015432          TST235:
4602 015432 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4603 015434 012700 000235          MOV      #235,R0          ;LOAD R0 WITH TEST NUMBER
4604 015440 013701 015460          MOV      2#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
4605 015444 012702 063236          MOV      #MBUFO,R2          ;R2 POINTS TO DEST OP
4606 015450 005004          CLR      R4          ;RESULT S / B = 0
4607 015452 005012          CLR      (R2)          ;INITIAL [DEST] = 0
4608 015454 000257          CCC          ;CLEAR FLAGS
4609 015456 000273          273          ;MAKE N:C = 1011
4610 015460 005412          2$:      NEG      (R2)          ;TEST THE NEG
4611
4612 015462 100403          BMI      3$
4613 015464 001002          BNE      3$          ;N:C = 0100?
4614 015466 102401          BVS      3$
4615 015470 103001          BCC      4$
4616
4617 015472 104001          3$:      ERROR    1          ;NEG FAILED TO ALTER CODES PROPERLY
4618
4619 015474 021204          4$:      CMP      (R2),R4          ;RESULT = 0?
4620 015476 001401          BEQ      TST236          ;;BR IF YES
4621
4622 015500 104001          5$:      ERROR    1          ;NEG DELIVERED WRONG RESULT
4623
4624
4625 ;:*****
4626 ;*TEST 236      NEG MODE 1 TEST : [DEST] GT 0
4627 ;:*****
4628 015502          TST236:
4629 015502 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4630 015504 012700 000236          MOV      #236,R0          ;LOAD R0 WITH TEST NUMBER
4631 015510 013701 015534          MOV      2#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
4632 015514 012702 063236          MOV      #MBUFO,R2          ;R2 POINTS TO DEST OP
4633 015520 012704 177776          MOV      #-2,R4          ;RESULT S / B = 177776
4634 015524 012712 000002          MOV      #2,(R2)          ;INITIAL [DEST] = 2
4635 015530 000257          CCC          ;CLEAR FLAGS
4636 015532 000266          266          ;MAKE N:C = 0110
4637 015534 005412          2$:      NEG      (2)          ;TEST THE NEG
4638
4639 015536 100003          BPL      3$

```

```

4640 015540 001402      BEQ      3$          ;N:C = 1001?
4641 015542 102401      BVS      3$
4642 015544 103401      BCS      4$
4643
4644 015546 104001      3$:      ERROR      1          ;NEG FAILED TO ALTER CODES PROPERLY
4645
4646 015550 021204      4$:      CMP      (R2),R4      ;CORRECT RESULT?
4647 015552 001401      BEQ      TST237      ;;BR IF YES
4648
4649 015554 104001      5$:      ERROR      1          ;NEG DELIVERED WRONG RESULT
4650

```

```

*****
;TEST 237      NEG MODE 1 TEST : [DEST] LT 0
*****
↑TST237:

```

```

4654 015556 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
4655 015556 012700 000237      MOV      #237,R0      ;;LOAD R0 WITH TEST NUMBER
4656 015560 012700 015610      MOV      2#2$,R1      ;;LOAD R1 WITH TEST INSTRUCTION WORD
4657 015564 013701 015610      MOV      #MBUFO,R2      ;R2 POINTS TO DEST OP
4658 015570 012702 063236      MOV      #2,R4        ;RESULT S / B = 2
4659 015574 012704 000002      MOV      #-2,(R2)     ;INITIAL [DEST] = 177776
4660 015600 012712 177776      CCC
4661 015604 000257      276          ;CLEAR FLAGS
4662 015606 000276          ;MAKE N:C = 1110
4663

```

```

4664 015610 005412      2$:      NEG      (R2)          ;TEST THE NEG
4665
4666 015612 100403      BMI      3$
4667 015614 001402      BEQ      3$          ;N:C = 0001?
4668 015616 102401      BVS      3$
4669 015620 103401      BCS      4$
4670

```

```

4671 015622 104001      3$:      ERROR      1          ;NEG FAILED TO ALTER CODES PROPERLY
4672
4673 015624 021204      4$:      CMP      (R2),R4      ;CORRECT RESULT = 2?
4674 015626 001401      BEQ      TST240      ;;BR IF YES
4675
4676 015630 104001      5$:      ERROR      1          ;NEG DELIVERED WRONG RESULT
4677

```

```

*****
;TEST 240      NEG MODE 1 TEST: [DEST] = 100000 (8)
*****
↑TST240:

```

```

4681 015632 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
4682 015632 012700 000240      MOV      #240,R0      ;;LOAD R0 WITH TEST NUMBER
4683 015634 012700 015662      MOV      2#2$,R1      ;;LOAD R1 WITH TEST INSTRUCTION WORD
4684 015640 013701 015662      MOV      #MBUFO,R2      ;R2 POINTS TO DEST OP
4685 015644 012702 063236      MOV      #100000,R4    ;RESULT S / B = 100000
4686 015650 012704 100000      MOV      R4,(R2)     ;INITIAL [DEST] = 100000
4687 015654 010412      CCC          ;CLEAR FLAGS
4688 015656 000257      SEZ          ;MAKE N:Z = 0100
4689 015660 000264
4690

```

```

4691 015662 005412      2$:      NEG      (R2)          ;TEST THE NEG
4692
4693 015664 100003      BPL      3$
4694 015666 001402      BEQ      3$          ;N:C = 1011?
4695 015670 102001      BVC      3$

```



```

4696 015672 103401          BCS      4S
4697
4698 015674 104001          3S:     ERROR    1          ;NEG FAILED TO ALTER CODES PROPERLY
4699
4700 015676 021204          4S:     CMP      (R2),R4      ;CORRECT RESULT = 100000?
4701 015700 001401          BEQ      TST241             ;;BR IF YES
4702
4703 015702 104001          5S:     ERROR    1          ;NEG DELIVERED WRONG RESULT
4704
4705
4706
4707
4708 015704
4709 015704 000004          ;*****
4710 015706 012700 000241          ;*TEST 241      ROR TEST - DMO - N:C = 1110
4711 015712 013701 015732          ;*****
4712 015716 012704 052525          ;TST241:
4713 015722 012703 125252          SCOPE
4714 015726 000257          MOV      #241,R0           ;CALL THE SCOPE LOOP UTILITY
4715 015730 000276          MOV      @#25,R1          ;LOAD R0 WITH TEST NUMBER
4716
4717 015732 006003          2S:     ROR      R3          ;LOAD R1 WITH TEST INSTRUCTION WORD
4718
4719 015734 100403          BMI      3S               ;RESULT S / B = 52525
4720 015736 001402          BEQ      3S               ;(DEST) = 125252
4721 015740 102401          BVS     3S               ;CLEAR FLAGS
4722 015742 103001          BCC     4S               ;N:C = 1111
4723
4724 015744 104002          3S:     ERROR    2          ;TEST THE ROR
4725
4726 015746 020403          BMI      3S               ;N:C = 0000 ?
4727 015750 001401          BEQ      3S
4728
4729 015752 104002          5S:     ERROR    2          ;ROR FAILED TO ALTER CODES PROPERLY
4730
4731
4732
4733
4734 015754
4735 015754 000004          ;*****
4736 015756 012700 000242          ;*TEST 242      ROR TEST - DMO - N:C = 1000
4737 015762 013701 016000          ;*****
4738 015766 005004          ;TST242:
4739 015770 012703 000001          SCOPE
4740 015774 000257          MOV      #242,R0           ;CALL THE SCOPE LOOP UTILITY
4741 015776 000270          MOV      @#25,R1          ;LOAD R0 WITH TEST NUMBER
4742
4743 016000 006003          2S:     ROR      R3          ;LOAD R1 WITH TEST INSTRUCTION WORD
4744
4745 016002 100403          BMI      3S               ;RESULT S / B = 000000
4746 016004 001002          BNE     3S               ;(DEST) = 1
4747 016006 102001          BVC     3S               ;CLEAR FLAGS
4748 016010 103401          BCS     4S               ;N:C = 1000
4749
4750 016012 104002          3S:     ERROR    2          ;TEST THE ROR
4751

```

```

4752 016014 020403      4S:   CMP      R4,R3      ;CORRECT RESULT ?
4753 016016 001401      BEQ      TST243      ;;BR IF YES
4754
4755 016020 104002      5S:   ERROR    2          ;ROR DELIVERED THE WRONG RESULT
4756
4757
4758
4759
4760 016022
4761 016022 000004      ;*****
4762 016024 012700 000243      ;*TEST 243      ROR TEST - DMO - N:C = 0111
4763 016030 013701 016050      ;*****
4764 016034 012704 125252      ;TST243:
4765 016040 012703 052525      SCOPE
4766 016044 000257      MOV      #243,R0      ;CALL THE SCOPE LOOP UTILITY
4767 016046 000267      MOV      @#25,R1      ;LOAD R0 WITH TEST NUMBER
4768
4769 016050 006003      2S:   ROR      R3          ;TEST THE ROR
4770
4771 016052 100003      BPL      3S            ;N:C = 1001 ?
4772 016054 001402      BEQ      3S
4773 016056 102401      BVS      3S
4774 016060 103401      BCS      4S
4775
4776 016062 104002      3S:   ERROR    2          ;ROR FAILED TO ALTER CODES PROPERLY
4777
4778 016064 020403      4S:   CMP      R4,R3      ;CORRECT RESULT ?
4779 016066 001401      BEQ      TST244      ;;BR IF YES
4780
4781 016070 104002      5S:   ERROR    2          ;ROR DELIVERED THE WRONG RESULT
4782
4783
4784
4785
4786 016072
4787 016072 000004      ;*****
4788 016074 012700 000244      ;*TEST 244      ASR TEST - DMO - N:C = 1000
4789 016100 013701 016116      ;*****
4790 016104 005004      ;TST244:
4791 016106 012703 000001      SCOPE
4792 016112 000257      MOV      #244,R0      ;CALL THE SCOPE LOOP UTILITY
4793 016114 000270      MOV      @#25,R1      ;LOAD R0 WITH TEST NUMBER
4794
4795 016116 006003      2S:   ROR      R3          ;TEST THE ROR
4796
4797 016120 100403      BMI      3S            ;N:C = 0111 ?
4798 016122 001002      BNE      3S
4799 016124 102001      BVC      3S
4800 016126 103401      BCS      4S
4801
4802 016130 104002      3S:   ERROR    2          ;ROR FAILED TO ALTER CODES PROPERLY
4803
4804 016132 020403      4S:   CMP      R4,R3      ;CORRECT RESULT ?
4805 016134 001401      BEQ      TST245      ;;BR IF YES
4806
4807 016136 104002      5S:   ERROR    2          ;ROR DELIVERED THE WRONG RESULT

```

4808
4809
4810
4811
4812 016140
4813 016140 000004
4814 016142 012700 000245
4815 016146 013701 016166
4816 016152 012704 152525
4817 016156 012703 125252
4818 016162 000257
4819 016164 000265
4820
4821 016166 006003
4822
4823 016170 100003
4824 016172 001402
4825 016174 102001
4826 016176 103001
4827
4828 016200 104002
4829
4830 016202 020403
4831 016204 001401
4832
4833 016206 104002
4834
4835
4836
4837
4838 016210
4839 016210 000004
4840 016212 012700 000246
4841 016216 013701 016236
4842 016222 012704 025252
4843 016226 012703 052525
4844 016232 000257
4845 016234 000274
4846
4847 016236 006003
4848
4849 016240 100403
4850 016242 001402
4851 016244 102001
4852 016246 103401
4853
4854 016250 104002
4855
4856 016252 020403
4857 016254 001401
4858
4859 016256 104002
4860
4861
4862
4863

```
*****  
; *TEST 245 ASR TEST - DMO - N:C = 0101  
*****  
TST245:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #245,R0 ;LOAD R0 WITH TEST NUMBER  
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #152525,R4 ;RESULT S / B = 152525  
MOV #125252,R3 ;[DEST] = 125252  
CCC ;CLEAR FLAGS  
265 ;N:C = 0101  
  
2$: ROR R3 ;TEST THE ROR  
  
BPL 3$ ;N:C = 1010 ?  
BEQ 3$  
BVC 3$  
BCC 4$  
  
3$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY  
  
4$: CMP R4,R3 ;CORRECT RESULT ?  
BEQ TST246 ;;BR IF YES  
  
5$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT  
  
*****  
; *TEST 246 ASR TEST - DMO - N:C = 1100  
*****  
TST246:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #246,R0 ;LOAD R0 WITH TEST NUMBER  
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #25252,R4 ;RESULT S / B = 25252  
MOV #52525,R3 ;[DEST] = 52525  
CCC ;CLEAR FLAGS  
274 ;N:C = 1100  
  
2$: ROR R3 ;TEST THE ROR  
  
BMI 3$ ;N:C = 0011 ?  
BEQ 3$  
BVC 3$  
BCS 4$  
  
3$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY  
  
4$: CMP R4,R3 ;CORRECT RESULT ?  
BEQ TST247 ;;BR IF YES  
  
5$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT  
  
*****  
; *TEST 247 ROR TEST - DMI - N:C = 1110  
*****
```

4864 016260
4865 016260 000004
4866 016262 012700 000247
4867 016266 013701 016312
4868 016272 012702 063236
4869 016276 012704 052525
4870 016302 012712 125252
4871 016306 000257
4872 016310 000276
4873
4874 016312 006012
4875
4876 016314 100403
4877 016316 001402
4878 016320 102401
4879 016322 103001
4880
4881 016324 104001
4882
4883 016326 020412
4884 016330 001402
4885 016332 011203
4886 016334 104001
4887
4888
4889
4890
4891 016336
4892 016336 000004
4893 016340 012700 000250
4894 016344 013701 016366
4895 016350 012702 063236
4896 016354 005004
4897 016356 012712 000001
4898 016362 000257
4899 016364 000270
4900
4901 016366 006012
4902
4903 016370 100403
4904 016372 001002
4905 016374 102001
4906 016376 103401
4907
4908 016400 104001
4909
4910 016402 020412
4911 016404 001402
4912
4913 016406 011203
4914 016410 104001
4915
4916
4917
4918
4919 016412

TST247:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #247,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;:DEST ADDR = MBUF0
MOV #52525,R4 ;:RESULT S / B = 52525
MOV #125252,(R2) ;:(DEST) = 125252
CCC ;CLEAR FLAGS
276 ;N:C = 1110
25: ROR (R2) ;TEST THE ROR
BMI 35 ;N:C = 0000 ?
BEQ 35
BVS 35
BCC 45
35: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY
45: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST250 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;ROR DELIVERED WRONG RESULT
;:*****
;:TEST 250 ROR TEST - DM1 - N:C = 1000
;:*****
TST250:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #250,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;:DEST ADDR = MBUF0
CLR R4 ;:RESULT S / B = 000000
MOV #1,(R2) ;:(DEST) = 1
CCC ;CLEAR FLAGS
SEN ;N:C = 1000
25: ROR (R2) ;TEST THE ROR
BMI 35 ;N:C = 0111 ?
BNE 35
BVC 35
BCS 45
35: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY
45: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST251 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;ROR DELIVERED WRONG RESULT
;:*****
;:TEST 251 ROR TEST - DM1 - N:C = 0111
;:*****
TST251:

```

4920 016412 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4921 016414 012700 000251  MOV      #251,R0      ;LOAD R0 WITH TEST NUMBER
4922 016420 013701 016444  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
4923 016424 012702 063236  MOV      #MBUF0,R2    ;DEST ADDR = MBUF0
4924 016430 012704 125252  MOV      #125252,R4   ;RESULT S / B = 125252
4925 016434 012712 052525  MOV      #52525,(R2)  ;[DEST] = 52525
4926 016440 000257          CCC              ;CLEAR FLAGS
4927 016442 000267          267             ;N:C = 0111
4928
4929 016444 006012          25:  ROR      (R2)      ;TEST THE ROR
4930
4931 016446 100003          BPL      3$          ;N:C = 1001 ?
4932 016450 001402          BEQ      3$
4933 016452 102401          BVS      3$
4934 016454 103401          BCS      4$
4935
4936 016456 104001          3$:  ERROR    1        ;ROR FAILED TO ALTER CODES PROPERLY
4937
4938 016460 020412          4$:  CMP      R4,(R2)  ;CORRECT RESULT ?
4939 016462 001402          BEQ      TST252      ;;BR IF YES
4940
4941 016464 011203          5$:  MOV      (R2),R3  ;GET THE WAS DATA
4942 016466 104001          ERROR    1        ;ROR DELIVERED WRONG RESULT
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975

```

;TEST 252 ASR TEST - DM1 - N:C = 1000

TST252:

```

SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV      #252,R0      ;LOAD R0 WITH TEST NUMBER
MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #MBUF0,R2    ;DEST ADDR = MBUF0
CLR      R4           ;RESULT S / B = 000000
MOV      #1,(R2)     ;[DEST] = 1
CCC              ;CLEAR FLAGS
SEN
25:  ROR      (R2)      ;TEST THE ROR
BMI      3$          ;N:C = 0111 ?
BNE      3$
BVC      3$
BCS      4$
3$:  ERROR    1        ;ROR FAILED TO ALTER CODES PROPERLY
4$:  CMP      R4,(R2)  ;CORRECT RESULT ?
BEQ      TST253      ;;BR IF YES
5$:  MOV      (R2),R3  ;GET THE WAS DATA
ERROR    1        ;ROR DELIVERED WRONG RESULT

```

;TEST 253 ASR TEST - DM1 - N:C = 1100

TST253:

```

4976 016544 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
4977 016546 012700 000253  MOV      #253,R0      ;:LOAD R0 WITH TEST NUMBER
4978 016552 013701 016576  MOV      @#25,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
4979 016556 012702 063236  MOV      #MBUF0,R2    ;:DEST ADDR = MBUF0
4980 016562 012704 025252  MOV      #25252,R4    ;:RESULT S / B = 25252
4981 016566 012712 052525  MOV      #52525,(R2)  ;:[DEST] = 52525
4982 016572 000257          CCC              ;:CLEAR FLAGS
4983 016574 000274          274              ;:N:C = 1100
4984
4985 016576 006012          25:  ROR      (R2)      ;:TEST THE ROR
4986
4987 016600 100403          BMI      3$          ;:N:C = 0011 ?
4988 016602 001402          BEQ      3$
4989 016604 102001          BVC      3$
4990 016606 103401          BCS      4$
4991
4992 016610 104001          3$:  ERROR    1        ;:ROR FAILED TO ALTER CODES PROPERLY
4993
4994 016612 020412          4$:  CMP      R4,(R2)   ;:CORRECT RESULT ?
4995 016614 001402          BEQ      TST254      ;;BR IF YES
4996
4997 016616 011203          5$:  MOV      (R2),R3   ;:GET THE WAS DATA
4998 016620 104001          ERROR    1          ;:ROR DELIVERED WRONG RESULT
4999
5000          ;:*****
5001          ;:TEST 254      ASR TEST - DM1 - N:C = 0101
5002          ;:*****
5003          †TST254:
5004 016622 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5005 016624 012700 000254  MOV      #254,R0      ;:LOAD R0 WITH TEST NUMBER
5006 016630 013701 016654  MOV      @#25,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
5007 016634 012702 063236  MOV      #MBUF0,R2    ;:DEST ADDR = MBUF0
5008 016640 012704 152525  MOV      #152525,R4   ;:RESULT S / B = 152525
5009 016644 012712 125252  MOV      #125252,(R2) ;:[DEST] = 125252
5010 016650 000257          CCC              ;:CLEAR FLAGS
5011 016652 000265          265              ;:N:C = 0101
5012
5013 016654 006012          2$:  ROR      (R2)      ;:TEST THE ROR
5014
5015 016656 100003          BPL      3$          ;:N:C = 1010 ?
5016 016660 001402          BEQ      3$
5017 016662 102001          BVC      3$
5018 016664 103001          BCC      4$
5019
5020 016666 104001          3$:  ERROR    1        ;:ROR FAILED TO ALTER CODES PROPERLY
5021
5022 016670 020412          4$:  CMP      R4,(R2)   ;:CORRECT RESULT ?
5023 016672 001402          BEQ      TST255      ;;BR IF YES
5024
5025 016674 011203          5$:  MOV      (R2),R3   ;:GET THE WAS DATA
5026 016676 104001          ERROR    1          ;:ROR DELIVERED WRONG RESULT
5027
5028          ;:*****
5029          ;:TEST 255      RORB TEST - DM2 - EVEN ADDRESS
5030          ;:*****
5031 016700          †TST255:

```

```

5032 016700 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5033 016702 012700 000255  MOV      #255,R0  ;;LOAD R0 WITH TEST NUMBER
5034 016706 013701 016732  MOV      @#25,R1  ;;LOAD R1 WITH TEST INSTRUCTION WORD
5035 016712 012702 063236  MOV      #MBUF0,R2 ;DEST ADDR = MBUF0
5036 016716 012704 000177  MOV      #177,R4  ;RESULT S / B = 177
5037 016722 010203          MOV      R2,R3   ;R3 CONTAINS DEST ADDR
5038 016724 012712 000377  MOV      #377,(R2);[DEST] = 377
5039 016730 000257          CCC           ;SCOPE SYNC "C" = 0
5040
5041 016732 106023          2$: RORB   (R3)+ ;TEST THE RORB
5042
5043 016734 103401          BCS      4$     ;BR IF ROR SET "C"
5044
5045 016736 104001          3$: ERROR  1     ;ROR FAILED TO SET "C"
5046
5047 016740 022703 063237  4$: CMP    #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?
5048 016744 001401          BEQ      6$     ;;BR IF YES
5049
5050 016746 104005          5$: ERROR  5     ;RORB FAILED TO UPDATE DEST REG
5051
5052 016750 020412          6$: CMP    R4,(R2) ;CORRECT RESULT ?
5053 016752 001402          BEQ      TST256 ;;BR IF YES
5054
5055 016754 011203          7$: MOV    (R2),R3 ;GET THE WAS DATA
5056 016756 104001          ERROR  1     ;RORB DELIVERED WRONG RESULT
5057

```

```

*****
; *TEST 256 RORB TEST - DM1 - EVEN ADDRESS
*****

```

```

5058
5059
5060
5061 016760
5062 016760 000004          TST256: SCOPE          ;CALL THE SCOPE LOOP UTILITY
5063 016762 012700 000256  MOV      #256,R0  ;;LOAD R0 WITH TEST NUMBER
5064 016766 013701 017014  MOV      @#25,R1  ;;LOAD R1 WITH TEST INSTRUCTION WORD
5065 016772 012702 063236  MOV      #MBUF0,R2 ;DEST ADDR = MBUF0
5066 016776 012704 000377  MOV      #377,R4  ;RESULT S / B = 377
5067 017002 010203          MOV      R2,R3   ;R3 CONTAINS DEST ADDR
5068 017004 012712 000376  MOV      #376,(R2);[DEST] = 376
5069 017010 000257          CCC           ;CLEAR FLAGS
5070 017012 000261          SEC           ;SCOPE SYNC - SET "C"
5071
5072 017014 106013          2$: RORB   (R3)  ;TEST THE RORB
5073
5074 017016 103001          BCC      4$     ;BR IF "C" CLR - IT SHOULD BE
5075
5076 017020 104001          3$: ERROR  1     ;RORB FAILED TO CLR "C"
5077
5078 017022 020412          4$: CMP    R4,(R2) ;CORRECT RESULT ?
5079 017024 001402          BEQ      TST257 ;;BR IF YES
5080
5081 017026 011203          5$: MOV    (R2),R3 ;GET THE WAS DATA
5082 017030 104001          ERROR  1     ;RORB DELIVERED WRONG RESULT
5083

```

```

*****
; *TEST 257 RORB TEST - DM2 - ODD ADDRESS
*****

```

```

5084
5085
5086
5087 017032
TST257:

```

B10

```

5088 017032 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5089 017034 012700 000257  MOV      #257,R0      ;LOAD R0 WITH TEST NUMBER
5090 017040 013701 017102  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
5091                                     .SBTTL USER CONTROLLED BREAKPOINT -- BITS
5092 017044 032737 000040 063160 BIT      #BITS,@#BPTLOC ;BREAKPOINT HALT SET ??
5093 017052 001401          BEQ      .+4          ;BR IF NOT
5094 017054 000000          HALT                    ;BREAK - DEPRESS CONTINUE TO RESTART
5095 017056 012702 063237  MOV      #MBUFD+1,R2  ;DEST ADDR = MBUFD+1
5096 017062 012704 077777  MOV      #77777,R4    ;RESULT S / B = 77777
5097 017066 012705 063236  MOV      #MBUFD,R5    ;POINT R5 TO CHECK RESULT
5098 017072 010203          MOV      R2,R3        ;R3 CONTAINS DEST ADDR
5099 017074 012715 177777  MOV      #-1,(R5)    ;[DEST] = 177777
5100 017100 000257          CCC                    ;SCOPE SYNC - "C" =0
5101
5102 017102 106023          2$:  RORB   (R3)+      ;TEST THE RORB
5103
5104 017104 103401          BCS     4$          ;BR IF "C" IS SET - IT SHOULD BE
5105
5106 017106 104001          3$:  ERROR 1          ;RORB FAILED TO SET "C"
5107
5108 017110 022703 063240  4$:  CMP    #MBUFD+2,R3 ;DID DEST REG GET INCREMENTED ?
5109 017114 001401          BEQ    6$          ;BR IF YES
5110
5111 017116 104005          5$:  ERROR 5          ;RORB FAILED TO UPDATE DEST REG
5112
5113 017120 020415          6$:  CMP    R4,(R5)   ;CORRECT RESULT ?
5114 017122 001402          BEQ    TST260      ;BR IF YES
5115
5116 017124 011503          MOV    (R5),R3     ;GET THE WAS DATA
5117 017126 104001          7$:  ERROR 1          ;RORB DELIVERED WRONG RESULT
5118
5119                                     ;*****
5120                                     ;*TEST 260 RORB TEST - DM1 - ODD ADDRESS
5121                                     ;*****
5122                                     †TST260:
5123 017130 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5124 017132 012700 000260  MOV      #260,R0      ;LOAD R0 WITH TEST NUMBER
5125 017136 013701 017166  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
5126 017142 012702 063237  MOV      #MBUFD+1,R2  ;DEST ADDR = MBUFD+1
5127 017146 012704 177777  MOV      #-1,R4       ;RESULT S / B = 177777
5128 017152 012705 063236  MOV      #MBUFD,R5    ;POINT R5 TO CHECK RESULT
5129 017156 010203          MOV      R2,R3        ;R3 CONTAINS DEST ADDR
5130 017160 012715 177377  MOV      #177377,(R5) ;[DEST] = 177377
5131 017164 000261          SEC                    ;SCOPE SYNC - SET "C"
5132
5133 017166 106023          2$:  RORB   (R3)+      ;TEST THE RORB
5134
5135 017170 103001          BCC    4$          ;BR IF "C" CLEAR - IT SHOULD BE
5136
5137 017172 104001          3$:  ERROR 1          ;RORB FAILED TO CLEAR "C"
5138
5139 017174 020415          4$:  CMP    R4,(R5)   ;CORRECT RESULT ?
5140 017176 001402          BEQ    TST261      ;BR IF YES
5141
5142 017200 011503          MOV    (R5),R3     ;GET THE WAS DATA
5143 017202 104001          5$:  ERROR 1          ;RORB DELIVERED WRONG RESULT
  
```



```

5144
5145
5146
5147
5148 017204
5149 017204 000004
5150 017206 012700 000261
5151 017212 013701 017242
5152 017216 012702 063237
5153 017222 012704 000377
5154 017226 012705 063236
5155 017232 010203
5156 017234 012715 000777
5157 017240 000257
5158
5159 017242 106223
5160
5161 017244 103401
5162
5163 017246 104001
5164
5165 017250 022703 063240
5166 017254 001401
5167
5168 017256 104005
5169
5170 017260 020415
5171 017262 001402
5172
5173 017264 011503
5174 017266 104001
5175
5176
5177
5178
5179 017270
5180 017270 000004
5181 017272 012700 000262
5182 017276 013701 017326
5183 017302 012702 063237
5184 017306 012704 140377
5185 017312 012705 063236
5186 017316 010203
5187 017320 012715 100377
5188 017324 000261
5189
5190 017326 106213
5191
5192 017330 103001
5193
5194 017332 104001
5195
5196 017334 020415
5197 017336 001402
5198
5199 017340 011503

```

```

*****
;TEST 261 ASRB TEST - DM2 - ODD ADDRESS
*****
TST261:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #261,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #377,R4 ;RESULT S / B = 377
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #777,(R5) ;[DEST] = 777
CCC ;SCOPE SYNC "C" = 0

25: ASRB (R3)+ ;TEST THE ASRB

BCS 45 ;BR IF CARRY SET - IT SHOULD BE

35: ERROR 1 ;ASRB FAILED TO SET THE CARRY

45: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 65 ;BR IF YES

55: ERROR 5 ;ASRB FAILED TO UPDATE DEST REG

65: CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST262 ;BR IF YES

75: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;ASRB DELIVERED WRONG RESULT

*****
;TEST 262 ASRB TEST - DM1 - ODD ADDRESS
*****
TST262:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #262,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #140377,R4 ;RESULT S / B = 140377
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #100377,(R5) ;[DEST] = 100377
SEC ;SCOPE SYNC - "C" = 1

25: ASRB (R3) ;TEST THE ASRB

BCC 45 ;BR IF CARRY CLEAR - IT SHOULD BE

35: ERROR 1 ;ASRB FAILED TO CLEAR THE CARRY

45: CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST263 ;BR IF YES

MOV (R5),R3 ;GET THE WAS DATA

```

```

5200 017342 104001 5S: ERROR 1 ;ASRB DELIVERED WRONG RESULT
5201
5202
5203 ;*****
5204 ;#TEST 263 ASRB TEST - DM2 - EVEN ADDRESS
5205 ;*****
5206 017344 000004 TST263: SCOPE ;CALL THE SCOPE LOOP UTILITY
5207 017346 012700 000263 MOV #263,R0 ;LOAD R0 WITH TEST NUMBER
5208 017352 013701 017376 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
5209 017356 012702 063236 MOV #MBUFD,R2 ;DEST ADDR = MBUFD
5210 017362 012704 000077 MOV #77,R4 ;RESULT S / B = 77
5211 017366 010203 MOV R2,R3 ;R3 CONTAINS DEST ADDR
5212 017370 012712 000177 MOV #177,(R2) ;[DEST] = 177
5213 017374 000257 CCC ;SCOPE SYNC - "C" = 0
5214
5215 017376 106223 2S: ASRB (R3)+ ;TEST THE ASRB
5216
5217 017400 103401 BCS 4S ;BR IF "C" = 1 - IT SHOULD BE
5218
5219 017402 104001 3S: ERROR 1 ;ASRB FAILED TO SET "C"
5220
5221 017404 022703 063237 4S: CMP #MBUFD+1,R3 ;DID DEST REG GET INCREMENTED ?
5222 017410 001401 BEQ 6S ;BR IF YES
5223
5224 017412 104005 5S: ERROR 5 ;ASRB FAILED TO UPDATE DEST REG
5225
5226 017414 020412 6S: CMP R4,(R2) ;CORRECT RESULT ?
5227 017416 001402 BEQ TST264 ;BR IF YES
5228
5229 017420 011203 7S: MOV (R2),R3 ;GET THE WAS DATA
5230 017422 104001 ERROR 1 ;ASRB DELIVERED WRONG RESULT
5231
5232 ;*****
5233 ;#TEST 264 ASRB TEST - DM1 - EVEN ADDRESS
5234 ;*****
5235 017424 000004 TST264: SCOPE ;CALL THE SCOPE LOOP UTILITY
5236 017426 012700 000264 MOV #264,R0 ;LOAD R0 WITH TEST NUMBER
5237 017426 013701 017456 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
5238 017432 012702 063236 MOV #MBUFD,R2 ;DEST ADDR = MBUFD
5239 017436 012704 000303 MOV #303,R4 ;RESULT S / B = 303
5240 017442 010203 MOV R2,R3 ;R3 CONTAINS DEST ADDR
5241 017446 012712 000206 MOV #206,(R2) ;[DEST] = 206
5242 017450 000261 SEC ;SCOPE SYNC - "C" = 1
5243 017454
5244
5245 017456 106213 2S: ASRB (R3) ;TEST THE CLASRB
5246
5247 017460 103001 BCC 4S ;BR IF CARRY CLEAR - ITSHOULD BE
5248
5249 017462 104001 3S: ERROR 1 ;ASRB FAILED TO CLEAR THE CARRY
5250
5251 017464 020412 4S: CMP R4,(R2) ;CORRECT RESULT ?
5252 017466 001402 BEQ TST265 ;BR IF YES
5253
5254 017470 011203 7S: MOV (R2),R3 ;GET THE WAS DATA
5255 017472 104001 ERROR 1 ;ASRB DELIVERED WRONG RESULT

```

5256
5257
5258
5259
5260 017474
5261 017474 000004
5262 017476 012700 000265
5263 017502 013701 017516
5264 017506 005004
5265 017510 005003
5266 017512 000257
5267 017514 000273
5268
5269 017516 005703
5270
5271 017520 100403
5272 017522 001002
5273 017524 102401
5274 017526 103001
5275
5276 017530 104002
5277
5278 017532 020403
5279 017534 001401
5280
5281 017536 104002
5282
5283
5284
5285
5286 017540
5287 017540 000004
5288 017542 012700 000266
5289 017546 013701 017564
5290 017552 005004
5291 017554 005104
5292 017556 010403
5293 017560 000257
5294 017562 000264
5295
5296 017564 005703
5297
5298 017566 100003
5299 017570 001402
5300 017572 102401
5301 017574 103001
5302
5303 017576 104002
5304
5305 017600 020403
5306 017602 001401
5307
5308 017604 104002
5309
5310
5311

```
*****  
*TEST 265 TST DMO TEST - N:C = 1011  
*****  
TST265:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #265,R0 ;LOAD R0 WITH TEST NUMBER  
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
CLR R4 ;RESULT S / B = 000000  
CLR R3 ;[DEST] = 000000  
CCC ;CLEAR CODES  
273 ;N:C=1011  
  
25: TST R3 ;TEST THE TST  
  
BMI 35 ;N:C = 0100 ?  
BNE 35  
BVS 35  
BCC 45  
  
35: ERROR 2 ;TST FAILED TO ALTER CODES PROPERLY  
  
45: CMP R4,R3 ;RESULT OK ?  
BEQ TST266 ;;BR IF YES  
  
55: ERROR 2 ;TST ALTERED THE [DEST]  
  
*****  
*TEST 266 TST DMO TEST - N:C = 0100  
*****  
TST266:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #266,R0 ;LOAD R0 WITH TEST NUMBER  
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
CLR R4 ;RESULT S / B = 177777  
COM R4 ;[DEST] = 177777  
MOV R4,R3 ;CLEAR CODES  
CCC ;N:C=0100  
264  
  
25: TST R3 ;TEST THE TST  
  
BPL 35 ;N:C = 1000 ?  
BEQ 35  
BVS 35  
BCC 45  
  
35: ERROR 2 ;TST FAILED TO ALTER CODES PROPERLY  
  
45: CMP R4,R3 ;RESULT OK ?  
BEQ TST267 ;;BR IF YES  
  
55: ERROR 2 ;TST ALTERED THE [DEST]  
  
*****  
*TEST 267 CLR DMO TEST - N:C = 1011  
*****
```


G10

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T271

MACY11 27(1006) 08-FEB-77 16:23 PAGE 100
 COM DMO TEST - N:C = 0110

5368	017732	012704	125252	MOV	#125252,R4	;RESULT S / B = 125252
5369	017736	012703	052525	MOV	#52525,R3	;[DEST] = 52525
5370	017742	000257		CCC		;CLEAR CODES
5371	017744	000266		266		;N:C = 0110
5372						
5373	017746	005103		2\$: COM	R3	;TEST THE COM
5374						
5375	017750	100003		BPL	3\$;N:C = 1001 ?
5376	017752	001402		BEQ	3\$	
5377	017754	102401		BVS	3\$	
5378	017756	103401		BCS	4\$	
5379						
5380	017760	104002		3\$: ERROR	2	;COM FAILED TO ALTER THE CODES PROPERLY
5381						
5382	017762	020403		4\$: CMP	R4,R3	;RESULT OK ?
5383	017764	001401		BEQ	TST272	;BR IF YES
5384						
5385	017766	104002		5\$: ERROR	2	;COM DELIVERED THE WRONG RESULT
5386						
5387						
5388						
5389						
5390	017770					
5391	017770	000004				
5392	017772	012700	000272	SCOPE		;CALL THE SCOPE LOOP UTILITY
5393	017776	013701	020014	MOV	#272,R0	;LOAD R0 WITH TEST NUMBER
5394	020002	005004		MOV	#2\$,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
5395	020004	012703	177777	CLR	R4	;RESULT S / B = 000000
5396	020010	000257		MOV	#-1,R3	;[DEST] = 177777
5397	020012	000271		CCC		;CLEAR CODES
5398				271		;N:C = 1001
5399	020014	005103		2\$: COM	R3	;TEST THE COM
5400						
5401	020016	100403		BMI	3\$;N:C = 0101 ?
5402	020020	001002		BNE	3\$	
5403	020022	102401		BVS	3\$	
5404	020024	103401		BCS	4\$	
5405						
5406	020026	104002		3\$: ERROR	2	;COM FAILED TO ALTER THE CODES PROPERLY
5407						
5408	020030	020403		4\$: CMP	R4,R3	;RESULT OK ?
5409	020032	001401		BEQ	TST273	;BR IF YES
5410						
5411	020034	104002		5\$: ERROR	2	;COM DELIVERED THE WRONG RESULT
5412						
5413						
5414						
5415						
5416	020036					
5417	020036	000004				
5418	020040	012700	000273	SCOPE		;CALL THE SCOPE LOOP UTILITY
5419	020044	013701	020062	MOV	#273,R0	;LOAD R0 WITH TEST NUMBER
5420	020050	005004		MOV	#2\$,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
5421	020052	012703	177777	CLR	R4	;RESULT S / B = 000000
5422	020056	000257		MOV	#-1,R3	;[DEST] = 177777
5423	020060	000273		CCC		;CLEAR CODES
				273		;N:C = 1011

H10

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T273

MACY11 27(1006) 08-FEB-77 16:23 PAGE 101
INC DMO TEST - N:C = 1011

```
5424
5425 020062 005203      2$:   INC      R3           ;TEST THE INC
5426
5427 020064 100403           BMI      3$           ;N:C = 0101 ?
5428 020066 001002           BNE      3$
5429 020070 102401           BVS      3$
5430 020072 103401           BCS      4$
5431
5432 020074 104002      3$:   ERROR    2           ;INC FAILED TO ALTER THE CODES PROPERLY
5433
5434 020076 020403      4$:   CMP      R4,R3       ;RESULT OK ?
5435 020100 001401           BEQ      T$T274       ;;BR IF YES
5436
5437 020102 104002      5$:   ERROR    2           ;INC DELIVERED THE WRONG RESULT
5438
5439
5440
5441
5442 020104
5443 020104 000004      ;*****
5444 020106 012700 000274      ;*TEST 274      INC DMO TEST - N:C = 0100
5445 020112 013701 020132      ;*****
5446 020116 012704 100000      ;*TEST 274:
5447 020122 012703 077777      SCOPE
5448 020126 000257           MOV      #274,R0       ;CALL THE SCOPE LOOP UTILITY
5449 020130 000264           MOV      @#2$,R1       ;LOAD R0 WITH TEST NUMBER
5450
5451 020132 005203      2$:   INC      R3           ;TEST THE INC
5452
5453 020134 100003           BPL      3$           ;N:C = 1010 ?
5454 020136 001402           BEQ      3$
5455 020140 102001           BVC      3$
5456 020142 103001           BCC      4$
5457
5458 020144 104002      3$:   ERROR    2           ;INC FAILED TO ALTER THE CODES PROPERLY
5459
5460 020146 020403      4$:   CMP      R4,R3       ;RESULT OK ?
5461 020150 001401           BEQ      T$T275       ;;BR IF YES
5462
5463 020152 104002      5$:   ERROR    2           ;INC DELIVERED THE WRONG RESULT
5464
5465
5466
5467
5468 020154
5469 020154 000004      ;*****
5470 020156 012700 000275      ;*TEST 275      DEC DMO TEST - N:C = 1011
5471 020162 013701 020200      ;*****
5472 020166 005004           ;*TEST 275:
5473 020170 012703 000001      SCOPE
5474 020174 000257           MOV      #275,R0       ;CALL THE SCOPE LOOP UTILITY
5475 020176 000273           MOV      @#2$,R1       ;LOAD R0 WITH TEST NUMBER
5476
5477 020200 005303      2$:   DEC      R3           ;TEST THE DEC
5478
5479 020202 100403           CLR      R4           ;LOAD R1 WITH TEST INSTRUCTION WORD
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865
5866
5867
5868
5869
5870
5871
5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907
5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950
5951
5952
5953
5954
5955
5956
5957
5958
5959
5960
5961
5962
5963
5964
5965
5966
5967
5968
5969
5970
5971
5972
5973
5974
5975
5976
5977
5978
5979
5980
5981
5982
5983
5984
5985
5986
5987
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
5998
5999
6000
```

```

5480 020204 001002          BNE      3$
5481 020206 102401          BVS      3$
5482 020210 103401          BCS      4$
5483
5484 020212 104002          3$:      ERROR    2          ;DEC FAILED TO ALTER THE CODES PROPERLY
5485
5486 020214 020403          4$:      CMP      R4,R3          ;RESULT OK ?
5487 020216 001401          BEQ      T$T276          ;;BR IF YES
5488
5489 020220 104002          5$:      ERROR    2          ;DEC DELIVERED THE WRONG RESULT
5490
5491
5492
5493
5494 020222
5495 020222 000004
5496 020224 012700 000276          SCOPE
5497 020230 013701 020250          MOV      #276,R0          ;CALL THE SCOPE LOOP UTILITY
5498 020234 012704 077777          MOV      @#25,R1          ;LOAD R0 WITH TEST NUMBER
5499 020240 012703 100000          MOV      #77777,R4          ;LOAD R1 WITH TEST INSTRUCTION WORD
5500 020244 000257          MOV      #100000,R3          ;RESULT S / B = 77777
5501 020246 000274          CCC          ;[DEST] = 100000
5502          274          ;CLEAR CODES
5503 020250 005303          2$:      DEC      R3          ;N:C = 1100
5504
5505 020252 100403          BMI      3$          ;TEST THE DEC
5506 020254 001402          BEQ      3$          ;N:C = 0010 ?
5507 020256 102001          BVC      3$
5508 020260 103001          BCC      4$
5509
5510 020262 104002          3$:      ERROR    2          ;DEC FAILED TO ALTER THE CODES PROPERLY
5511
5512 020264 020403          4$:      CMP      R4,R3          ;RESULT OK ?
5513 020266 001401          BEQ      T$T277          ;;BR IF YES
5514
5515 020270 104002          5$:      ERROR    2          ;DEC DELIVERED THE WRONG RESULT
5516
5517
5518
5519
5520 020272
5521 020272 000004
5522 020274 012700 000277          SCOPE
5523 020300 013701 020314          MOV      #277,R0          ;CALL THE SCOPE LOOP UTILITY
5524 020304 012704 177777          MOV      @#25,R1          ;LOAD R0 WITH TEST NUMBER
5525 020310 005003          MOV      #-1,R4          ;LOAD R1 WITH TEST INSTRUCTION WORD
5526 020312 000257          CLR      R3          ;RESULT S / B = 177777
5527          CCC          ;[DEST] = 000000
5528 020314 005303          2$:      DEC      R3          ;CLEAR CODES
5529
5530 020316 100003          BPL      3$          ;TEST THE DEC
5531 020320 001402          BEQ      3$          ;N:C = 1000 ?
5532 020322 102401          BVS      3$
5533 020324 103001          BCC      4$
5534
5535 020326 104002          3$:      ERROR    2          ;DEC FAILED TO ALTER THE CODES PROPERLY

```

J10

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 103
DEC DMO TEST - N:C = 0000

```

5536
5537 020330 020403      4$:  CMP      R4,R3      ;RESULT OK ?
5538 020332 001401      BEQ      TST300      ;;BR IF YES
5539
5540 020334 104002      5$:  ERROR    2          ;DEC DELIVERED THE WRONG RESULT
5541
5542      ;*****
5543      ;*TEST 300      ASL DMO TEST - N:C = 1000
5544      ;*****
5545      †TST300:
5546 020336 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
5547 020340 012700 000300  MOV      #300,R0      ;:LOAD R0 WITH TEST NUMBER
5548 020344 013701 020362  MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
5549 020350 005004      CLR      R4          ;:RESULT S / B = 000000
5550 020352 012703 100000  MOV      #100000,R3   ;:(DEST) = 100000
5551 020356 000257      CCC          ;:CLEAR CODES
5552 020360 000270      SEN          ;:N:C = 1000
5553
5554 020362 006303      2$:  ASL      R3          ;TEST THE ASL
5555
5556 020364 100403      BMI      3$          ;N:C = 0111 ?
5557 020366 001002      BNE      3$
5558 020370 102001      BVC      3$
5559 020372 103401      BCS      4$
5560
5561 020374 104002      3$:  ERROR    2          ;ASL FAILED TO ALTER THE CODES PROPERLY
5562
5563 020376 020403      4$:  CMP      R4,R3      ;RESULT OK ?
5564 020400 001401      BEQ      TST301      ;;BR IF YES
5565
5566 020402 104002      5$:  ERROR    2          ;ASL DELIVERED THE WRONG RESULT
5567
5568      ;*****
5569      ;*TEST 301      ASL DMO TEST - N:C = 0101
5570      ;*****
5571      †TST301:
5572 020404 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
5573 020406 012700 000301  MOV      #301,R0      ;:LOAD R0 WITH TEST NUMBER
5574 020412 013701 020432  MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
5575 020416 012704 100000  MOV      #100000,R4   ;:RESULT S / B = 100000
5576 020422 012703 040000  MOV      #40000,R3   ;:(DEST) = 40000
5577 020426 000257      CCC          ;:CLEAR CODES
5578 020430 000265      265        ;:N:C = 0101
5579
5580 020432 006303      2$:  ASL      R3          ;TEST THE ASL
5581
5582 020434 100003      BPL      3$          ;N:C = 1010 ?
5583 020436 001402      BEQ      3$
5584 020440 102001      BVC      3$
5585 020442 103001      BCC      4$
5586
5587 020444 104002      3$:  ERROR    2          ;ASL FAILED TO ALTER THE CODES PROPERLY
5588
5589 020446 020403      4$:  CMP      R4,R3      ;RESULT OK ?
5590 020450 001401      BEQ      TST302      ;;BR IF YES
5591

```


K10

5592 020452 104002
5593
5594
5595
5596
5597 020454
5598 020454 000004
5599 020456 012700 000302
5600 020462 013701 020476
5601 020466 005004
5602 020470 005003
5603 020472 000257
5604 020474 000262
5605
5606 020476 006303
5607
5608 020500 100403
5609 020502 001002
5610 020504 102401
5611 020506 103001
5612
5613 020510 104002
5614
5615 020512 020403
5616 020514 001401
5617
5618 020516 104002
5619
5620
5621
5622
5623 020520
5624 020520 000004
5625 020522 012700 000303
5626 020526 013701 020546
5627 020532 012704 052525
5628 020536 012703 125252
5629 020542 000257
5630 020544 000275
5631
5632 020546 006103
5633
5634 020550 100403
5635 020552 001402
5636 020554 102001
5637 020556 103401
5638
5639 020560 104002
5640
5641 020562 020403
5642 020564 001401
5643
5644 020566 104002
5645
5646
5647

55: ERROR 2 ;ASL DELIVERED THE WRONG RESULT
;*****
;#TEST 302 ASL DMO TEST - N:C = 0010
;*****
†TST302:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #302,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R3 ;[DEST] = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010
25: ASL R3 ;TEST THE ASL
;N:C = 0100 ?
BMI 35
BNE 35
BVS 35
BCC 45
35: ERROR 2 ;ASL FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST303 ;;BR IF YES
55: ERROR 2 ;ASL DELIVERED THE WRONG RESULT
;*****
;#TEST 303 ROL DMO TEST - N:C = 1101
;*****
†TST303:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #303,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #52525,R4 ;RESULT S / B = 52525
MOV #125252,R3 ;[DEST] = 125252
CCC ;CLEAR CODES
275 ;N:C = 1101
25: ROL R3 ;TEST THE ROL
;N:C = 0011 ?
BMI 35
BEQ 35
BVC 35
BCS 45
35: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST304 ;;BR IF YES
55: ERROR 2 ;ROL DELIVERED THE WRONG RESULT
;*****
;#TEST 304 ROL DMO TEST - N:C = 0101

```

5648
5649 020570
5650 020570 000004
5651 020572 012700 000304
5652 020576 013701 020616
5653 020602 012704 125253
5654 020606 012703 052525
5655 020612 000257
5656 020614 000265
5657
5658 020616 006103
5659
5660 020620 100003
5661 020622 001402
5662 020624 102001
5663 020626 103001
5664
5665 020630 104002
5666 020632 020403
5667 020634 001401
5668
5669 020636 104002
5670
5671
5672
5673
5674 020640
5675 020640 000004
5676 020642 012700 000305
5677 020646 013701 020662
5678 020652 005004
5679 020654 005003
5680 020656 000257
5681 020660 000262
5682
5683 020662 006103
5684
5685 020664 100403
5686 020666 001002
5687 020670 102401
5688 020672 103001
5689
5690 020674 104002
5691
5692 020676 020403
5693 020700 001401
5694
5695 020702 104002
5696
5697
5698
5699
5700 020704
5701 020704 000004
5702 020706 012700 000306
5703 020712 013701 020732

```

```

*****
TST304:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #304,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125253,R4 ;RESULT S / B = 125253
MOV #52525,R3 ;[DEST] = 52525
CCC ;CLEAR CODES
265 ;N:C = 0101

25: ROL R3 ;TEST THE ROL

BPL 35 ;N:C = 1010 ?
BEQ 35
BVC 35
BCC 45

35: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST305 ;;BR IF YES

55: ERROR 2 ;ROL DELIVERED THE WRONG RESULT
*****
*TEST 305 ROL DMO TEST - N:C = 0010
*****
TST305:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #305,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R3 ;[DEST] = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010

25: ROL R3 ;TEST THE ROL

BMI 35 ;N:C = 0100 ?
BNE 35
BVS 35
BCC 45

35: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST306 ;;BR IF YES

55: ERROR 2 ;ROL DELIVERED THE WRONG RESULT
*****
*TEST 306 ADC DMO TEST - N:C = 0101
*****
TST306:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #306,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

```

M10

MAINDEC-11-DGKDA-A KD11-K BASIC LOGIC TESTS
 DGKDA.P11 08-FEB-77 16:17 T306

MACY11 27(1006) 08-FEB-77 16:23 PAGE 106
 ADC DMO TEST - N:C = 0101

```

5704 020716 012704 100000      MOV      #100000,R4      ;RESULT S / B = 100000
5705 020722 012703 077777      MOV      #77777,R3      ;[DEST] = 77777
5706 020726 000257              CCC                      ;CLEAR CODES
5707 020730 000265              265                      ;N:C = 0101
5708
5709 020732 005503      2$:      ADC      R3      ;TEST THE ADC
5710
5711 020734 100003              BPL      3$              ;N:C = 1010 ?
5712 020736 001402              BEQ      3$
5713 020740 102001              BVC      3$
5714 020742 103001              BCC      4$
5715
5716 020744 104002      3$:      ERROR      2      ;ADC FAILED TO ALTER THE CODES PROPERLY
5717
5718 020746 020403      4$:      CMP      R4,R3      ;RESULT OK ?
5719 020750 001401              BEQ      TST307          ;;BR IF YES
5720
5721 020752 104002      5$:      ERROR      2      ;ADC DELIVERED THE WRONG RESULT
5722
5723      ;*****
5724      ;*TEST 307      ADC DMO TEST - N:C = 1011
5725      ;*****
5726      †TST307:
5727 020754 000004              SCOPE                    ;CALL THE SCOPE LOOP UTILITY
5728 020756 012700 000307      MOV      #307,R0          ;LOAD R0 WITH TEST NUMBER
5729 020762 013701 021000      MOV      @#2$,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
5730 020766 005004              CLR      R4              ;RESULT S / B = 000000
5731 020770 012703 177777      MOV      #-1,R3          ;[DEST] = 177777
5732 020774 000257              CCC                      ;CLEAR CODES
5733 020776 000273              273                      ;N:C = 1011
5734
5735 021000 005503      2$:      ADC      R3      ;TEST THE ADC
5736
5737 021002 100403              BMI      3$              ;N:C = 0101 ?
5738 021004 001002              BNE      3$
5739 021006 102401              BVS      3$
5740 021010 103401              BCS      4$
5741
5742 021012 104002      3$:      ERROR      2      ;ADC FAILED TO ALTER THE CODES PROPERLY
5743
5744 021014 020403      4$:      CMP      R4,R3      ;RESULT OK ?
5745 021016 001401              BEQ      TST310          ;;BR IF YES
5746
5747 021020 104002      5$:      ERROR      2      ;ADC DELIVERED THE WRONG RESULT
5748
5749      ;*****
5750      ;*TEST 310      ADC DMO TEST - N:C = 1010
5751      ;*****
5752      †TST310:
5753 021022 000004              SCOPE                    ;CALL THE SCOPE LOOP UTILITY
5754 021024 012700 000310      MOV      #310,R0          ;LOAD R0 WITH TEST NUMBER
5755 021030 013701 021050      MOV      @#2$,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
5756 021034 012704 177777      MOV      #-1,R4          ;RESULT S / B = 177777
5757 021040 012703 177777      MOV      #-1,R3          ;[DEST] = 177777
5758 021044 000257              CCC                      ;CLEAR CODES
5759 021046 000272              272                      ;N:C = 1010
  
```

N10

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T310

MACY11 27(1006) 08-FEB-77 16:23 PAGE 107
ADC DMO TEST - N:C = 1010

```

5760
5761 021050 005503      2$:  ADC      R3          ;TEST THE ADC
5762
5763 021052 100003          BPL      3$          ;N:C = 1000 ?
5764 021054 001402          BEQ      3$
5765 021056 102401          BVS      3$
5766 021060 103001          BCC      4$
5767
5768 021062 104002      3$:  ERROR    2          ;ADC FAILED TO ALTER THE CODES PROPERLY
5769
5770 021064 020403      4$:  CMP      R4,R3        ;RESULT OK ?
5771 021066 001401          BEQ      TST311      ;;BR IF YES
5772
5773 021070 104002      5$:  ERROR    2          ;ADC DELIVERED THE WRONG RESULT
5774

```

```

*****
; *TEST 311      SBC DMO TEST - N:C = 1011
*****
TST311:

```

```

5778 021072
5779 021072 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5780 021074 012700 000311  MOV      #311,R0      ;;LOAD R0 WITH TEST NUMBER
5781 021100 013701 021116  MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
5782 021104 005004          CLR      R4          ;RESULT S / B = 000000
5783 021106 012703 000001  MOV      #1,R3        ;[DEST] = +1
5784 021112 000257          CCC          ;CLEAR CODES
5785 021114 000273          273          ;N:C = 1011
5786

```

```

5787 021116 005603      2$:  SBC      R3          ;TEST THE SBC
5788
5789 021120 100403          BMI      3$          ;N:C = 0100 ?
5790 021122 001002          BNE      3$
5791 021124 102401          BVS      3$
5792 021126 103001          BCC      4$
5793
5794 021130 104002      3$:  ERROR    2          ;SBC FAILED TO ALTER THE CODES PROPERLY
5795
5796 021132 020403      4$:  CMP      R4,R3        ;RESULT OK ?
5797 021134 001401          BEQ      TST312      ;;BR IF YES
5798
5799 021136 104002      5$:  ERROR    2          ;SBC DELIVERED THE WRONG RESULT
5800

```

```

*****
; *TEST 312      SBC DMO TEST - N:C = 0101
*****
TST312:

```

```

5804 021140
5805 021140 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5806 021142 012700 000312  MOV      #312,R0      ;;LOAD R0 WITH TEST NUMBER
5807 021146 013701 021166  MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
5808 021152 012704 077777  MOV      #077777,R4   ;RESULT S / B = 077777
5809 021156 012703 100000  MOV      #100000,R3   ;[DEST] = 100000
5810 021162 000257          CCC          ;CLEAR CODES
5811 021164 000265          265          ;N:C = 0101
5812

```

```

5813 021166 005603      2$:  SBC      R3          ;TEST THE SBC
5814
5815 021170 100403          BMI      3$          ;N:C = 0010 ?

```

5816 021172 001402
5817 021174 102001
5818 021176 103001
5819
5820 021200 104002
5821
5822 021202 020403
5823 021204 001401
5824
5825 021206 104002
5826
5827
5828
5829
5830 021210
5831 021210 000004
5832 021212 012700 000313
5833 021216 013701 021236
5834 021222 012704 000001
5835 021226 012703 000001
5836 021232 000257
5837 021234 000276
5838
5839 021236 005603
5840
5841 021240 100403
5842 021242 001402
5843 021244 102401
5844 021246 103001
5845
5846 021250 104002
5847
5848 021252 020403
5849 021254 001401
5850
5851 021256 104002
5852
5853
5854
5855
5856 021260
5857 021260 000004
5858 021262 012700 000314
5859 021266 013701 021304
5860 021272 012704 177777
5861 021276 005003
5862 021300 000257
5863 021302 000267
5864
5865 021304 005603
5866
5867 021306 100003
5868 021310 001402
5869 021312 102401
5870 021314 103401
5871

BEQ 3\$
BVC 3\$
BCC 4\$
3\$: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,R3 ;RESULT OK ?
BEQ TST313 ;;BR IF YES
5\$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT
;*****
;*TEST 313 SBC DMO TEST - N:C = 1110
;*****
TST313:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #313,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 1
MOV #1,R3 ;[DEST] = 1
CCC ;CLEAR CODES
276 ;N:C = 1110
2\$: SBC R3 ;TEST THE SBC
;N:C = 0000 ?
3\$: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,R3 ;RESULT OK ?
BEQ TST314 ;;BR IF YES
5\$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT
;*****
;*TEST 314 SBC DMO TEST - N:C = 0111
;*****
TST314:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #314,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
CLR R3 ;[DEST] = 000000
CCC ;CLEAR CODES
267 ;N:C = 0111
2\$: SBC R3 ;TEST THE SBC
;N:C = 1001 ?
BPL 3\$
BEQ 3\$
BVS 3\$
BCS 4\$

```

5872 021316 104002 35: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY
5873
5874 021320 020403 45: CMP R4,R3 ;RESULT OK ?
5875 021322 001401 BEQ TST315 ;;BR IF YES
5876
5877 021324 104002 55: ERROR 2 ;SBC DELIVERED THE WRONG RESULT
5878

```

```

*****
;TEST 315 TST DMI TEST - N:C = 1011
*****
TST315:

```

```

5882 021326 000004 000315 SCOPE ;CALL THE SCOPE LOOP UTILITY
5883 021326 012700 021354 MOV #315,R0 ;LOAD R0 WITH TEST NUMBER
5884 021330 012700 063236 MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
5885 021334 013701 063236 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
5886 021340 012702 063236 CLR R4 ;RESULT S / B = 000000
5887 021344 005004 063236 CLR (R2) ;[DEST] = 000000
5888 021346 005012 063236 CCC ;CLEAR CODES
5889 021350 000257 063236 CCC ;N:C=1011
5890 021352 000273 063236 273
5891

```

```

5892 021354 005712 25: TST (R2) ;TEST THE TST
5893
5894 021356 100403 BMI 35 ;N:C = 0100 ?
5895 021360 001002 BNE 35
5896 021362 102401 BVS 35
5897 021364 103001 BCC 45
5898

```

```

5899 021366 104001 35: ERROR 1 ;TST FAILED TO ALTER CODES PROPERLY
5900
5901 021370 020412 45: CMP R4,(R2) ;RESULT OK ?
5902 021372 001402 BEQ TST316 ;;BR IF YES
5903
5904 021374 011203 55: MOV (R2),R3 ;GET THE WAS DATA
5905 021376 104001 55: ERROR 1 ;TST ALTERED THE [DEST]
5906

```

```

*****
;TEST 316 TST DMI TEST - N:C = 0100
*****
TST316:

```

```

5910 021400 000004 000316 SCOPE ;CALL THE SCOPE LOOP UTILITY
5911 021400 012700 021432 MOV #316,R0 ;LOAD R0 WITH TEST NUMBER
5912 021402 012700 063236 MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
5913 021406 013701 063236 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
5914 021412 012702 063236 CLR R4 ;RESULT S / B = 177777
5915 021416 005004 063236 COM R4 ;[DEST] = 177777
5916 021420 005104 177777 MOV #-1,(R2) ;CLEAR CODES
5917 021422 012712 177777 CCC ;N:C=0100
5918 021426 000257 177777 CCC
5919 021430 000264 177777 264
5920

```

```

5921 021432 005712 25: TST (R2) ;TEST THE TST
5922
5923 021434 100003 BPL 35 ;N:C = 1000 ?
5924 021436 001402 BEQ 35
5925 021440 102401 BVS 35
5926 021442 103001 BCC 45
5927

```

D11

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 110
 DQKDA.P11 08-FEB-77 16:17 T316 TST DM1 TEST - N:C = 0100

5928	021444	104001				3S:	ERROR 1	;TST FAILED TO ALTER CODES PROPERLY
5929								
5930	021446	020412				4S:	CMP R4,(R2)	;RESULT OK ?
5931	021450	001402					BEQ TST317	;BR IF YES
5932								
5933	021452	011203				5S:	MOV (R2),R3	;GET THE WAS DATA
5934	021454	104001					ERROR 1	;TST ALTERED THE [DEST]
5935								
5936								
5937								
5938								
5939	021456							
5940	021456	000004						
5941	021460	012700	000317					
5942	021464	013701	021520					
5943								
5944	021470	032737	000100	063160				
5945	021476	001401						
5946	021500	000000						
5947	021502	012702	063236					
5948	021506	005004						
5949	021510	012712	177777					
5950	021514	000257						
5951	021516	000273						
5952								
5953	021520	005012				2S:	CLR (R2)	;TEST THE CLR
5954								
5955	021522	100403						
5956	021524	001002						
5957	021526	102401						
5958	021530	103001						
5959								
5960	021532	104001				3S:	ERROR 1	;CLR FAILED TO ALTER THE CODES PROPERLY
5961								
5962	021534	020412				4S:	CMP R4,(R2)	;RESULT OK ?
5963	021536	001402					BEQ TST320	;BR IF YES
5964								
5965	021540	011203				5S:	MOV (R2),R3	;GET THE WAS DATA
5966	021542	104001					ERROR 1	;CLR DELIVERED THE WRONG RESULT
5967								
5968								
5969								
5970								
5971	021544							
5972	021544	000004						
5973	021546	012700	000320					
5974	021552	013701	021572					
5975	021556	012702	063236					
5976	021562	005004						
5977	021564	013712	063250					
5978	021570	000257						
5979								
5980	021572	005022				2S:	CLR (R2)+	;TEST THE CLR
5981								
5982	021574	100403						
5983	021576	001002						

5984 021600 102401
5985 021602 103001
5986
5987 021604 104001
5988
5989 021606 022702 063240
5990 021612 001401
5991
5992 021614 104005
5993
5994 021616 020442
5995 021620 001402
5996
5997 021622 011203
5998 021624 104001
5999

BVS 3\$
BCC 4\$
3\$: ERROR 1 ;CLR FAILED TO ALTER THE CODES PROPERLY
4\$: CMP #MBUFO+2,R2 ;DID CLR INCREMENT DEST REG
BEQ 6\$;BR IF YES
5\$: ERROR 5 ;CLR FAILED TO UPDATE DEST REG
6\$: CMP R4 -(R2) ;RESULT OK ?
BEQ TST321 ;;BR IF YES
7\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CLR DELIVERED THE WRONG RESULT

;TEST 321 COM DM1 TEST - N:C = 0110

TST321:

6003 021626
6004 021626 000004
6005 021630 012700 000321
6006 021634 013701 021660
6007 021640 012702 063236
6008 021644 012704 125252
6009 021650 012712 052525
6010 021654 000257
6011 021656 000266
6012

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #321,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #125252,R4 ;RESULT S / B = 125252
MOV #52525,(R2) ;[DEST] = 52525
CCC ;CLEAR CODES
266 ;N:C = 0110

6013 021660 005112
6014
6015 021662 100003
6016 021664 001402
6017 021666 102401
6018 021670 103401
6019
6020 021672 104001
6021 021674 020412
6022 021676 001402
6023

2\$: COM (R2) ;TEST THE CLR
BPL 3\$;N:C = 1001 ?
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;COM FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST322 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;COM DELIVERED THE WRONG RESULT

6024 021700 011203
6025 021702 104001
6026
6027

;TEST 322 COM DM1 TEST - N:C = 1001

TST322:

6030 021704
6031 021704 000004
6032 021706 012700 000322
6033 021712 013701 021734
6034 021716 012702 063236
6035 021722 005004
6036 021724 012712 177777
6037 021730 000257
6038 021732 000271
6039

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #322,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
MOV #-1,(R2) ;[DEST] = 177777
CCC ;CLEAR CODES
271 ;N:C = 1001

F11

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.A.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 112
 COM DM1 TEST - N:C = 1001

```

6040 021734 005112      25:  COM      (R2)          ;TEST THE COM
6041
6042 021736 100403      BMI      35          ;N:C = 0101 ?
6043 021740 001002      BNE      35
6044 021742 102401      BVS      35
6045 021744 103401      BCS      45
6046
6047 021746 104001      35:  ERROR    1          ;COM FAILED TO ALTER THE CODES PROPERLY
6048 021750 020412      45:  CMP      R4,(R2)    ;RESULT OK ?
6049 021752 001402      BEQ      TST323      ;;BR IF YES
6050
6051 021754 011203      55:  MOV      (R2),R3    ;GET THE WAS DATA
6052 021756 104001      ERROR    1          ;COM DELIVERED THE WRONG RESULT
6053
6054
6055
6056
6057 021760
6058 021760 000004
6059 021762 012700 000323
6060 021766 013701 022010
6061 021772 012702 063236
6062 021776 005004
6063 022000 012712 177777
6064 022004 000257
6065 022006 000273
6066
6067 022010 005212      25:  INC      (R2)          ;TEST THE INC
6068
6069 022012 100403      BMI      35          ;N:C = 0101 ?
6070 022014 001002      BNE      35
6071 022016 102401      BVS      35
6072 022020 103401      BCS      45
6073
6074 022022 104001      35:  ERROR    1          ;INC FAILED TO ALTER THE CODES PROPERLY
6075 022024 020412      45:  CMP      R4,(R2)    ;RESULT OK ?
6076 022026 001402      BEQ      TST324      ;;BR IF YES
6077
6078 022030 011203      55:  MOV      (R2),R3    ;GET THE WAS DATA
6079 022032 104001      ERROR    1          ;INC DELIVERED THE WRONG RESULT
6080
6081
6082
6083
6084 022034
6085 022034 000004
6086 022036 012700 000324
6087 022042 013701 022066
6088 022046 012702 063236
6089 022052 012704 100000
6090 022056 012712 077777
6091 022062 000257
6092 022064 000264
6093
6094 022066 005212      25:  INC      (R2)          ;TEST THE INC
6095
  
```

```

;*****
;TEST 323      INC DM1 TEST - N:C = 1011
;*****
TST323:
  
```

```

SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV      #323,R0 ;LOAD R0 WITH TEST NUMBER
MOV      #25,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #MBUF0,R2 ;DEST ADDR = MBUF0
CLR      R4      ;RESULT S / B = 000000
MOV      #-1,(R2) ;[DEST] = 177777
CCC
273           ;CLEAR CODES
           ;N:C = 1011
  
```

```

;*****
;TEST 324      INC DM1 TEST - N:C = 0100
;*****
TST324:
  
```

```

SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV      #324,R0 ;LOAD R0 WITH TEST NUMBER
MOV      #25,R1  ;LOAD R1 WITH TEST INSTRUCTION WOFO
MOV      #MBUF0,R2 ;DEST ADDR = MBUF0
MOV      #100000,R4 ;RESULT S / B = 100000
MOV      #77777,(R2) ;[DEST] = 77777
CCC
264           ;CLEAR CODES
           ;N:C = 0100
  
```

G11

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T324

MACY11 27(1006) 08-FEB-77 16:23 PAGE 113
 INC DM1 TEST - N:C = 0100

6096	022070	100003		BPL	3S	;N:C = 1010 ?
6097	022072	001402		BEQ	3S	
6098	022074	102001		BVC	3S	
6099	022076	103001		BCC	4S	
6100						
6101	022100	104001		3S: ERROR	1	;INC FAILED TO ALTER THE CODES PROPERLY
6102	022102	020412		4S: CMP	R4,(R2)	;RESULT OK ?
6103	022104	001402		BEQ	TST325	;BR IF YES
6104						
6105	022106	011203		5S: MOV	(R2),R3	;GET THE WAS DATA
6106	022110	104001		ERROR	1	;INC DELIVERED THE WRONG RESULT
6107						
6108						
6109						
6110						
6111	022112					
6112	022112	000004				
6113	022114	012700	000325	SCOPE		;CALL THE SCOPE LOOP UTILITY
6114	022120	013701	022142	MOV	#325,R0	;LOAD R0 WITH TEST NUMBER
6115	022124	012702	063236	MOV	#25,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
6116	022130	005004		MOV	#MBUFO,R2	;DEST ADDR = MBUFO
6117	022132	012712	000001	CLR	R4	;RESULT S / B = 000000
6118	022136	000257		MOV	#1,(R2)	;[DEST] = 1
6119	022140	000273		CCC		;CLEAR CODES
6120				273		;N:C = 1011
6121	022142	005312		2S: DEC	(R2)	;TEST THE DEC
6122						
6123	022144	100403		BMI	3S	;N:C = 0101 ?
6124	022146	001002		BNE	3S	
6125	022150	102401		BVS	3S	
6126	022152	103401		BCS	4S	
6127						
6128	022154	104001		3S: ERROR	1	;DEC FAILED TO ALTER THE CODES PROPERLY
6129	022156	020412		4S: CMP	R4,(R2)	;RESULT OK ?
6130	022160	001402		BEQ	TST326	;BR IF YES
6131						
6132	022162	011203		5S: MOV	(R2),R3	;GET THE WAS DATA
6133	022164	104001		ERROR	1	;DEC DELIVERED THE WRONG RESULT
6134						
6135						
6136						
6137						
6138	022166					
6139	022166	000004				
6140	022170	012700	000326	SCOPE		;CALL THE SCOPE LOOP UTILITY
6141	022174	013701	022220	MOV	#326,R0	;LOAD R0 WITH TEST NUMBER
6142	022200	012702	063236	MOV	#25,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
6143	022204	012704	077777	MOV	#MBUFO,R2	;DEST ADDR = MBUFO
6144	022210	012712	100000	MOV	#77777,R4	;RESULT S / B = 77777
6145	022214	000257		MOV	#100000,(R2)	;[DEST] = 100000
6146	022216	000274		CCC		;CLEAR CODES
6147				274		;N:C = 1100
6148	022220	005312		2S: DEC	(R2)	;TEST THE DEC
6149						
6150	022222	100403		BMI	3S	;N:C = 0010 ?
6151	022224	001402		BEQ	3S	

H11

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T326

MACY11 27(1006) 08-FEB-77 16:23 PAGE 114
DEC DM1 TEST - N:C = 1100

```
6152 022226 102001          BVC      3$
6153 022230 103001          BCC      4$
6154
6155 022232 104001          3$:      ERROR      1          ;DEC FAILED TO ALTER THE CODES PROPERLY
6156 022234 020412          4$:      CMP        R4,(R2)      ;RESULT OK ?
6157 022236 001402          BEQ      TST327          ;;BR IF YES
6158
6159 022240 011203          MOV      (R2),R3        ;GET THE WAS DATA
6160 022242 104001          5$:      ERROR      1          ;DEC DELIVERED THE WRONG RESULT
6161
6162
6163
6164
6165 022244
6166 022244 000004          ;*****
6167 022246 012700 000327          ;*TEST 327      DEC DM1 TEST - N:C = 0000
6168 022252 013701 022272          ;*****
6169 022256 012702 063236          ;*TEST 327:
6170 022262 012704 177777          SCOPE
6171 022266 005012          MOV      #327,R0        ;CALL THE SCOPE LOOP UTILITY
6172 022270 000257          MOV      @#2$,R1        ;LOAD R0 WITH TEST NUMBER
6173
6174 022272 005312          2$:      DEC      (R2)        ;TEST THE DEC
6175
6176 022274 100003          BPL      3$            ;N:C = 1000 ?
6177 022276 001402          BEQ      3$
6178 022300 102401          BVS      3$
6179 022302 103001          BCC      4$
6180
6181 022304 104001          3$:      ERROR      1          ;DEC FAILED TO ALTER THE CODES PROPERLY
6182 022306 020412          4$:      CMP      R4,(R2)      ;RESULT OK ?
6183 022310 001402          BEQ      TST330        ;;BR IF YES
6184
6185 022312 011203          MOV      (R2),R3        ;GET THE WAS DATA
6186 022314 104001          5$:      ERROR      1          ;DEC DELIVERED THE WRONG RESULT
6187
6188
6189
6190
6191 022316
6192 022316 000004          ;*****
6193 022320 012700 000330          ;*TEST 330      ASL DM1 TEST - N:C = 1000
6194 022324 013701 022346          ;*****
6195 022330 012702 063236          ;*TEST 330:
6196 022334 005004          SCOPE
6197 022336 012712 100000          MOV      #330,R0        ;CALL THE SCOPE LOOP UTILITY
6198 022342 000257          MOV      @#2$,R1        ;LOAD R0 WITH TEST NUMBER
6199 022344 000270          MOV      #MBUFO,R2      ;LOAD R1 WITH TEST INSTRUCTION WORD
6200
6201 022346 006312          2$:      ASL      (R2)        ;TEST THE ASL
6202
6203 022350 100403          BMI      3$            ;N:C = 0111 ?
6204 022352 001002          BNE      3$
6205 022354 102001          BVC      3$
6206 022356 103401          BCS      4$
6207
```

6208 022360 104001
6209 022362 020412
6210 022364 001402
6211
6212 022366 011203
6213 022370 104001
6214
6215
6216
6217
6218 022372
6219 022372 000004
6220 022374 012700 000331
6221 022400 013701 022424
6222 022404 012702 063236
6223 022410 012704 100000
6224 022414 012712 040000
6225 022420 000257
6226 022422 000265
6227
6228 022424 006312
6229
6230 022426 100003
6231 022430 001402
6232 022432 102001
6233 022434 103001
6234
6235 022436 104001
6236 022440 020412
6237 022442 001402
6238
6239 022444 011203
6240 022446 104001
6241
6242
6243
6244
6245 022450
6246 022450 000004
6247 022452 012700 000332
6248 022456 013701 022476
6249 022462 012702 063236
6250 022466 005004
6251 022470 005012
6252 022472 000257
6253 022474 000262
6254
6255 022476 006312
6256
6257 022500 100403
6258 022502 001002
6259 022504 102401
6260 022506 103001
6261
6262 022510 104001
6263 022512 020412

3\$: ERROR 1 ;ASL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST331 ;;BR IF YES

5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ASL DELIVERED THE WRONG RESULT

*TEST 331 ASL DM1 TEST - N:C = 0101

TST331:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #331,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;RESULT S / B = 100000
MOV #40000,(R2) ;[DEST] = 40000
CCC ;CLEAR CODES
265 ;N:C = 0101

2\$: ASL (R2) ;TEST THE ASL

BPL 3\$;N:C = 1010 ?
BEQ 3\$
BVC 3\$
BCC 4\$

3\$: ERROR 1 ;ASL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST332 ;;BR IF YES

5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ASL DELIVERED THE WRONG RESULT

*TEST 332 ASL DM1 TEST - N:C = 0010

TST332:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #332,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010

2\$: ASL (R2) ;TEST THE ASL

BMI 3\$;N:C = 0100 ?
BNE 3\$
BVS 3\$
BCC 4\$

3\$: ERROR 1 ;ASL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?

```

6264 022514 001402
6265
6266 022516 011203
6267 022520 104001
6268
6269
6270
6271
6272 022522
6273 022522 000004
6274 022524 012700 000333
6275 022530 013701 022554
6276 022534 012702 063236
6277 022540 012704 052525
6278 022544 012712 125252
6279 022550 000257
6280 022552 000275
6281
6282 022554 006112
6283
6284 022556 100403
6285 022560 001402
6286 022562 102001
6287 022564 103401
6288
6289 022566 104001
6290 022570 020412
6291 022572 001402
6292
6293 022574 011203
6294 022576 104001
6295
6296
6297
6298
6299 022600
6300 022600 000004
6301 022602 012700 000334
6302 022606 013701 022632
6303 022612 012702 063236
6304 022616 012704 125253
6305 022622 012712 052525
6306 022626 000257
6307 022630 000265
6308
6309 022632 006112
6310
6311 022634 100003
6312 022636 001402
6313 022640 102001
6314 022642 103001
6315
6316 022644 104001
6317 022646 020412
6318 022650 001402
6319
    
```

```

BEQ TST333 ;;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5$: ERROR 1 ;ASL DELIVERED THE WRONG RESULT

;*****
;#TEST 333 ROL DM1 TEST - N:C = 1101
;*****
TST333:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #333,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #52525,R4 ;RESULT S / B = 52525
MOV #125252,(R2) ;[DEST] = 125252
CCC ;CLEAR CODES
275 ;N:C = 1101

2$: ROL (R2) ;TEST THE ROL

BMI 3$ ;N:C = 0011 ?
BEQ 3$
BVC 3$
BCS 4$

3$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST334 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
5$: ERROR 1 ;ROL DELIVERED THE WRONG RESULT

;*****
;#TEST 334 ROL DM1 TEST - N:C = 0101
;*****
TST334:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #334,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #125253,R4 ;RESULT S / B = 125253
MOV #52525,(R2) ;[DEST] = 52525
CCC ;CLEAR CODES
265 ;N:C = 0101

2$: ROL (R2) ;TEST THE ROL

BPL 3$ ;N:C = 1010 ?
BEQ 3$
BVC 3$
BCC 4$

3$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST335 ;;BR IF YES
    
```

6320 022652 011203
6321 022654 104001
6322
6323
6324
6325
6326 022656
6327 022656 000004
6328 022660 012700 000335
6329 022664 013701 022704
6330 022670 012702 063236
6331 022674 005004
6332 022676 005012
6333 022700 000257
6334 022702 000262
6335
6336 022704 006112
6337
6338 022706 100403
6339 022710 001002
6340 022712 102401
6341 022714 103001
6342
6343 022716 104001
6344 022720 020412
6345 022722 001402
6346
6347 022724 011203
6348 022726 104001
6349
6350
6351
6352
6353 022730
6354 022730 000004
6355 022732 012700 000336
6356 022736 013701 022762
6357 022742 012702 063236
6358 022746 012704 100000
6359 022752 012712 077777
6360 022756 000257
6361 022760 000265
6362
6363 022762 005512
6364
6365 022764 100003
6366 022766 001402
6367 022770 102001
6368 022772 103001
6369
6370 022774 104001
6371 022776 020412
6372 023000 001402
6373
6374 023002 011203
6375 023004 104001

```

MOV (R2),R3 ;GET THE WAS DATA
5$: ERROR 1 ;ROL DELIVERED THE WRONG RESULT

;*****
;#TEST 335 ROL DM1 TEST - N:C = 0010
;*****
TST335:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #335,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;;DEST ADDR = MBUF0
CLR R4 ;;RESULT S / B = 000000
CLR (R2) ;;[DEST] = 000000
CCC ;;CLEAR CODES
SEV ;;N:C = 0010

2$: ROL (R2) ;TEST THE ROL
BMI 3$ ;N:C = 0100 ?
BNE 3$
BVS 3$
BCC 4$

3$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;;RESULT OK ?
BEQ TST336 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROL DELIVERED THE WRONG RESULT

;*****
;#TEST 336 ADC DM1 TEST - N:C = 0101
;*****
TST336:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #336,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;;DEST ADDR = MBUF0
MOV #100000,R4 ;;RESULT S / B = 100000
MOV #77777,(R2) ;;[DEST] = 77777
CCC ;;CLEAR CODES
265 ;;N:C = 0101

2$: ADC (R2) ;TEST THE ADC
BPL 3$ ;N:C = 1010 ?
BEQ 3$
BVC 3$
BCC 4$

3$: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;;RESULT OK ?
BEQ TST337 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ADC DELIVERED THE WRONG RESULT

```

```

6376
6377
6378
6379
6380 023006
6381 023006 000004
6382 023010 012700 000337
6383 023014 013701 023036
6384 023020 012702 063236
6385 023024 005004
6386 023026 012712 177777
6387 023032 000257
6388 023034 000273
6389
6390 023036 005512
6391
6392 023040 100403
6393 023042 001002
6394 023044 102401
6395 023046 103401
6396
6397 023050 104001
6398 023052 020412
6399 023054 001402
6400
6401 023056 011203
6402 023060 104001
6403
6404
6405
6406
6407 023062
6408 023062 000004
6409 023064 012700 000340
6410 023070 013701 023114
6411 023074 012702 063236
6412 023100 012704 177777
6413 023104 012712 177777
6414 023110 000257
6415 023112 000272
6416
6417 023114 005512
6418
6419 023116 100003
6420 023120 001402
6421 023122 102401
6422 023124 103001
6423
6424 023126 104001
6425 023130 020412
6426 023132 001402
6427
6428 023134 011203
6429 023136 104001
6430
6431

```

```

*****
;TEST 337 ADC DM1 TEST - N:C = 1011
*****
TST337:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #337,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
MOV #-1,(R2) ;[DEST] = 177777
CCC ;CLEAR CODES
273 ;N:C = 1011

25: ADC (R2) ;TEST THE ADC

BMI 35 ;N:C = 0101 ?
BNE 35
BVS 35
BCS 45

35: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,(R2) ;RESULT OK ?
BEQ TST340 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;ADC DELIVERED THE WRONG RESULT

*****
;TEST 340 ADC DM1 TEST - N:C = 1010
*****
TST340:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #340,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #-1,R4 ;RESULT S / B = 177777
MOV #-1,(R2) ;[DEST] = 177777
CCC ;CLEAR CODES
272 ;N:C = 1010

25: ADC (R2) ;TEST THE ADC

BPL 35 ;N:C = 1000 ?
BEQ 35
BVS 35
BCC 45

35: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,(R2) ;RESULT OK ?
BEQ TST341 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;ADC DELIVERED THE WRONG RESULT

*****

```

M11

6432
6433
6434 023140
6435 023140 000004
6436 023142 012700 000341
6437 023146 013701 023170
6438 023152 012702 063236
6439 023156 005004
6440 023160 012712 000001
6441 023164 000257
6442 023166 000273
6443
6444 023170 005612
6445
6446 023172 100403
6447 023174 001002
6448 023176 102401
6449 023200 103001
6450
6451 023202 104001
6452 023204 020412
6453 023206 001402
6454
6455 023210 011203
6456 023212 104001
6457
6458
6459
6460
6461 023214
6462 023214 000004
6463 023216 012700 000342
6464 023222 013701 023246
6465 023226 012702 063236
6466 023232 012704 077777
6467 023236 012712 100000
6468 023242 000257
6469 023244 000265
6470
6471 023246 005612
6472
6473 023250 100403
6474 023252 001402
6475 023254 102001
6476 023256 103001
6477
6478 023260 104001
6479 023262 020412
6480 023264 001402
6481
6482 023266 011203
6483 023270 104001
6484
6485
6486
6487

```
;*TEST 341 SBC DM1 TEST - N:C = 1011
*****
TST341:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #341,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
MOV #1,(R2) ;[DEST] = +1
CCC ;CLEAR CODES
273 ;N:C = 1011

25: SBC (R2) ;TEST THE SBC
;N:C = 0100 ?

BMI 35
BNE 35
BVS 35
BCC 45

35: ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,(R2) ;RESULT OK ?
BEQ TST342 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;SBC DELIVERED THE WRONG RESULT

*****
;*TEST 342 SBC DM1 TEST - N:C = 0101
*****
TST342:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #342,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #077777,R4 ;RESULT S / B = 077777
MOV #100000,(R2) ;[DEST] = 100000
CCC ;CLEAR CODES
265 ;N:C = 0101

25: SBC (R2) ;TEST THE SBC
;N:C = 0010 ?

BMI 35
BEQ 35
BVC 35
BCC 45

35: ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,(R2) ;RESULT OK ?
BEQ TST343 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;SBC DELIVERED THE WRONG RESULT

*****
;*TEST 343 SBC DM1 TEST - N:C = 1110
*****
```


6488 023272
6489 023272 000004
6490 023274 012700 000343
6491 023300 013701 023324
6492 023304 012702 063236
6493 023310 012704 000001
6494 023314 012712 000001
6495 023320 000257
6496 023322 000276

TST343:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #343,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;;DEST ADDR = MBUFO
MOV #1,R4 ;;RESULT S / B = 1
MOV #1,(R2) ;;(DEST) = 1
CCC ;CLEAR CODES
276 ;N:C = 1110

6497
6498 023324 005612
6499
6500 023326 100403
6501 023330 001402
6502 023332 102401
6503 023334 103001
6504

25:

SBC (R2) ;TEST THE SBC

BMI 35 ;N:C = 0000 ?
BEQ 35
BVS 35
BCC 45

6505 023336 104001
6506 023340 020412
6507 023342 001402
6508

35:

ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,(R2) ;;RESULT OK ?
BEQ TST344 ;;BR IF YES

6509 023344 011203
6510 023346 104001
6511

55:

MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SBC DELIVERED THE WRONG RESULT

6512
6513
6514

;TEST 344 SBC DMI TEST - N:C = 0111

6515 023350
6516 023350 000004
6517 023352 012700 000344
6518 023356 013701 023400
6519 023362 012702 063236
6520 023366 012704 177777
6521 023372 005012
6522 023374 000257
6523 023376 000267
6524

TST344:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #344,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;;DEST ADDR = MBUFO
MOV #-1,R4 ;;RESULT S / B = 177777
CLR (R2) ;;(DEST) = 000000
CCC ;CLEAR CODES
267 ;N:C = 0111

6525 023400 005612
6526
6527 023402 100003
6528 023404 001402
6529 023406 102401
6530 023410 103401
6531

25:

SBC (R2) ;TEST THE SBC

BPL 35 ;N:C = 1001 ?
BEQ 35
BVS 35
BCS 45

6532 023412 104001
6533 023414 020412
6534 023416 001402
6535

35:

ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,(R2) ;;RESULT OK ?
BEQ TST345 ;;BR IF YES

6536 023420 011203
6537 023422 104001
6538

55:

MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SBC DELIVERED THE WRONG RESULT

6539
6540
6541

;TEST 345 NEGB - MODE 0 TEST - N:C = 0110

6542 023424
6543 023424 000004

TST345:

SCOPE ;CALL THE SCOPE LOOP UTILITY

```

6544 023426 012700 000345      MOV      #345,R0      ;;LOAD R0 WITH TEST NUMBER
6545 023432 013701 023452      MOV      #25,R1      ;;LOAD R1 WITH TEST INSTRUCTION WORD
6546 023436 012704 177776      MOV      #177776,R4   ;;RESULT S / B = 376 (LO BYTE)
6547 023442 012703 177402      MOV      #177402,R3   ;;[DEST] = 177402
6548 023446 000257              CCC                ;;CLEAR FLAGS
6549 023450 000266              266                ;;N:C = 0110
6550
6551 023452 105403      2$:      NEGB      R3      ;TEST THE NEGB
6552
6553 023454 100003              BPL      3$          ;N:C = 1001
6554 023456 001402              BEQ      3$
6555 023460 102401              BVS      3$
6556 023462 103401              BCS      4$
6557
6558 023464 104002      3$:      ERROR      2      ;NEGB FAILED TO ALTER CODES PROPERLY
6559
6560 023466 020403      4$:      CMP      R4,R3      ;CORRECT RESULT ?
6561 023470 001401              BEQ      T$T346      ;;BR IF YES
6562
6563 023472 104002      5$:      ERROR      2      ;NEGB DELIVERED THE WRONG RESULT
6564
6565      ;;*****
6566      ;;*TEST 346      NEGB - MODE 0 TEST - N:C = 0011
6567      ;;*****
6568      †T$T346:
6569 023474 000004              SCOPE
6570 023476 012700 000346      MOV      #346,R0      ;CALL THE SCOPE LOOP UTILITY
6571 023502 013701 023522      MOV      #25,R1      ;LOAD R0 WITH TEST NUMBER
6572 023506 012704 177400      MOV      #177400,R4   ;LOAD R1 WITH TEST INSTRUCTION WORD
6573 023512 012703 177400      MOV      #177400,R3   ;RESULT S / B = 000 (LO BYTE)
6574 023516 000257              CCC                ;[DEST] = 177400
6575 023520 000263              263                ;CLEAR FLAGS
6576
6577 023522 105403      2$:      NEGB      R3      ;TEST THE NEGB
6578
6579 023524 100403              BMI      3$          ;N:C = 0100
6580 023526 001002              BNE      3$
6581 023530 102401              BVS      3$
6582 023532 103001              BCC      4$
6583
6584 023534 104002      3$:      ERROR      2      ;NEGB FAILED TO ALTER CODES PROPERLY
6585
6586 023536 020403      4$:      CMP      R4,R3      ;CORRECT RESULT ?
6587 023540 001401              BEQ      T$T347      ;;BR IF YES
6588
6589 023542 104002      5$:      ERROR      2      ;NEGB DELIVERED THE WRONG RESULT
6590
6591      ;;*****
6592      ;;*TEST 347      NEGB - MODE 0 TEST - N:C = 1101
6593      ;;*****
6594 023544 000004              SCOPE
6595 023544 012700 000347      MOV      #347,R0      ;CALL THE SCOPE LOOP UTILITY
6596 023546 013701 023572      MOV      #25,R1      ;LOAD R0 WITH TEST NUMBER
6597 023552 012704 177600      MOV      #177600,R4   ;LOAD R1 WITH TEST INSTRUCTION WORD
6598 023556 012703 177600      MOV      #177600,R3   ;RESULT S / B = 200 (LO BYTE)
6599 023562 012703 177600              MOV                ;[DEST] = 177600

```

```

6600 023566 000257          CCC          ;CLEAR FLAGS
6601 023570 000275          275          ;N:C = 1101
6602
6603 023572 105403          2$:  NEGB    R3          ;TEST THE NEGB
6604
6605 023574 100003          BPL     3$          ;N:C = 1011
6606 023576 001402          BEQ     3$
6607 023600 102001          BVC     3$
6608 023602 103401          BCS     4$
6609
6610 023604 104002          3$:  ERROR  2          ;NEGB FAILED TO ALTER CODES PROPERLY
6611
6612 023606 020403          4$:  CMP     R4,R3          ;CORRECT RESULT ?
6613 023610 001401          BEQ     TST350          ;;BR IF YES
6614
6615 023612 104002          5$:  ERROR  2          ;NEGB DELIVERED THE WRONG RESULT
6616

```

```

;*****
;TEST 350      CLRB - MODE 0 TEST - N:C = 1011
;*****
TST350:

```

```

6620 023614
6621 023614 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6622 023616 012700 000350          MOV     #350,R0          ;;LOAD R0 WITH TEST NUMBER
6623 023622 013701 023642          MOV     2#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
6624 023626 012704 177400          MOV     #177400,R4          ;RESULT S / B = 000 (LO BYTE)
6625 023632 012703 177777          MOV     #-1,R3          ;[DEST] = 177777
6626 023636 000257          CCC          ;CLEAR FLAGS
6627 023640 000273          273          ;N:C = 1011
6628

```

```

6629 023642 105003          2$:  CLRB    R3          ;TEST THE CLRB
6630
6631 023644 100403          BMI     3$          ;N:C = 0100 ?
6632 023646 001002          BNE     3$
6633 023650 102401          BVS     3$
6634 023652 103001          BCC     4$
6635
6636 023654 104002          3$:  ERROR  2          ;CLRB FAILED TO SET CODES PROPERLY
6637
6638 023656 020403          4$:  CMP     R4,R3          ;RESULT CORRECT ?
6639 023660 001401          BEQ     TST351          ;;BR IF YES
6640
6641 023662 104002          5$:  ERROR  2          ;CLRB DELIVERED THE WRONG RESULT
6642

```

```

;*****
;TEST 351      CLRB - MODE 0 TEST - N:C = 0100
;*****
TST351:

```

```

6643
6644
6645
6646 023664
6647 023664 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6648 023666 012700 000351          MOV     #351,R0          ;;LOAD R0 WITH TEST NUMBER
6649 023672 013701 023712          MOV     2#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
6650 023676 012704 177400          MOV     #177400,R4          ;RESULT S / B = 000 (LO BYTE)
6651 023702 012703 177777          MOV     #-1,R3          ;[DEST] = 177777
6652 023706 000257          CCC          ;CLEAR FLAGS
6653 023710 000264          SEZ          ;N:C = 0100
6654

```

```

6655 023712 105003          2$:  CLRB    R3          ;TEST THE CLRB

```

```

6656
6657 023714 100403          BMI    3$          ;N:C = 0100 ?
6658 023716 001002          BNE    3$
6659 023720 102401          BVS    3$
6660 023722 103001          BCC    4$
6661
6662 023724 104002          3$:    ERROR    2          ;CLRB FAILED TO SET CODES PROPERLY
6663
6664 023726 020403          4$:    CMP     R4,R3          ;RESULT CORRECT ?
6665 023730 001401          BEQ    TST352          ;;BR IF YES
6666
6667 023732 104002          5$:    ERROR    2          ;CLRB DELIVERED THE WRONG RESULT
6668
6669          ;*****
6670          ;*TEST 352      CLRB TEST - DM2 - ODD ADDRESS
6671          ;*****
6672          TST352:
6673 023734 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6674 023736 012700 000352          MOV     #352,R0          ;;LOAD R0 WITH TEST NUMBER
6675 023742 013701 023772          MOV     @#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
6676 023746 012702 063237          MOV     #MBUF0+1,R2          ;DEST ADDR = MBUF0+1
6677 023752 012704 000377          MOV     #377,R4          ;RESULT S / B = 377
6678 023756 012705 063236          MOV     #MBUF0,R5          ;POINT R5 TO CHECK RESULT
6679 023762 010203          MOV     R2,R3          ;R3 CONTAINS DEST ADDR
6680 023764 012715 177777          MOV     #-1,(R5)          ;[DEST] = 177777
6681 023770 000257          CCC          ;SCOPE SYNC
6682
6683 023772 105023          2$:    CLRB    (R3)+          ;TEST THE CLRB
6684
6685 023774 022703 063240          CMP     #MBUF0+2,R3          ;DID DEST REG GET INCREMENTED ?
6686 024000 001401          BEQ    4$          ;BR IF YES
6687
6688 024002 104005          3$:    ERROR    5          ;CLRB FAILED TO UPDATE DEST REG
6689
6690 024004 020415          4$:    CMP     R4,(R5)          ;CORRECT RESULT ?
6691 024006 001402          BEQ    TST353          ;;BR IF YES
6692
6693 024010 011503          5$:    MOV     (R5),R3          ;GET THE WAS DATA
6694 024012 104001          ERROR    1          ;CLRB DELIVERED WRONG RESULT
6695
6696          ;*****
6697          ;*TEST 353      CLRB TEST - DM1 - ODD ADDRESS
6698          ;*****
6699          TST353:
6700 024014 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6701 024016 012700 000353          MOV     #353,R0          ;;LOAD R0 WITH TEST NUMBER
6702 024022 013701 024052          MOV     @#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
6703 024026 012702 063237          MOV     #MBUF0+1,R2          ;DEST ADDR = MBUF0+1
6704 024032 012704 000377          MOV     #377,R4          ;RESULT S / B = 377
6705 024036 012705 063236          MOV     #MBUF0,R5          ;POINT R5 TO CHECK RESULT
6706 024042 010203          MOV     R2,R3          ;R3 CONTAINS DEST ADDR
6707 024044 012715 177777          MOV     #-1,(R5)          ;[DEST] = 177777
6708 024050 000257          CCC          ;SCOPE SYNC
6709
6710 024052 105013          2$:    CLRB    (R3)          ;TEST THE CLRB
6711

```

6712 024054 020415
6713 024056 001402
6714
6715 024060 011503
6716 024062 104001
6717
6718
6719
6720

CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST354 ;;BR IF YES
MOV (R5),R3 ;GET THE WAS DATA
3\$: ERROR 1 ;CLR8 DELIVERED WRONG RESULT

;TEST 354 CLR8 TEST - DM2 - EVEN ADDRESS

TST354:

6721 024064
6722 024064 000004
6723 024066 012700 000354
6724 024072 013701 024116
6725 024076 012702 063236
6726 024102 012704 177400
6727 024106 010203
6728 024110 012712 177777
6729 024114 000257
6730

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #354,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #177400,R4 ;RESULT S / B = 177400
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC

6731 024116 105023
6732
6733 024120 022703 063237
6734 024124 001401
6735
6736 024126 104005
6737

2\$: CLR8 (R3)+ ;TEST THE CLR8
CMP #MBUFD+1,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 4\$;;BR IF YES

6738 024130 020412
6739 024132 001402
6740
6741 024134 011203
6742 024136 104001
6743

3\$: ERROR 5 ;CLR8 FAILED TO UPDATE DEST REG
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST355 ;;BR IF YES

6744
6745
6746
6747 024140
6748 024140 000004
6749 024142 012700 000355
6750 024146 013701 024172
6751 024152 012702 063236
6752 024156 012704 177400
6753 024162 010203
6754 024164 012712 177777
6755 024170 000257
6756

5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CLR8 DELIVERED WRONG RESULT

;TEST 355 CLR8 TEST - DM1 - EVEN ADDRESS

TST355:

6757 024172 105013
6758
6759 024174 020412
6760 024176 001402
6761
6762 024200 011203
6763 024202 104001
6764
6765
6766
6767

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #355,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #177400,R4 ;RESULT S / B = 177400
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC

2\$: CLR8 (R3) ;TEST THE CLR8
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST356 ;;BR IF YES

3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CLR8 DELIVERED WRONG RESULT

;TEST 356 NEGB TEST - DM2 - ODD ADDRESS

6768 024204
6769 024204 000004
6770 024206 012700 000356
6771 024212 013701 024242
6772 024216 012702 063237
6773 024222 012704 000777
6774 024226 012705 063236
6775 024232 010203
6776 024234 012715 177777
6777 024240 000257
6778
6779 024242 105423
6780
6781 024244 022703 063240
6782 024250 001401
6783
6784 024252 104005
6785
6786 024254 020415
6787 024256 001402
6788
6789 024260 011503
6790 024262 104001
6791
6792
6793
6794
6795 024264
6796 024264 000004
6797 024266 012700 000357
6798 024272 013701 024334
6799
6800 024276 032737 000200 063160
6801 024304 001401
6802 024306 000000
6803 024310 012702 063237
6804 024314 012704 000777
6805 024320 012705 063236
6806 024324 010203
6807 024326 012715 177777
6808 024332 000257
6809
6810 024334 105413
6811
6812 024336 020415
6813 024340 001402
6814
6815 024342 011503
6816 024344 104001
6817
6818
6819
6820
6821 024346
6822 024346 000004
6823 024350 012700 000360

TST356:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #356,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO+1,R2 ;DEST ADDR = MBUFO+1
MOV #777,R4 ;RESULT S / B = 777
MOV #MBUFO,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R5) ;[DEST] = 177777
CCC ;SCOPE SYNC
2\$: NEGB (R3)+ ;TEST THE NEGB
CMP #MBUFO+2,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 4\$;BR IF YES
3\$: ERROR 5 ;NEGB FAILED TO UPDATE DEST REG
4\$: CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST357 ;BR IF YES
5\$: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;NEGB DELIVERED WRONG RESULT

;TEST 357 NEGB TEST - DM1 - ODD ADDRESS

TST357:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #357,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
.SBTTL USER CONTROLLED BREAKPOINT -- BIT7
BIT #BIT7,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK - DEPRESS CONTINUE TO RESTART
MOV #MBUFO+1,R2 ;DEST ADDR = MBUFO+1
MOV #777,R4 ;RESULT S / B = 777
MOV #MBUFO,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R5) ;[DEST] = 177777
CCC ;SCOPE SYNC
2\$: NEGB (R3) ;TEST THE NEGB
CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST360 ;BR IF YES
3\$: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;NEGB DELIVERED WRONG RESULT

;TEST 360 NEGB TEST - DM2 - EVEN ADDRESS

TST360:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #360,R0 ;LOAD RO WITH TEST NUMBER

```

6824 024354 013701 024400      MOV      2#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
6825 024360 012702 063236      MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
6826 024364 012704 177401      MOV      #177401,R4  ;RESULT S / B = 177401
6827 024370 010203              MOV      R2,R3       ;R3 CONTAINS DEST ADDR
6828 024372 012712 177777      MOV      #-1,(R2)    ;[DEST] = 177777
6829 024376 000257              CCC                  ;SCOPE SYNC
6830
6831 024400 105423      2$:      NEGB      (R3)+      ;TEST THE NEGB
6832
6833 024402 022703 063237      CMP      #MBUFO+1,R3 ;DID DEST REG GET INCREMENTED ?
6834 024406 001401      BEQ      4$           ;BR IF YES
6835
6836 024410 104005      3$:      ERROR      5           ;NEGB FAILED TO UPDATE DEST REG
6837
6838 024412 020412      4$:      CMP      R4,(R2)     ;CORRECT RESULT ?
6839 024414 001402      BEQ      T$T361      ;;BR IF YES
6840
6841 024416 011203      5$:      MOV      (R2),R3      ;GET THE WAS DATA
6842 024420 104001      ERROR      1           ;NEGB DELIVERED WRONG RESULT
6843
6844      ;:*****
6845      ;:TEST 361      NEGB TEST - DM1 - EVEN ADDRESS
6846      ;:*****
6847      T$T361:
6848      SCOPE
6849      MOV      #361,R0      ;CALL THE SCOPE LOOP UTILITY
6850      MOV      2#25,R1     ;LOAD R0 WITH TEST NUMBER
6851      MOV      #MBUFO,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
6852      MOV      #177401,R4  ;DEST ADDR = MBUFO
6853      MOV      R2,R3       ;RESULT S / B = 177401
6854      MOV      #-1,(R2)    ;R3 CONTAINS DEST ADDR
6855      MOV      #-1,(R2)    ;[DEST] = 177777
6856      CCC                  ;SCOPE SYNC
6857
6857 024454 105413      2$:      NEGB      (R3)       ;TEST THE NEGB
6858
6859 024456 020412      CMP      R4,(R2)     ;CORRECT RESULT ?
6860 024460 001402      BEQ      T$T362      ;;BR IF YES
6861
6862 024462 011203      3$:      MOV      (R2),R3      ;GET THE WAS DATA
6863 024464 104001      ERROR      1           ;NEGB DELIVERED WRONG RESULT
6864
6865      ;:*****
6866      ;:TEST 362      ADD TEST - SMO,DMO - N:C = 1010
6867      ;:*****
6868      T$T362:
6869      SCOPE
6870      MOV      #362,R0      ;CALL THE SCOPE LOOP UTILITY
6871      MOV      2#25,R1     ;LOAD R0 WITH TEST NUMBER
6872      CLR      R4           ;LOAD R1 WITH TEST INSTRUCTION WORD
6873      MOV      #-1,R5      ;RESULT S / B = 000000
6874      MOV      #+1,R3      ;SRC OPR = 177777
6875      CCC                  ;[DEST] = +1
6876      272                  ;CLEAR FLAGS
6877
6878 024516 060503      2$:      ADD      R5,R3       ;N:C = 1010
6879
    
```

```

6880 024520 100403          BMI    3$          ;N:C = 0101
6881 024522 001002          BNE    3$
6882 024524 102401          BVS    3$
6883 024526 103401          BCS    4$
6884
6885 024530 104002          3$:   ERROR    2          ;ADD FAILED TO ALTER CODES PROPERLY
6886
6887 024532 020403          4$:   CMP     R4,R3          ;CORRECT RESULT ?
6888 024534 001401          BEQ    TST363          ;;BR IF YES
6889
6890 024536 104002          5$:   ERROR    2          ;ADD DELIVERED THE WRONG RESULT
6891
6892
6893
6894

```

```

*****
;*TEST 363      ADD TEST - SMO,DMD - N:C = 0101
*****
TST363:

```

```

6895 024540          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6896 024540 000004          MOV     #363,R0          ;LOAD R0 WITH TEST NUMBER
6897 024542 012700 000363  MOV     J#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
6898 024546 013701 024572  MOV     #100006,R4        ;RESULT S / B = 100006
6899 024552 012704 100006  MOV     #77777,R5        ;SRC OPR = 77777
6900 024556 012705 077777  MOV     #7,R3           ;[DEST] = 7
6901 024562 012703 000007  CCC          ;CLEAR FLAGS
6902 024566 000257          265          ;N:C = 0101
6903 024570 000265
6904

```

```

6905 024572 060503          2$:   ADD     R5,R3          ;TEST THE ADD
6906
6907 024574 100003          BPL    3$          ;N:C = 1010
6908 024576 001402          BEQ    3$
6909 024600 102001          BVC    3$
6910 024602 103001          BCC    4$
6911
6912 024604 104002          3$:   ERROR    2          ;ADD FAILED TO ALTER CODES PROPERLY
6913
6914 024606 020403          4$:   CMP     R4,R3          ;CORRECT RESULT ?
6915 024610 001401          BEQ    TST364          ;;BR IF YES
6916
6917 024612 104002          5$:   ERROR    2          ;ADD DELIVERED THE WRONG RESULT
6918
6919
6920
6921

```

```

*****
;*TEST 364      ADD SM1,DMD TEST
*****
TST364:

```

```

6922 024614          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6923 024614 000004          MOV     #364,R0          ;LOAD R0 WITH TEST NUMBER
6924 024616 012700 000364  MOV     J#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
6925 024622 013701 024642  MOV     #DWTA,R4        ;RESULT S / B = #DWTA
6926 024626 012704 063246  MOV     #ATA,R5         ;SOURCE ADDR = ATA
6927 024632 012705 063222  CLR     R3             ;[DEST] = 0
6928 024636 005003          CCC          ;SCOPE SYNC
6929 024640 000257
6930
6931 024642 061503          2$:   ADD     (R5),R3        ;TEST THE ADD - SM1,DMD
6932
6933 024644 020403          CMP     R4,R3          ;RESULT = #DWTA?
6934 024646 001401          BEQ    4$             ;BR IF YES
6935

```



```

6936 024650 104002      3$:  ERROR 2          ;ADD DELIVERED WRONG RESULT
6937
6938 024652 022705 063222  4$:  CMP      #ATA,R5      ;DID ADD CHANGE REG.
6939 024656 001401          BEQ      TST365          ;;BR IF NOT
6940
6941 024660 104005      5$:  ERROR 5          ;REG GOT MODIFIED
6942
6943      ;*****
6944      ;*TEST 365      ADD SM2,DMO TEST
6945      ;*****
6946      †TST365:
6947 024662 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6948 024664 012700 000365  MOV      #365,R0      ;:LOAD R0 WITH TEST NUMBER
6949 024670 013701 024710  MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
6950 024674 012704 063246  MOV      #DWTA,R4      ;:RESULT S / B = #DWTA
6951 024700 012705 063222  MOV      #ATA,R5      ;:SOURCE ADDR = ATA
6952 024704 005003          CLR      R3          ;:[DEST] = 0
6953 024706 000257          CCC          ;:SCOPE SYNC
6954
6955 024710 062503      2$:  ADD      (R5)+,R3      ;TEST THE ADD - SM2,DMO
6956
6957 024712 020403          CMP      R4,R3      ;:RESULT = #DWTA
6958 024714 001401          BEQ      4$          ;:BR IF YES
6959
6960 024716 104002      3$:  ERROR 2          ;ADD DELIVERED WRONG RESULT
6961
6962 024720 022705 063224  4$:  CMP      #ATA+2,R5      ;DID ADD AUTO INCREMENT SOURCE REG?
6963 024724 001401          BEQ      TST366          ;;BR IF YES
6964
6965 024726 104005      5$:  ERROR 5          ;ADD FAILED TO UPDATE SOURCE REG.
6966
6967      ;*****
6968      ;*TEST 366      ADD SM3,DMO TEST
6969      ;*****
6970      †TST366:
6971 024730 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6972 024732 012700 000366  MOV      #366,R0      ;:LOAD R0 WITH TEST NUMBER
6973 024736 013701 024762  MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
6974 024742 012704 063246  MOV      #DWTA,R4      ;:RESULT S / B = #DWTA
6975 024746 012705 063232  MOV      #ATA+10,R5     ;:R5 POINTS TO SOURCE ADDR
6976 024752 010437 063236  MOV      R4,@#M$BUFD   ;:[SOURCE] = #DWTA
6977 024756 005003          CLR      R3          ;:[DEST] = 0
6978 024760 000257          CCC          ;:SCOPE SYNC
6979
6980 024762 063503      2$:  ADD      @(R5)+,R3      ;TEST THE ADD - SM3,DMO
6981
6982 024764 020437 063236  4$:  CMP      R4,@#M$BUFD   ;:RESULT = #DWTA?
6983 024770 001401          BEQ      4$          ;:BR IF YES
6984
6985 024772 104002      3$:  ERROR 2          ;ADD DELIVERED WRONG RESULT
6986
6987 024774 022705 063234  4$:  CMP      #ATA+12,R5     ;DID ADD AUTO INCREMENT SOURCE REG?
6988 025000 001401          BEQ      TST367          ;;BR IF YES
6989
6990 025002 104005      5$:  ERROR 5          ;ADD FAILED TO UPDATE SOURCE REG.
6991

```



```

7048 025140 012704 063236      MOV      #MBUFO,R4      ;RESULT S / B = MBUFO
7049 025144 012705 063222      MOV      #ATA,R5       ;BASE SOURCE ADDR = ATA
7050 025150 005003                CLR      R3            ;[DEST] = 0
7051 025152 000257                CCC                    ;SCOPE SYNC
7052
7053 025154 066503 000010      2$:     ADD      10(R5),R3 ;TEST THE ADD - SM6,DMO
7054
7055 025160 020403                CMP      R4,R3         ;RESULT =MBUFO?
7056 025162 001401                BEQ      TST372        ;;BR IF YES
7057
7058 025164 104002                3$:     ERROR    2      ;ADD DELIVERED WRONG RESULT
7059
7060
7061
7062
7063 025166                ;:*****
7064 025166 000004                ;:TEST 372      ADD SM7,DMO TEST
7065 025170 012700 000372                ;:*****
7066 025174 013701 025220      TST372: SCOPE          ;CALL THE SCOPE LOOP UTILITY
7067 025200 012704 063246      MOV      #372,R0       ;LOAD R0 WITH TEST NUMBER
7068 025204 012705 063222      MOV      @#25,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
7069 025210 010437 063236      MOV      #DWTA,R4      ;RESULT S / B = #DWTA
7070 025214 005003                MOV      #ATA,R5       ;BASE SOURCE ADDR = ATA
7071 025216 000257                MOV      R4,@#MBUFO    ;[SOURCE] = #DWTA
7072
7073 025220 067503 000010      2$:     ADD      @10(R5),R3 ;TEST THE ADD - SM7,DMO
7074
7075 025224 020403                CMP      R4,R3         ;RESULT = #DWTA?
7076 025226 001401                BEQ      TST373        ;;BR IF YES
7077
7078 025230 104002                3$:     ERROR    2      ;ADD DELIVERED WRONG RESULT
7079
7080
7081
7082
7083 025232                ;:*****
7084 025232 000004                ;:TEST 373      ADD SM1,DM1 TEST
7085 025234 012700 000373                ;:*****
7086 025240 013701 025264      TST373: SCOPE          ;CALL THE SCOPE LOOP UTILITY
7087 025244 012702 063236      MOV      #373,R0       ;LOAD R0 WITH TEST NUMBER
7088 025250 012704 063246      MOV      @#25,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
7089 025254 012705 063222      MOV      #MBUFO,R2     ;DEST ADDR = MBUFO
7090 025260 005012                MOV      #DWTA,R4      ;RESULT S / B = #DWTA
7091 025262 000257                MOV      #ATA,R5       ;SOURCE ADDR = ATA
7092
7093 025264 061512                2$:     ADD      (R5),(R2) ;TEST THE ADD - SM1,DM1
7094
7095 025266 020412                CMP      R4,(R2)       ;RESULT = #DWTA?
7096 025270 001402                BEQ      TST374        ;;BR IF YES
7097
7098 025272 011203                3$:     MOV      (R2),R3   ;GET WAS DATA
7099 025274 104001                ERROR    1            ;ADD DELIVERED WRONG RESULT
7100
7101
7102
7103

```

7104 025276
7105 025276 000004
7106 025300 012700 000374
7107 025304 013701 025330
7108 025310 012702 063236
7109 025314 012704 063246
7110 025320 012705 063222
7111 025324 005012
7112 025326 000257

TST374: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #374,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA,R4 ;:RESULT S / B = #DWTA
MOV #ATA,R5 ;:SOURCE ADDR = ATA
CLR (R2) ;:[DEST] = 0
CCC ;:SCOPE SYNC

7114 025330 062512
7115
7116 025332 020412
7117 025334 001402
7118
7119 025336 011203
7120 025340 104001

25: ADD (R5)+,(R2) ;:TEST THE ADD - SM2,DM1
CMP R4,(R2) ;:RESULT = #DWTA?
BEQ TST375 ;:BR IF YES
35: MOV (R2),R3 ;:GET WAS DATA
ERROR 1 ;:ADD DELIVERED WRONG RESULT

7121
7122
7123
7124

:::*****
:TEST 375 ADD SM1,DM2 TEST
:::*****

7125 025342
7126 025342 000004
7127 025344 012700 000375
7128 025350 013701 025376
7129 025354 012702 063236
7130 025360 012704 063246
7131 025364 012705 063222
7132 025370 010203
7133 025372 005012
7134 025374 000257

TST375: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #375,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA,R4 ;:RESULT S / B = #DWTA
MOV #ATA,R5 ;:SOURCE ADDR = ATA
MOV R2,R3 ;:[R3] = DEST ADDR
CLR (R2) ;:[DEST] = 0
CCC ;:SCOPE SYNC

7135
7136 025376 061523
7137
7138 025400 020412
7139 025402 001406

25: ADD (R5),(R3)+ ;:TEST THE ADD - SM1,DM2
CMP R4,(R2) ;:RESULT = #DWTA?
BEQ 45 ;:BR IF YES

7140
7141 025404 010337 063242
7142 025410 011203
7143 025412 104001

35: MOV R3,@#MBUF1 ;:SAVE UPDATED DEST ADDR
MOV (R2),R3 ;:GET WAS DATA
ERROR 1 ;:ADD DELIVERED WRONG RESULT

7144
7145 025414 013703 063242
7146 025420 022703 063240
7147 025424 001401
7148
7149 025426 104005

45: MOV @#MBUF1,R3 ;:RESTORE UPDATED DEST ADDR
CMP #MBUFO+2,R3 ;:DID ADD INCREMENT DEST REG
BEQ TST376 ;:BR IF YES
55: ERROR 5 ;:ADD FAILED TO UPDATE DEST REG

7150
7151
7152
7153

:::*****
:TEST 376 ADD SM2,DM2 TEST
:::*****

7154 025430
7155 025430 000004
7156 025432 012700 000376
7157 025436 013701 025464
7158 025442 012702 063236
7159 025446 012704 063246

TST376: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #376,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA,R4 ;:RESULT S / B = #DWTA

```

7160 025452 012705 063222      MOV      #ATA,R5      ;SOURCE ADDR = ATA
7161 025456 010203              MOV      R2,R3        ;[R3] = DEST ADDR
7162 025460 005012              CLR      (R2)         ;[DEST] = 0
7163 025462 000257              CCC                     ;SCOPE SYNC
7164
7165 025464 062523      2$:  ADD      (R5)+,(R3)+ ;TEST THE ADD - SM2,DM2
7166
7167 025466 020412              CMP      R4,(R2)     ;RESULT = #DWTA
7168 025470 001406              BEQ      4$          ;BR IF YES
7169
7170 025472 010337 063242      MOV      R3,@#MBUF1  ;SAVE UPDATED DEST ADDR
7171 025476 011203              MOV      (R2),R3     ;GET WAS DATA
7172 025500 104001      3$:  ERROR  1          ;ADD DELIVERED WRONG RESULT
7173
7174 025502 013703 063242      MOV      @#MBUF1,R3  ;RESTORE UPDATED DEST ADDR
7175 025506 022703 063240      4$:  CMP      #MBUF0+2,R3 ;DID ADD INCREMENT DEST REG?
7176 025512 001401              BEQ      TST377      ;;BR IF YES
7177
7178 025514 104005      5$:  ERROR  5          ;ADD FAILED TO UPDATE DEST REG
7179
7180
7181
7182
7183 025516
7184 025516 000004
7185 025520 012700 000377
7186 025524 013701 025554
7187 025530 012702 063236
7188 025534 012704 063246
7189 025540 012705 063222
7190 025544 012703 063232
7191 025550 005012
7192 025552 000257
7193
7194 025554 061533      2$:  ADD      (R5),@(R3)+ ;TEST THE ADD - SM1,DM3
7195
7196 025556 020412              CMP      R4,(R2)     ;RESULT = #DWTA?
7197 025560 001406              BEQ      4$          ;BR IF YES
7198
7199 025562 010337 063242      MOV      R3,@#MBUF1  ;SAVE R3
7200 025566 011203              MOV      (R2),R3     ;GET WAS DATA
7201 025570 104001      3$:  ERROR  1          ;ADD DELIVERED WRONG RESULT
7202
7203 025572 013703 063242      MOV      @#MBUF1,R3  ;RESTORE R3
7204 025576 022703 063234      4$:  CMP      #ATA+12,R3 ;DID ADD INCREMENT DEST REG
7205 025602 001401              BEQ      TST400      ;;BR IF YES
7206
7207 025604 104005      5$:  ERROR  5          ;ADD FAILED TO UPDATE DEST REG
7208
7209
7210
7211
7212 025606
7213 025606 000004
7214 025610 012700 000400
7215 025614 013701 025644

```

```

*****
;TEST 377      ADD SM1,DM3 TEST
*****
TST377:

```

```

SCOPE
MOV      #377,R0      ;CALL THE SCOPE LOOP UTILITY
MOV      @#2$,R1     ;LOAD R0 WITH TEST NUMBER
MOV      #MBUF0,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #DWTA,R4    ;DEST ADDR = MBUF0
MOV      #ATA,R5     ;RESULT S / B = #DWTA
MOV      #ATA+10,R3  ;SOURCE ADDR = ATA
CLR      (R2)        ;[R3] = ADDR OF DEST ADDR
CCC                     ;[DEST] = 0
SCOPE SYNC

```

```

2$:  ADD      (R5),@(R3)+ ;TEST THE ADD - SM1,DM3
CMP      R4,(R2)     ;RESULT = #DWTA?
BEQ      4$          ;BR IF YES

```

```

3$:  ERROR  1          ;ADD DELIVERED WRONG RESULT
MOV      @#MBUF1,R3  ;RESTORE R3
MOV      (R2),R3     ;GET WAS DATA

```

```

4$:  CMP      #ATA+12,R3 ;DID ADD INCREMENT DEST REG
BEQ      TST400      ;;BR IF YES
5$:  ERROR  5          ;ADD FAILED TO UPDATE DEST REG

```

```

*****
;TEST 400      ADD SM2,DM3 TEST
*****
TST400:

```

```

SCOPE
MOV      #400,R0     ;CALL THE SCOPE LOOP UTILITY
MOV      @#2$,R1    ;LOAD R0 WITH TEST NUMBER

```

```

7216 025620 012702 063236      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
7217 025624 012704 063246      MOV      #DWTA,R4      ;RESULT S / B = #DWTA
7218 025630 012705 063222      MOV      #ATA,R5      ;SOURCE ADDR = ATA
7219 025634 012703 063232      MOV      #ATA+10,R3    ;[R3] = ADDR OF DEST ADDR
7220 025640 005012                CLR      (R2)          ;[DEST] = 0
7221 025642 000257                CCC                      ;SCOPE SYNC
7222
7223 025644 062533      2$:  ADD      (R5)+,2(R3)+  ;TEST THE ADD - SM2,DM3
7224
7225 025646 020412      CMP      R4,(R2)      ;RESULT = #DWTA?
7226 025650 001406      BEQ      4$          ;BR IF YES
7227
7228 025652 010337 063242      MOV      R3,2#MBUF1   ;SAVE R3
7229 025656 011203      MOV      (R2),R3     ;GET WAS DATA
7230 025660 104001      3$:  ERROR  1          ;ADD DELIVERED WRONG RESULT
7231
7232 025662 013703 063242      MOV      2#MBUF1,R3  ;RESTORE R3
7233 025666 022703 063234      4$:  CMP      #ATA+12,R3 ;DID ADD INCREMENT DEST REG
7234 025672 001401      BEQ      TST401      ;;BR IF YES
7235
7236 025674 104005      5$:  ERROR  5          ;ADD FAILED TO UPDATE DEST REG
7237
7238
7239
7240
7241 025676
7242 025676 000004
7243 025700 012700 000401
7244 025704 013701 025734
7245 025710 012702 063236
7246 025714 012704 063246
7247 025720 012705 063222
7248 025724 012703 063240
7249 025730 005012
7250 025732 000257
7251
7252 025734 061543      2$:  ADD      (R5),-(R3) ;TEST THE ADD - SM1,DM4
7253
7254 025736 020412      CMP      R4,(R2)      ;RESULT = #DWTA?
7255 025740 001406      BEQ      4$          ;BR IF YES
7256
7257 025742 010337 063242      MOV      R3,2#MBUF1   ;SAVE R3
7258 025746 011203      MOV      (R2),R3     ;GET WAS DATA
7259 025750 104001      3$:  ERROR  1          ;ADD DELIVERED WRONG RESULT
7260
7261 025752 013703 063242      MOV      2#MBUF1,R3  ;RESTORE R3
7262 025756 020302      4$:  CMP      R3,R2     ;DID ADD INCREMENT DEST REG?
7263 025760 001401      BEQ      TST402      ;;BR IF YES
7264
7265 025762 104005      5$:  ERROR  5          ;ADD FAILED TO UPDATE DEST REG.
7266
7267
7268
7269
7270 025764
7271 025764 000004

```

```

*****
;TEST 401      ADD SM1,DM4 TEST
*****
TST401:

```

```

*****
;TEST 402      ADD SM2,DM4 TEST
*****
TST402:

```

```

7272 025766 012700 000402      MOV      #402,R0      ;;LOAD R0 WITH TEST NUMBER
7273 025772 013701 026022      MOV      @#25,R1     ;;LOAD R1 WITH TEST INSTRUCTION WORD
7274 025776 012702 063236      MOV      #MBUF0,R2   ;;DEST ADDR = MBUF0
7275 026002 012704 063246      MOV      #DWTA,R4    ;;RESULT S / B = #DWTA
7276 026006 012705 063222      MOV      #ATA,R5     ;;SOURCE ADDR = ATA
7277 026012 012703 063240      MOV      #MBUF0+2,R3 ;;R3 POINTS TO DEST ADDR +2
7278 026016 005012                CLR      (R2)        ;;[DEST] = 0
7279 026020 000257                CCC                    ;;SCOPE SYNC
7280
7281 026022 061543                2$:      ADD      (R5),-(R3)  ;;TEST THE ADD - SM2,DM4
7282
7283 026024 020412                CMP      R4,(R2)     ;;RESULT = #DWTA?
7284 026026 001406                BEQ      4$          ;;BR IF YES
7285
7286 026030 010337 063242      MOV      R3,@#MBUF1 ;;SAVE R3
7287 026034 011203                MOV      (R2),R3    ;;GET WAS DATA
7288 026036 104001                3$:      ERROR    1        ;;ADD DELIVERED WRONG RESULT
7289
7290 026040 013703 063242      MOV      @#MBUF1,R3 ;;RESTORE R3
7291 026044 020302                4$:      CMP      R3,R2     ;;DID ADD INCREMENT DEST REG?
7292 026046 001401                BEQ      TST403     ;;BR IF YES
7293
7294 026050 104005                5$:      ERROR    5        ;;ADD FAILED TO UPDATE DEST REG.
7295
7296
7297
7298
7299 026052
7300 026052 000004                ;;*****
7301 026054 012700 000403      ;;#TEST 403      ADD SM1,DMS TEST
7302 026060 013701 026110      ;;*****
7303 026064 012702 063236      TST403:
7304 026070 012704 063246      SCOPE
7305 026074 012705 063222      MOV      #403,R0    ;;CALL THE SCOPE LOOP UTILITY
7306 026100 012703 063234      MOV      @#25,R1    ;;LOAD R0 WITH TEST NUMBER
7307 026104 005012                MOV      #MBUF0,R2  ;;LOAD R1 WITH TEST INSTRUCTION WORD
7308 026106 000257                MOV      #DWTA,R4   ;;DEST ADDR = MBUF0
7309
7310 026110 061553                2$:      ADD      (R5),@-(R3)  ;;TEST THE ADD - SM1,DMS
7311
7312 026112 020412                CMP      R4,(R2)    ;;RESULT = #DWTA?
7313 026114 001406                BEQ      4$          ;;BR IF YES
7314
7315 026116 010337 063242      MOV      R3,@#MBUF1 ;;SAVE R3
7316 026122 011203                MOV      (R2),R3    ;;GET WAS DATA
7317 026124 104001                3$:      ERROR    1        ;;ADD DELIVERED WRONG RESULT
7318
7319 026126 013703 063242      MOV      @#MBUF1,R3 ;;RESTORE R3
7320 026132 022703 063232      4$:      CMP      #ATA+10,R3 ;;DID ADD DECREMENT DEST REG?
7321 026136 001401                BEQ      TST404     ;;BR IF YES
7322
7323 026140 104005                5$:      ERROR    5        ;;ADD FAILED TO UPDATE DEST REG.
7324
7325
7326
7327

```

7328 026142
7329 026142 000004
7330 026144 012700 000404
7331 026150 013701 026200
7332 026154 012702 063236
7333 026160 012704 063246
7334 026164 012705 063222
7335 026170 012703 063234
7336 026174 005012
7337 026176 000257

TST404: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #404,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #DWTA,R4 ;RESULT S / B = #DWTA
MOV #ATA,R5 ;SOURCE ADDR = ATA
MOV #ATA+12,R3 ;R3 CONTAINS ADDR OF DEST ADDR PLUS 2
CLR (R2) ;[DEST] = 0
CCC ;SCOPE SYNC

7338
7339 026200 062553
7340
7341 026202 020412
7342 026204 001406
7343
7344 026206 010337 063242
7345 026212 011203
7346 026214 104001
7347
7348 026216 013703 063242
7349 026222 022703 063232
7350 026226 001401

2\$: ADD (R5)+,@-(R3) ;TEST THE ADD - SM2,DM5
CMP R4,(R2) ;RESULT = #DWTA?
BEQ 4\$;BR IF YES
3\$: MOV R3,@#MBUF1 ;SAVE R3
MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT
4\$: MOV @#MBUF1,R3 ;RESTORE R3
CMP #ATA+10,R3 ;DID ADD DECREMENT DEST REG?
BEQ TST405 ;;BR IF YES

7351
7352 026230 104005
7353
7354
7355
7356

5\$: ERROR 5 ;ADD FAILED TO UPDATE DEST REG

;TEST 405 ADD SM1,DM6 TEST

7357 026232
7358 026232 000004
7359 026234 012700 000405
7360 026240 013701 026270
7361 026244 012702 063242
7362 026250 012704 063246
7363 026254 012705 063222
7364 026260 012703 063236
7365 026264 005012
7366 026266 000257

TST405: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #405,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD+4,R2 ;DEST ADDR = MBUFD+4
MOV #DWTA,R4 ;RESULT S / B = #DWTA
MOV #ATA,R5 ;SOURCE ADDR = ATA
MOV #MBUFD,R3 ;[R3] = BASE DEST ADDR
CLR (R2) ;[DEST] = 0
CCC ;SCOPE SYNC

7367
7368 026270 061563 000004
7369
7370 026274 020412
7371 026276 001402
7372
7373 026300 011203
7374 026302 104001
7375

2\$: ADD (R5),4(R3) ;TEST THE ADD - SM1,DM6
CMP R4,(R2) ;RESULT = #DWTA?
BEQ TST406 ;;BR IF YES
3\$: MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT

7376
7377
7378
7379 026304

;TEST 406 ADD SM2,DM6 TEST

7380 026304 000004
7381 026306 012700 000406
7382 026312 013701 026342
7383 026316 012702 063242

TST406: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #406,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD+4,R2 ;DEST ADDR = MBUFD+4


```

7384 026322 012704 063246      MOV      #DWT A,R4      ;RESULT S / B = #DWT A
7385 026326 012705 063222      MOV      #ATA,R5      ;SOURCE ADDR = ATA
7386 026332 012703 063236      MOV      #M BUFO,R3   ;[R3] = BASE DEST ADDR
7387 026336 005012              CLR      (R2)         ;[DEST] = 0
7388 026340 000257              CCC                     ;SCOPE SYNC
7389
7390 026342 062563 000004      2$:     ADD      (R5)+,4(R3) ;TEST THE ADD - SM2,DM6
7391
7392 026346 020412              CMP      R4,(R2)     ;RESULT = #DWT A?
7393 026350 001402              BEQ      TST407      ;;BR IF YES
7394
7395 026352 011203              MOV      (R2),R3     ;GET WAS DATA
7396 026354 104001      3$:     ERROR    1      ;ADD DELIVERED WRONG RESULT
7397
7398 ;*****
7399 ;*TEST 407      ADD SM1,DM7 TEST
7400 ;*****
7401 TST407:
7402      SCOPE
7403      MOV      #407,R0      ;CALL THE SCOPE LOOP UTILITY
7404      MOV      @#2$,R1     ;LOAD R0 WITH TEST NUMBER
7405      MOV      #M BUFO,R2  ;LOAD R1 WITH TEST INSTRUCTION WORD
7406      MOV      #DWT A,R4   ;DEST ADDR = M BUFO
7407      MOV      #ATA,R5     ;RESULT S / B = #DWT A
7408      MOV      R5,R3      ;SOURCE ADDR = ATA
7409      CLR      (R2)       ;BASE DEST ADDR = ATA
7410      CCC                     ;[DEST] = 0
7411      ;SCOPE SYNC
7412 026412 061573 000010      2$:     ADD      (R5),@10(R3) ;TEST THE ADD - SM1,DM7
7413
7414 026416 020412              CMP      R4,(R2)     ;RESULT = #DWT A?
7415 026420 001402              BEQ      TST410      ;;BR IF YES
7416
7417 026422 011203              MOV      (R2),R3     ;GET WAS DATA
7418 026424 104001      3$:     ERROR    1      ;ADD DELIVERED WRONG RESULT
7419
7420 ;*****
7421 ;*TEST 410      ADD SM2,DM7 TEST
7422 ;*****
7423 TST410:
7424      SCOPE
7425      MOV      #410,R0      ;CALL THE SCOPE LOOP UTILITY
7426      MOV      @#2$,R1     ;LOAD R0 WITH TEST NUMBER
7427      MOV      #M BUFO,R2  ;LOAD R1 WITH TEST INSTRUCTION WORD
7428      MOV      #DWT A,R4   ;DEST ADDR = M BUFO
7429      MOV      #ATA,R5     ;RESULT S / B = #DWT A
7430      MOV      R5,R3      ;SOURCE ADDR = ATA
7431      CLR      (R2)       ;BASE DEST ADDR = ATA
7432      CCC                     ;[DEST] = 0
7433      ;SCOPE SYNC
7434 026462 062573 000010      2$:     ADD      (R5)+,@10(R3) ;TEST THE ADD - SM2,DM7
7435
7436 026466 020412              CMP      R4,(R2)     ;RESULT = #DWT A?
7437 026470 001402              BEQ      TST411      ;;BR IF YES
7438
7439 026472 011203              MOV      (R2),R3     ;GET WAS DATA

```

```

7440 026474 104001 3$: ERROR 1 ;ADD DELIVERED WRONG RESULT
7441
7442
7443 ;*****
7444 ;*TEST 411 "XOR RA,RB" TEST - A=B=00000 N:C=1010
7445 ;*****
7446 026476 000004 T$T411: SCOPE ;CALL THE SCOPE LOOP UTILITY
7447 026500 012700 000411 MOV #411,R0 ;;LOAD R0 WITH TEST NUMBER
7448 026504 013701 026520 MOV J#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7449 026510 005004 CLR R4 ;RESULT AND MASK = 00000
7450 026512 005003 CLR R3 ;[DEST] = 00000
7451 026514 000257 CCC ;SCOPE SYNC
7452 026516 000272 272 ;MAKE N:C=1010
7453
7454 026520 074403 2$: XOR R4,R3 ;TEST THE XOR
7455
7456 026522 100403 BMI 3$ ;N:C=0100 ??
7457 026524 001002 BNE 3$
7458 026526 102401 BVS 3$
7459 026530 103001 BCC 4$
7460
7461 026532 104002 3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
7462
7463 026534 020403 4$: CMP R4,R3 ;RESULT CORRECT?
7464 026536 001401 BEQ T$T412 ;;BR IF YES
7465
7466 026540 104002 5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
7467
7468 ;*****
7469 ;*TEST 412 "XOR RA,RB" TEST - A=B=177777 N:C=0101
7470 ;*****
7471 026542 T$T412: SCOPE ;CALL THE SCOPE LOOP UTILITY
7472 026542 000004 MOV #412,R0 ;;LOAD R0 WITH TEST NUMBER
7473 026544 012700 000412 MOV J#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7474 026550 013701 026570 CLR R4 ;RESULT = 00000
7475 026554 005004 MOV #-1,R5 ;MASK = 177777
7476 026556 012705 177777 MOV R5,R3 ;[DEST]=177777
7477 026562 010503 MOV R5,R3 ;SCOPE SYNC
7478 026564 000257 CCC ;MAKE N:C=0101
7479 026566 000265 265
7480
7481 026570 074503 2$: XOR R5,R3 ;TEST THE XOR
7482
7483 026572 100403 BMI 3$ ;N:C=0101 ??
7484 026574 001002 BNE 3$
7485 026576 102401 BVS 3$
7486 026600 103401 BCS 4$
7487
7488 026602 104002 3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
7489
7490 026604 020403 4$: CMP R4,R3 ;RESULT CORRECT?
7491 026606 001401 BEQ T$T413 ;;BR IF YES
7492
7493 026610 104002 5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
7494
7495 ;*****

```

7496
7497
7498 026612
7499 026612 000004
7500 026614 012700 000413
7501 026620 013701 026644
7502 026624 012704 177777
7503 026630 012705 125252
7504 026634 012703 052525
7505 026640 000257
7506 026642 000266
7507
7508 026644 074503
7509
7510 026646 100003
7511 026650 001402
7512 026652 102401
7513 026654 103001
7514
7515 026656 104002
7516
7517 026660 020403
7518 026662 001401
7519
7520 026664 104002
7521
7522
7523
7524
7525 026666
7526 026666 000004
7527 026670 012700 000414
7528 026674 013701 026720
7529 026700 012704 177777
7530 026704 012705 052525
7531 026710 012703 125252
7532 026714 000257
7533 026716 000271
7534
7535 026720 074503
7536
7537 026722 100003
7538 026724 001402
7539 026726 102401
7540 026730 103401
7541
7542 026732 104002
7543
7544 026734 020403
7545 026736 001401
7546
7547 026740 104002
7548
7549
7550
7551

```
;*TEST 413 "XOR RA,RB" TEST - A=125252,B=052525 N:C=0110
*****
TST413:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #413,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S/B = 177777
MOV #125252,R5 ;MASK=125252
MOV #052525,R3 ;[DEST] = 052525
CCC ;SCOPE SYNC
266 ;MAKE N:C=0110

2$: XOR R5,R3 ;TEST THE XOR

BPL 3$ ;N:C=1000 ??
BEQ 3$
BVS 3$
BCC 4$

3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY

4$: CMP R4,R3 ;RESULT CORRECT?
BEQ TST414 ;;BR IF YES

5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
*****
;*TEST 414 "XOR RA,RB" TEST - A=052525,B=125252 N:C=1001
*****
TST414:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #414,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S/B = 177777
MOV #52525,R5 ;MASK=052525
MOV #125252,R3 ;[DEST] = 125252
CCC ;SCOPE SYNC
271 ;MAKE N:C=1001

2$: XOR R5,R3 ;TEST THE XOR

BPL 3$ ;N:C=1001 ??
BEQ 3$
BVS 3$
BCS 4$

3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY

4$: CMP R4,R3 ;RESULT CORRECT?
BEQ TST415 ;;BR IF YES

5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
*****
;*TEST 415 "XOR RA,(RB)" TEST - A=B=000000 N:C=1010
*****
```

7552 026742
7553 026742 000004
7554 026744 012700 000415
7555 026750 013701 026772
7556 026754 005004
7557 026756 005005
7558 026760 012702 063236
7559 026764 005012
7560 026766 000257
7561 026770 000272
7562
7563 026772 074512
7564
7565 026774 100403
7566 026776 001002
7567 027000 102401
7568 027002 103001
7569
7570 027004 104001
7571
7572 027006 020412
7573 027010 001402
7574
7575 027012 011203
7576 027014 104001
7577
7578
7579
7580
7581 027016
7582 027016 000004
7583 027020 012700 000416
7584 027024 013701 027052
7585 027030 005004
7586 027032 012705 177777
7587 027036 012702 063236
7588 027042 012712 177777
7589 027046 000257
7590 027050 000265
7591
7592 027052 074512
7593
7594 027054 100403
7595 027056 001002
7596 027060 102401
7597 027062 103401
7598
7599 027064 104001
7600
7601 027066 020412
7602 027070 001402
7603
7604 027072 011203
7605 027074 104001
7606
7607

```
TST415:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #415,R0                      ;LOAD R0 WITH TEST NUMBER
MOV      @#25,R1                      ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R4                          ;RESULT S / B = 000000
CLR      R5                          ;MASK = 000000
MOV      #MBUFO,R2                   ;DEST ADDR = MBUFO
CLR      (R2)                         ;[DEST] = 000000
CCC      272                          ;SCOPE SYNC
;MAKE N:C=1010

2$:   XOR      R5,(R2)                ;TEST THE XOR

      BMI      3$                    ;N:C = 0100 ??
      BNE      3$
      BVS      3$
      BCC      4$

3$:   ERROR    1                      ;XOR FAILED TO ALTER CODES PROPERLY

4$:   CMP      R4,(R2)                ;RESULT CORRECT?
      BEQ      TST416                ;;BR IF YES

5$:   MOV      (R2),R3                ;GET THE WAS DATA
      ERROR    1                      ;XOR DELIVERED THE WRONG RESULT

;*****
;TEST 416 "XOR RA,(RB)" TEST - A=B=177777 N:C=0101
;*****
TST416:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #416,R0                      ;LOAD R0 WITH TEST NUMBER
MOV      @#25,R1                      ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R4                          ;RESULT S / B = 000000
MOV      #-1,R5                       ;MASK = 177777
MOV      #MBUFO,R2                   ;DEST ADDR = MBUFO
MOV      #-1,(R2)                    ;[DEST] = 177777
CCC      265                          ;SCOPE SYNC
;MAKE N:C=0101

2$:   XOR      R5,(R2)                ;TEST THE XOR

      BMI      3$                    ;N:C = 0101 ??
      BNE      3$
      BVS      3$
      BCS      4$

3$:   ERROR    1                      ;XOR FAILED TO ALTER CODES PROPERLY

4$:   CMP      R4,(R2)                ;RESULT CORRECT?
      BEQ      TST417                ;;BR IF YES

5$:   MOV      (R2),R3                ;GET THE WAS DATA
      ERROR    1                      ;XOR DELIVERED THE WRONG RESULT

;*****
```

```

7608 ;*TEST 417 "XOR RA,(RB)" TEST - A=125252,B=052525 N:C=0110
7609 ;*****
7610 027076 000004 TST417: SCOPE ;CALL THE SCOPE LOOP UTILITY
7611 027076 012700 000417 MOV #417,R0 ;LOAD RO WITH TEST NUMBER
7612 027100 013701 027146 MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7613 027104 013701 027146 .SBTTL USER CONTROLLED BREAKPOINT -- BIT8
7614 027110 032737 000400 063160 BIT #BIT8,@#BPTLOC ;BREAKPOINT HALT SET ??
7615 027116 001401 BEQ .+4 ;BR IF NOT
7616 027120 000000 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
7617 027122 012704 177777 MOV #-1,R4 ;RESULT S/B = 177777
7618 027126 012705 125252 MOV #125252,R5 ;MASK = 125252
7619 027132 012702 063236 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
7620 027136 012712 052525 MOV #052525,(R2) ;[DEST] = 052525
7621 027142 000257 CCC ;SCOPE SYNC
7622 027144 000266 266 ;MAKE N:C=0110
7623
7624
7625 027146 074512 25: XOR R5,(R2) ;TEST THE XOR
7626
7627 027150 100003 BPL 35 ;N:C = 1000 ??
7628 027152 001402 BEQ 35
7629 027154 102401 BVS 35
7630 027156 103001 BCC 45
7631
7632 027160 104001 35: ERROR 1 ;XOR FAILED TO ALTER CODES PROPERLY
7633
7634 027162 020412 45: CMP R4,(R2) ;RESULT CORRECT?
7635 027164 001402 BEQ TST420 ;BR IF YES
7636
7637 027166 011203 MOV (R2),R3 ;GET THE WAS DATA
7638 027170 104001 55: ERROR 1 ;XOR DELIVERED THE WRONG RESULT
7639
7640 ;*****
7641 ;*TEST 420 "XOR RA,(RB)" TEST - A=052525,B=125252 N:C=1001
7642 ;*****
7643 027172 TST420: SCOPE ;CALL THE SCOPE LOOP UTILITY
7644 027172 000004 MOV #420,R0 ;LOAD RO WITH TEST NUMBER
7645 027174 012700 000420 MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7646 027200 013701 027230 MOV #-1,R4 ;RESULT S/B = 177777
7647 027204 012704 177777 MOV #52525,R5 ;MASK = 052525
7648 027210 012705 052525 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
7649 027214 012702 063236 MOV #125252,(R2) ;[DEST] = 125252
7650 027220 012712 125252 CCC ;SCOPE SYNC
7651 027224 000257 271 ;MAKE N:C=1001
7652 027226 000271
7653
7654 027230 074512 25: XOR R5,(R2) ;TEST THE XOR
7655
7656 027232 100003 BPL 35 ;N:C = 1001 ??
7657 027234 001402 BEQ 35
7658 027236 102401 BVS 35
7659 027240 103401 BCS 45
7660
7661 027242 104001 35: ERROR 1 ;XOR FAILED TO ALTER CODES PROPERLY
7662
7663 027244 020412 45: CMP R4,(R2) ;RESULT CORRECT?

```

7664 027246 001402
7665
7666 027250 011203
7667 027252 104001
7668
7669
7670
7671
7672 027254
7673 027254 000004
7674 027256 012700 000421
7675 027262 013701 027302
7676 027266 005004
7677 027270 012703 052525
7678 027274 010305
7679 027276 000257
7680 027300 000273
7681
7682 027302 160503
7683
7684 027304 100403
7685 027306 001002
7686 027310 102401
7687 027312 103001
7688
7689 027314 104002
7690
7691 027316 020304
7692 027320 001401
7693
7694 027322 104002
7695
7696
7697
7698
7699 027324
7700 027324 000004
7701 027326 012700 000422
7702 027332 013701 027352
7703 027336 005004
7704 027340 012703 125252
7705 027344 010305
7706 027346 000257
7707 027350 000273
7708
7709 027352 160503
7710
7711 027354 100403
7712 027356 001002
7713 027360 102401
7714 027362 103001
7715
7716 027364 104002
7717
7718 027366 020304
7719 027370 001401

BEQ TST421 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;XOR DELIVERED THE WRONG RESULT
:;*****
:;TEST 421 SUB TEST SMO,DMD - (SRC) = (DEST) = +,+
:;*****
TST421:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #421,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 0
MOV #052525,R3 ;[R3] = DEST OP = 52525
MOV R3,R5 ;[R5] = SRC OP = 52525
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2\$: SUB R5,R3 ;TEST THE SUB
BMI 3\$
BNE 3\$;DID N:C = 0100
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;WAS RESULT = 0?
BEQ TST422 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
:;*****
:;TEST 422 SUB TEST SMO,DMD - (SRC) = (DEST) = -,-
:;*****
TST422:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #422,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 0
MOV #125252,R3 ;[R3] = DEST OP = 125252
MOV R3,R5 ;[R5] = SOURCE OP = 125252
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2\$: SUB R5,R3 ;TEST THE SUB
BMI 3\$
BNE 3\$;N:C = 0100?
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;RESULT = 0?
BEQ TST423 ;;BR IF YES

```

7720
7721 027372 104002      5$:      ERROR      2      ;SUB DELIVERED WRONG RESULT
7722
7723      ;*****
7724      ;*TEST 423      SUB TEST SMO,DMO - (SRC) = (DEST) = -,+
7725      ;*****
7726      †ST423:
7727 027374 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
7728 027376 012700 000423  MOV      #423,R0      ;LOAD R0 WITH TEST NUMBER
7729 027402 013701 027426  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
7730 027406 012704 000002  MOV      #2,R4      ;RESULT S / B = 2
7731 027412 012703 000001  MOV      #1,R3      ;[R3] = DEST OP = 1
7732 027416 012705 177777  MOV      #-1,R5      ;[R5] = SRC OP = -1
7733 027422 000257      CCC      ;CLEAR FLAGS
7734 027424 000276      276      ;MAKE N:C = 1110
7735
7736 027426 160503      2$:      SUB      R5,R3      ;TEST THE SUB
7737
7738 027430 100403      BMI      3$
7739 027432 001402      BEQ      3$      ;N:C = 0001
7740 027434 102401      BVS      3$
7741 027436 103401      BCS      4$
7742
7743 027440 104002      3$:      ERROR      2      ;SUB FAILED TO ALTER CODES PROPERLY
7744
7745 027442 020304      4$:      CMP      R3,R4      ;RESULT = +2?
7746 027444 001401      BEQ      TST424      ;;BR IF YES
7747
7748 027446 104002      5$:      ERROR      2      ;SUB DELIVERED WRONG RESULT
7749
7750      ;*****
7751      ;*TEST 424      SUB TEST SMO,DMO (SRC) = -(DEST) = +,-
7752      ;*****
7753      †ST424:
7754 027450 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
7755 027452 012700 000424  MOV      #424,R0      ;LOAD R0 WITH TEST NUMBER
7756 027456 013701 027502  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
7757 027462 012704 177776  MOV      #-2,R4      ;RESULT S / B = -2
7758 027466 012703 177777  MOV      #-1,R3      ;[R3] = [DEST] = -1
7759 027472 012705 000001  MOV      #1,R5      ;[R5] = [SOURCE] = +1
7760 027476 000257      CCC      ;CLEAR FLAGS
7761 027500 000267      267      ;MAKE N:C = 0111
7762
7763 027502 160503      2$:      SUB      R5,R3      ;TEST THE SUB
7764
7765 027504 100003      BPL      3$
7766 027506 001402      BEQ      3$      ;N:C = 1000
7767 027510 102401      BVS      3$
7768 027512 103001      BCC      4$
7769
7770 027514 104002      3$:      ERROR      2      ;SUB DID NOT ALTER CODES PROPERLY
7771
7772 027516 020403      4$:      CMP      R4,R3      ;RESULT = -2?
7773 027520 001401      BEQ      TST425      ;;BR IF YES
7774
7775 027522 104002      5$:      ERROR      2      ;SUB DELIVERED WRONG RESULT

```

```

7776
7777
7778
7779
7780 027524
7781 027524 000004
7782 027526 012700 000425
7783 027532 013701 027556
7784 027536 012704 077777
7785 027542 012703 100000
7786 027546 012705 000001
7787 027552 000257
7788 027554 000274
7789
7790 027556 160503
7791
7792 027560 100403
7793 027562 001402
7794 027564 102001
7795 027566 103001
7796
7797 027570 104002
7798
7799 027572 020304
7800 027574 001401
7801
7802 027576 104002
7803
7804
7805
7806
7807 027600
7808 027600 000004
7809 027602 012700 000426
7810 027606 013701 027634
7811 027612 012702 063236
7812 027616 012704 177777
7813 027622 012705 000001
7814 027626 005012
7815 027630 000257
7816 027632 000266
7817
7818 027634 160512
7819
7820 027636 100003
7821 027640 001402
7822 027642 102401
7823 027644 103401
7824
7825 027646 104001
7826
7827 027650 020412
7828 027652 001402
7829
7830 027654 011203
7831 027656 104001

```

```

*****
;TEST 425 SUB TEST SMO,DMO - "V" BIT SETS
*****
TST425:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #425,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #77777,R4 ;RESULT = 77777
MOV #100000,R3 ;[R3] = DEST OP = 100000
MOV #1,R5 ;[R5] = SRC OP = 1
CCC ;CLEAR FLAGS
274 ;MAKE N:C = 1100

2$: SUB R5,R3 ;TEST THE SUB

BMI 3$
BEQ 3$ ;N:C = 0011 ("V" BIT SHOULD SET)
BVC 3$
BCC 4$

3$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY

4$: CMP R3,R4 ;RESULT = 77777?
BEQ TST426 ;;BR IF YES

5$: ERROR 2 ;SUB DELIVERED WRONG RESULT

*****
;TEST 426 SUB TEST - SMO,DMI - N:C = 0110
*****
TST426:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #426,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #-1,R4 ;RESULT S / B = 177777
MOV #+1,R5 ;SRC OPR = +1
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110

2$: SUB R5,(R2) ;TEST THE SUB

BPL 3$
BEQ 3$ ;N:C = 1001
BVS 3$
BCS 4$

3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST427 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SUB DELIVERED THE WRONG RESULT

```



```

7832
7833
7834
7835
7836 027660
7837 027660 000004
7838 027662 012700 000427
7839 027666 013701 027714
7840 027672 012702 063236
7841 027676 005004
7842 027700 012705 177777
7843 027704 012712 177777
7844 027710 000257
7845 027712 000272
7846
7847 027714 160512
7848
7849 027716 100403
7850 027720 001002
7851 027722 102401
7852 027724 103001
7853
7854 027726 104001
7855
7856 027730 020412
7857 027732 001402
7858
7859 027734 011203
7860 027736 104001
7861
7862
7863
7864
7865 027740
7866 027740 000004
7867 027742 012700 000430
7868 027746 013701 027774
7869 027752 012702 063236
7870 027756 012704 077777
7871 027762 012705 000001
7872 027766 012712 100000
7873 027772 000257
7874
7875 027774 160512
7876
7877 027776 100403
7878 030000 001402
7879 030002 102001
7880 030004 103001
7881
7882 030006 104001
7883
7884 030010 020412
7885 030012 001402
7886
7887 030014 011203

```

```

*****
;*TEST 427 SUB TEST - SMO,DM1 - N:C = 1010
*****
†ST427:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #427,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
MOV #-1,R5 ;SRC OPR = 177777
MOV #-1,(R2) ;[DEST] = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010

25: SUB R5,(R2) ;TEST THE SUB

BMI 35 ;N:C = 0100
BNE 35
BVS 35
BCC 45

35: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY

45: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST430 ;;BR IF YES

55: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SUB DELIVERED THE WRONG RESULT

*****
;*TEST 430 SUB TEST - SMO,DM1 - N:C = 0000
*****
†ST430:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #430,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #77777,R4 ;RESULT S / B = 77777
MOV #+1,R5 ;SRC OPR = +1
MOV #100000,(R2) ;[DEST] = 100000
CCC ;CLEAR FLAGS

25: SUB R5,(R2) ;TEST THE SUB

BMI 35 ;N:C = 0010
BEQ 35
BVC 35
BCC 45

35: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY

45: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST431 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA

```

M13

7888 030016 104001
7889
7890
7891
7892
7893 030020
7894 030020 000004
7895 030022 012700 000431
7896 030026 013701 030050
7897 030032 012704 177777
7898 030036 012705 063760
7899 030042 005003
7900 030044 000257
7901 030046 000266
7902
7903 030050 161503
7904
7905 030052 100003
7906 030054 001402
7907 030056 102401
7908 030060 103401
7909
7910 030062 104002
7911
7912 030064 020403
7913 030066 001401
7914
7915 030070 104002
7916
7917
7918
7919
7920 030072
7921 030072 000004
7922 030074 012700 000432
7923 030100 013701 030120
7924 030104 005004
7925 030106 012705 063250
7926 030112 011503
7927 030114 000257
7928 030116 000272
7929
7930 030120 161503
7931
7932 030122 100403
7933 030124 001002
7934 030126 102401
7935 030130 103001
7936
7937 030132 104002
7938
7939 030134 020403
7940 030136 001401
7941
7942 030140 104002
7943

5\$: ERROR 1 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 431 SUB TEST - SM1,DMO - N:C = 0110
;*****
†T431:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #431,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
MOV #DWTB+2,R5 ;SRC ADDR = DWTB+2
CLR R3 ;[DEST] = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2\$: SUB (R5),R3 ;TEST THE SUB
BPL 3\$;N:C = 1001
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T432 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 432 SUB TEST - SM1,DMO - N:C = 1010
;*****
†T432:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #432,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #DWTB+2,R5 ;SRC ADDR = DWTB+2
MOV (R5),R3 ;[DEST] = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010
2\$: SUB (R5),R3 ;TEST THE SUB
BMI 3\$;N:C = 0100
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T433 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT

7944
7945
7946
7947 030142
7948 030142 000004
7949 030144 012700 000433
7950 030150 013701 030176
7951 030154 012704 077777
7952 030160 012705 063242
7953 030164 012703 100000
7954 030170 012715 000001
7955 030174 000257
7956
7957 030176 161503
7958
7959 030200 100403
7960 030202 001402
7961 030204 102001
7962 030206 103001
7963
7964 030210 104002
7965
7966 030212 020403
7967 030214 001401
7968
7969 030216 104002
7970
7971
7972
7973
7974 030220
7975 030220 000004
7976 030222 012700 000434
7977 030226 013701 030260
7978 030232 012702 063236
7979 030236 012704 177777
7980 030242 012705 063242
7981 030246 012715 000001
7982 030252 005012
7983 030254 000257
7984 030256 000266
7985
7986 030260 161512
7987
7988 030262 100003
7989 030264 001402
7990 030266 102401
7991 030270 103401
7992
7993 030272 104001
7994
7995 030274 020412
7996 030276 001402
7997
7998 030300 011203
7999 030302 104001

```
*****
*TEST 433      SUB TEST - SM1,DMO - N:C = 0000
*****
TST433:
SCOPE
MOV #433,R0      ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1      ;LOAD R0 WITH TEST NUMBER
MOV #77777,R4   ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF1,R5   ;RESULT S / B = 77777
MOV #100000,R3  ;SRC ADDR =MBUF1
MOV #+1,(R5)    ;[DEST] = 100000
CCC             ;SRC OPR = +1
               ;CLEAR FLAGS
25:  SUB (R5),R3 ;TEST THE SUB
               ;N:C = 0010
      BMI 35
      BEQ 35
      BVC 35
      BCC 45
35:  ERROR 2     ;SUB FAILED TO ALTER CODES PROPERLY
45:  CMP R4,R3  ;CORRECT RESULT ?
      BEQ TST434 ;;BR IF YES
55:  ERROR 2     ;SUB DELIVERED THE WRONG RESULT
*****
*TEST 434      SUB SM1,DM1 TEST - N:C = 0110
*****
TST434:
SCOPE
MOV #434,R0      ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1      ;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4      ;DEST ADDR = MBUFO
MOV #MBUF1,R5   ;RESULT S / B = 177777
MOV #+1,(R5)   ;SOURCE ADDR = MBUF1
CLR (R2)        ;[SOURCE] = 000001
CCC             ;[DEST] = 000000
266            ;CLEAR FLAGS
               ;N:C = 0110
25:  SUB (R5),(R2) ;TEST THE SUB
               ;N:C = 1001 ?
      BPL 35
      BEQ 35
      BVS 35
      BCS 45
35:  ERROR 1     ;SUB FAILED TO ALTER CODES PROPERLY
45:  CMP R4,(R2) ;CORRECT RESULT ?
      BEQ TST435 ;;BR IF YES
55:  MOV (R2),R3 ;GET THE WAS DATA
      ERROR 1    ;SUB DELIVERED THE WRONG RESULT
```

```

8000
8001
8002
8003
8004 030304
8005 030304 000004
8006 030306 012700 000435
8007 030312 013701 030346
8008 030316 012702 063236
8009 030322 012704 177777
8010 030326 012705 063242
8011 030332 012715 000001
8012 030336 005012
8013 030340 010203
8014 030342 000257
8015 030344 000266
8016
8017 030346 161523
8018
8019 030350 100003
8020 030352 001402
8021 030354 102401
8022 030356 103401
8023
8024 030360 104005
8025
8026 030362 020412
8027 030364 001402
8028
8029 030366 011203
8030 030370 104001
8031
8032
8033
8034
8035 030372
8036 030372 000004
8037 030374 012700 000436
8038 030400 013701 030424
8039 030404 012702 063236
8040 030410 012704 125252
8041 030414 010205
8042 030416 012712 052526
8043 030422 000257
8044
8045 030424 005425
8046
8047 030426 020412
8048 030430 001402
8049
8050 030432 011203
8051 030434 104001
8052
8053 030436 022705 063240
8054 030442 001401
8055

```

```

*****
;TEST 435 SUB SM1,DM2 TEST - N:C = 0110
*****
TST435:
SCOPE
MOV #435,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;DEST ADDR = MBUF0
MOV #MBUF1,R5 ;RESULT S / B = 177777
CLR (R2) ;SOURCE ADDR = MBUF1
MOV R2,R3 ;[SOURCE] = 000001
CCC ;[DEST] = 000000
266 ;R3 GETS DEST ADDR
;CLEAR FLAGS
;N:C = 0110

25: SUB (R5),(R3)+ ;TEST THE SUB
;N:C = 1001 ?

BPL 35
BEQ 35
BVS 35
BCS 45

35: ERROR 5 ;SUB FAILED TO ALTER CODES PROPERLY

45: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST436 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
55: ERROR 1 ;SUB DELIVERED THE WRONG RESULT

*****
;TEST 436 NEG DM2 TEST
*****
TST436:
SCOPE
MOV #436,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125252,R4 ;DEST ADDR = MBUF0
MOV R2,R5 ;RESULT S / B = 125252
MOV #52526,(R2) ;[R5] = DEST ADDR
CCC ;[DEST] = 52526
;SCOPE SYNC

25: NEG (R5)+ ;TEST THE NEG - MODE 2

CMP R4,(R2) ;RESULT = 125252?
BEQ 45 ;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
35: ERROR 1 ;NEG DELIVERED WRONG RESULT

45: CMP #MBUF0+2,R5 ;DID REG. GET AUTO INCREMENTED?
BEQ TST437 ;;BR IF YES

```

8056 030444 104005
8057
8058
8059
8060
8061 030446
8062 030446 000004
8063 030450 012700 000437
8064 030454 013701 030502
8065 030460 012702 063236
8066 030464 012704 125252
8067 030470 012705 063232
8068 030474 012712 052526
8069 030500 000257
8070
8071 030502 005435
8072
8073 030504 020412
8074 030506 001402
8075
8076 030510 011203
8077 030512 104001
8078
8079 030514 022705 063234
8080 030520 001401
8081
8082 030522 104005
8083
8084
8085
8086
8087 030524
8088 030524 000004
8089 030526 012700 000440
8090 030532 013701 030560
8091 030536 012702 063236
8092 030542 012704 125252
8093 030546 012705 063240
8094 030552 012712 052526
8095 030556 000257
8096
8097 030560 005445
8098
8099 030562 020412
8100 030564 001402
8101
8102 030566 011203
8103 030570 104001
8104
8105 030572 020502
8106 030574 001401
8107
8108 030576 104005
8109
8110
8111

```
5$: ERROR 5 ;NEG FAILED TO UPDATE REG.
;*****
;TEST 437 NEG DM3 TEST
;*****
TST437:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #437,R0 ;LOAD R0 WITH TEST NUMBER
MOV 2#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #ATA+10,R5 ;[ATA+10] = MBUF0
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG 2(R5)+ ;TEST THE NEG - MODE 3

CMP R4,(R2) ;RESULT = 125252?
BEQ 4$ ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
3$: ERROR 1 ;NEG DELIVERED WRONG RESULT

CMP #ATA+12,R5 ;DID REG GET AUTO INCREMENTED?
BEQ TST440 ;;BR IF YES

5$: ERROR 5 ;NEG FAILED TO UPDATE REG.
;*****
;TEST 440 NEG DM4 TEST
;*****
TST440:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #440,R0 ;LOAD R0 WITH TEST NUMBER
MOV 2#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #MBUF0+2,R5 ;[R5] = DEST ADDR + 2
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG -(R5) ;TEST THE NEG - MODE 4

CMP R4,(R2) ;RESULT = 125252?
BEQ 4$ ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
3$: ERROR 1 ;NEG DELIVERED WRONG RESULT

CMP R5,R2 ;DID REG GET AUTO INCREMENTED?
BEQ TST441 ;;BR IF YES

5$: ERROR 5 ;NEG FAILED TO UPDATE REG
;*****
;TEST 441 NEG DM5 TEST
```

```

8112
8113 030600
8114 030600 000004
8115 030602 012700 000441
8116 030606 013701 030634
8117 030612 012702 063236
8118 030616 012704 125252
8119 030622 012705 063234
8120 030626 012712 052526
8121 030632 000257
8122
8123 030634 005455
8124
8125 030636 020412
8126 030640 001402
8127
8128 030642 011203
8129 030644 104001
8130
8131 030646 022705 063232
8132 030652 001401
8133
8134 030654 104005
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149 030712 005465 000002
8150
8151 030716 020412
8152 030720 001402
8153
8154 030722 011203
8155 030724 104001
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167

```

```

*****
TST441:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #441,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBOFD,R2 ;DEST ADDR = MBOFD
MOV #125252,R4 ;RESULT S / B = 125252
MOV #ATA+12,R5 ;[R5] = (ADR OF MBOFD) +2
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG @-(R5) ;TEST THE NEG - MODE 5

CMP R4,(R2) ;RESULT = 125252?
BEQ 4$ ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
3$: ERROR 1 ;NEG DELIVERED WRONG RESULT

CMP #ATA+10,R5 ;DID NEG UPDATE REG
4$: BEQ TST442 ;BR IF YES

5$: ERROR 5 ;NEG FAILED TO UPDATE REG

```

```

*****
*TEST 442 NEG DM6 TEST
*****
TST442:

```

```

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #442,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBOFD,R2 ;DEST ADDR = MBOFD
MOV #125252,R4 ;RESULT S / B = 125252
MOV #MBOFD-2,R5 ;[R5] = BASE ADDR
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG 2(R5) ;TEST THE NEG - MODE 6

CMP R4,(R2) ;RESULT = 125252?
BEQ TST443 ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
3$: ERROR 1 ;NEG DELIVERED WRONG RESULT

```

```

*****
*TEST 443 NEG DM7 TEST
*****
TST443:

```

```

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #443,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBOFD,R2 ;DEST ADDR = MBOFD
MOV #125252,R4 ;RESULT S / B = 125252
MOV #ATA,R5 ;[R5] = BASE ADDR
MOV #52526,(R2) ;[DEST] = 52526

```

```

8168 030760 000257          CCC          ;SCOPE SYNC
8169
8170 030762 005475 000010 2$:  NEG      210(R5)      ;TEST THE NEG - MODE 7
8171
8172 030766 020412          CMP      R4,(R2)      ;RESULT = 125252?
8173 030770 001402          BEQ      TST444        ;;BR IF YES
8174
8175 030772 011203          MOV      (R2),R3      ;GET WAS DATA
8176 030774 104001          3$:  ERROR  1          ;NEG DELIVERED WRONG RESULT
8177
8178
8179
8180
8181 030776
8182 030776 000004          ;*****
8183 031000 012700 000444  ;*TEST 444      MOV SM1,DM1 TEST - N:C = 0100
8184 031004 013701 031034  ;*****
8185 031010 005004          †TST444:
8186 031012 005104          SCOPE          ;CALL THE SCOPE LOOP UTILITY
8187 031014 012702 063236  MOV      #444,R0      ;;LOAD R0 WITH TEST NUMBER
8188 031020 012705 063250  MOV      2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
8189 031024 010203          CLR      R4          ;RESULT S / B = 177777
8190 031026 005012          COM      R4
8191 031030 000257          MOV      #MBUFO,R2    ;DEST ADDR = MBUFO
8192 031032 000264          MOV      #DWTA+2,R5  ;SOURCE ADDR = DWTA+2
8193
8194 031034 011513          2$:  MOV      (R5),(R3) ;TEST THE MOV - SM1,DM1
8195
8196 031036 100003          BPL      3$          ;N:C = 1000 ?
8197 031040 001402          BEQ      3$
8198 031042 102401          BVS      3$
8199 031044 103001          BCC      4$
8200
8201 031046 104001          3$:  ERROR  1          ;MOV FAILED TO ALTER CODES PROPERLY
8202
8203 031050 020412          4$:  CMP      R4,(R2)  ;RESULT CORRECT ??
8204 031052 001403          BEQ      TST445        ;;BR IF YES
8205
8206 031054 005003          CLR      R3          ;GET THE WAS DATA
8207 031056 051203          BIS      (R2),R3
8208 031060 104001          5$:  ERROR  1          ;MOV DELIVERED THE WRONG RESULT
8209
8210
8211
8212
8213 031062
8214 031062 000004          ;*****
8215 031064 012700 000445  ;*TEST 445      MOV SM2,DM1 TEST - N:C = 0100
8216 031070 013701 031120  ;*****
8217 031074 005004          †TST445:
8218 031076 005104          SCOPE          ;CALL THE SCOPE LOOP UTILITY
8219 031100 012702 063236  MOV      #445,R0      ;;LOAD R0 WITH TEST NUMBER
8220 031104 012705 063250  MOV      2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
8221 031110 010203          CLR      R4          ;RESULT S / B = 177777
8222 031112 005012          COM      R4
8223 031114 000257          MOV      #MBUFO,R2    ;DEST ADDR = MBUFO
                        MOV      #DWTA+2,R5  ;SOURCE ADDR = DWTA+2
                        MOV      R2,R3      ;BASE DEST ADDR = MBUFO
                        CLR      (R2)      ;MAKE [DEST] = 000000
                        CCC          ;CLEAR FLAGS

```

```

8224 031116 000264          264          ;N:C = 0100
8225
8226 031120 012513          25:  MOV      (R5)+,(R3)      ;TEST THE MOV - SM2,DM1
8227
8228 031122 100003          BPL      35          ;N:C = 1000 ?
8229 031124 001402          BEQ      35
8230 031126 102401          BVS      35
8231 031130 103001          BCC      45
8232
8233 031132 104001          35:  ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8234
8235 031134 020412          45:  CMP      R4,(R2)      ;RESULT CORRECT ??
8236 031136 001403          BEQ      TST446        ;;BR IF YES
8237
8238 031140 005003          CLR      R3          ;GET THE WAS DATA
8239 031142 051203          BIS      (R2),R3
8240 031144 104001          55:  ERROR    1          ;MOV DELIVERED THE WRONG RESULT
8241

```

```

*****
;TEST 446      MOV SM1,DM1 TEST - N:C = 1011
*****

```

```

8242
8243
8244
8245 031146
8246 031146 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
8247 031150 012700 000446  MOV      #446,R0      ;;LOAD R0 WITH TEST NUMBER
8248 031154 013701 031204  MOV      2#25,R1      ;;LOAD R1 WITH TEST INSTRUCTION WORD
8249 031160 005004          CLR      R4          ;RESULT S / B = 000000
8250 031162 012702 063236  MOV      #MBUFD,R2      ;DEST ADDR = MBUFD
8251 031166 012705 063246  MOV      #DWTA,R5      ;SOURCE ADDR = DWTA
8252 031172 010203          MOV      R2,R3      ;BASE DEST ADDR = MBUFD
8253 031174 005012          CLR      (R2)        ;MAKE [DEST] = 177777
8254 031176 005112          COM      (R2)
8255 031200 000257          CCC
8256 031202 000273          273          ;CLEAR FLAGS
8257

```

```

8258 031204 011513          25:  MOV      (R5),(R3)      ;TEST THE MOV - SM1,DM1
8259
8260 031206 100403          BMI      35          ;N:C = 0101 ?
8261 031210 001002          BNE      35
8262 031212 102401          BVS      35
8263 031214 103401          BCS      45
8264

```

```

8265 031216 104001          35:  ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8266
8267 031220 020412          45:  CMP      R4,(R2)      ;RESULT CORRECT ??
8268 031222 001403          BEQ      TST447        ;;BR IF YES
8269

```

```

8270 031224 005003          CLR      R3          ;GET THE WAS DATA
8271 031226 051203          BIS      (R2),R3
8272 031230 104001          55:  ERROR    1          ;MOV DELIVERED THE WRONG RESULT
8273

```

```

*****
;TEST 447      MOV SM2,DM1 TEST - N:C = 1011
*****

```

```

8274
8275
8276
8277 031232
8278 031232 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
8279 031234 012700 000447  MOV      #447,R0      ;;LOAD R0 WITH TEST NUMBER

```



```

8280 031240 013701 031270      MOV      2#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
8281 031244 005004              CLR      R4          ;RESULT S / B = 000000
8282 031246 012702 063236      MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
8283 031252 012705 063246      MOV      #DWTA,R5    ;SOURCE ADDR = DWTA
8284 031256 010203              MOV      R2,R3       ;BASE DEST ADDR = MBUFO
8285 031260 005012              CLR      (R2)        ;MAKE [DEST] = 177777
8286 031262 005112              COM      (R2)
8287 031264 000257              CCC
8288 031266 000273 273          ;CLEAR FLAGS
8289
8290 031270 012513      25:     MOV      (R5)+,(R3) ;TEST THE MOV - SM2,DM1
8291
8292 031272 100403              BMI      3$          ;N:C = 0101 ?
8293 031274 001002              BNE      3$
8294 031276 102401              BVS      3$
8295 031300 103401              BCS      4$
8296
8297 031302 104001      3$:     ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8298
8299 031304 020412      4$:     CMP      R4,(R2)   ;RESULT CORRECT ??
8300 031306 001403              BEQ      TST450      ;;BR IF YES
8301
8302 031310 005003              CLR      R3          ;GET THE WAS DATA
8303 031312 051203              BIS      (R2),R3
8304 031314 104001      5$:     ERROR    1          ;MOV DELIVERED THE WRONG RESULT
8305
8306
8307
8308
8309 031316
8310 031316 000004
8311 031320 012700 000450
8312 031324 013701 031354
8313 031330 005004
8314 031332 005104
8315 031334 012702 063236
8316 031340 012705 063250
8317 031344 010203
8318 031346 005012
8319 031350 000257
8320 031352 000264
8321
8322 031354 011523      25:     MOV      (R5),(R3)+ ;TEST THE MOV - SM1,DM2
8323
8324 031356 100003              BPL      3$          ;N:C = 1000 ?
8325 031360 001402              BEQ      3$
8326 031362 102401              BVS      3$
8327 031364 103001              BCC      4$
8328
8329 031366 104001      3$:     ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8330
8331 031370 022703 063240      4$:     CMP      #MBUFO+2,R3 ;DID MOV INCREMENT DEST REG ?
8332 031374 001401              BEQ      6$          ;BR IF YES
8333
8334 031376 104005      5$:     ERROR    5          ;MOV FAILED TO UPDATE DEST REG
8335

```

```

*****
; *TEST 450      MOV SM1,DM2 TEST - N:C = 0100
*****
TST450:

```

```

SCOPE
MOV      #450,R0      ;CALL THE SCOPE LOOP UTILITY
MOV      2#25,R1     ;LOAD R0 WITH TEST NUMBER
CLR      R4          ;LOAD R1 WITH TEST INSTRUCTION WORD
COM      R4          ;RESULT S / B = 177777
MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
MOV      #DWTA+2,R5  ;SOURCE ADDR = DWTA
MOV      R2,R3       ;BASE DEST ADDR = MBUFO
CLR      (R2)        ;MAKE [DEST] = 000000
CCC
264          ;CLEAR FLAGS
          ;N:C = 0100

```

```

8336 031400 020412
8337 031402 001403
8338
8339 031404 005003
8340 031406 051203
8341 031410 104001
8342
8343
8344
8345
8346 031412
8347 031412 000004
8348 031414 012700 000451
8349 031420 013701 031450
8350 031424 005004
8351 031426 005104
8352 031430 012702 063236
8353 031434 012705 063250
8354 031440 010203
8355 031442 005012
8356 031444 000257
8357 031446 000264
8358
8359 031450 012523
8360
8361 031452 100003
8362 031454 001402
8363 031456 102401
8364 031460 103001
8365
8366 031462 104001
8367
8368 031464 022703 063240
8369 031470 001401
8370
8371 031472 104005
8372
8373 031474 020412
8374 031476 001403
8375
8376 031500 005003
8377 031502 051203
8378 031504 104001
8379
8380
8381
8382
8383 031506
8384 031506 000004
8385 031510 012700 000452
8386 031514 013701 031546
8387 031520 005004
8388 031522 005104
8389 031524 012702 063236
8390 031530 012705 063250
8391 031534 012703 063232

```

```

6S:  CMP      R4,(R2)      ;RESULT CORRECT ??
      BEQ      TST451      ;;BR IF YES

      CLR      R3          ;GET THE WAS DATA
      BIS      (R2),R3
7S:  ERROR    1            ;MOV DELIVERED THE WRONG RESULT

;*****
; *TEST 451      MOV SM2,DM2 TEST - N:C = 0100
;*****
TST451:
      SCOPE
      MOV      #451,R0      ;CALL THE SCOPE LOOP UTILITY
      MOV      2#2$,R1     ;LOAD R0 WITH TEST NUMBER
      CLR      R4          ;LOAD R1 WITH TEST INSTRUCTION WORD
      COM      R4          ;RESULT S / B = 177777
      MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
      MOV      #DWTA+2,R5 ;SOURCE ADDR = DWTA
      MOV      R2,R3      ;BASE DEST ADDR = MBUFO
      CLR      (R2)       ;MAKE [DEST] = 000000
      CCC
      264                ;CLEAR FLAGS
                        ;N:C = 0100

2S:  MOV      (R5)+,(R3)+  ;TEST THE MOV - SM2,DM2

      BPL      3$
      BEQ      3$
      BVS      3$
      BCC      4$
                        ;N:C = 1000 ?

3S:  ERROR    1            ;MOV FAILED TO ALTER CODES PROPERLY

4S:  CMP      #MBUFO+2,R3 ;DID MOV INCREMENT DEST REG ?
      BEQ      6$         ;BR IF YES

5S:  ERROR    5            ;MOV FAILED TO UPDATE DEST REG

6S:  CMP      R4,(R2)      ;RESULT CORRECT ??
      BEQ      TST452      ;;BR IF YES

      CLR      R3          ;GET THE WAS DATA
      BIS      (R2),R3
7S:  ERROR    1            ;MOV DELIVERED THE WRONG RESULT

;*****
; *TEST 452      MOV SM1,DM3 TEST - N:C = 0100
;*****
TST452:
      SCOPE
      MOV      #452,R0      ;CALL THE SCOPE LOOP UTILITY
      MOV      2#2$,R1     ;LOAD R0 WITH TEST NUMBER
      CLR      R4          ;LOAD R1 WITH TEST INSTRUCTION WORD
      COM      R4          ;RESULT S / B = 177777
      MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
      MOV      #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
      MOV      #ATA+10,R3 ;BASE DEST ADDR = ATA+10

```

```

8392 031540 005012          CLR      (R2)          ;MAKE [DEST] = 000000
8393 031542 000257          CCC          ;CLEAR FLAGS
8394 031544 000264          264        ;N:C = 0100
8395
8396 031546 011533          2$:      MOV      (R5),2(R3)+ ;TEST THE MOV - SM1,DM3
8397
8398 031550 100003          BPL      3$          ;N:C = 1000 ?
8399 031552 001402          BEQ      3$
8400 031554 102401          BVS      3$
8401 031556 103001          BCC      4$
8402
8403 031560 104001          3$:      ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8404
8405 031562 022703 063234          4$:      CMP      #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
8406 031566 001401          BEQ      6$          ;BR IF YES
8407
8408 031570 104005          5$:      ERROR    5          ;MOV FAILED TO UPDATE DEST REG
8409
8410 031572 020412          6$:      CMP      R4,(R2)    ;RESULT CORRECT ??
8411 031574 001403          BEQ      TST453      ;;BR IF YES
8412
8413 031576 005003          CLR      R3          ;GET THE WAS DATA
8414 031600 051203          BIS      (R2),R3
8415 031602 104001          7$:      ERROR    1          ;MOV DELIVERED THE WRONG RESULT
8416
8417
8418
8419
8420 031604
8421 031604 000004          ;*****
8422 031606 012700 000453          ;#TEST 453      MOV SM2,DM3 TEST - N:C = 0100
8423 031612 013701 031644          ;*****
8424 031616 005004          TST453:
8425 031620 005104          SCOPE
8426 031622 012702 063236          MOV      #453,R0      ;CALL THE SCOPE LOOP UTILITY
8427 031626 012705 063250          MOV      2#2$,R1     ;;LOAD R0 WITH TEST NUMBER
8428 031632 012703 063232          MOV      #MBUF0,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
8429 031636 005012          CLR      R4          ;RESULT S / B = 177777
8430 031640 000257          COM      R4
8431 031642 000264          MOV      #MBUF0,R2   ;DEST ADDR = MBUF0
8432
8433 031644 012533          2$:      MOV      (R5)+,2(R3)+ ;TEST THE MOV - SM2,DM3
8434
8435 031646 100003          BPL      3$          ;N:C = 1000 ?
8436 031650 001402          BEQ      3$
8437 031652 102401          BVS      3$
8438 031654 103001          BCC      4$
8439
8440 031656 104001          3$:      ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8441
8442 031660 022703 063234          4$:      CMP      #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
8443 031664 001401          BEQ      6$          ;BR IF YES
8444
8445 031666 104005          5$:      ERROR    5          ;MOV FAILED TO UPDATE DEST REG
8446
8447 031670 020412          6$:      CMP      R4,(R2)    ;RESULT CORRECT ??

```

8448 031672 001403
8449
8450 031674 005003
8451 031676 051203
8452 031700 104001
8453
8454
8455
8456
8457 031702
8458 031702 000004
8459 031704 012700 000454
8460 031710 013701 031742
8461 031714 005004
8462 031716 005104
8463 031720 012702 063236
8464 031724 012705 063250
8465 031730 012703 063240
8466 031734 005012
8467 031736 000257
8468 031740 000264
8469
8470 031742 011543
8471
8472 031744 100003
8473 031746 001402
8474 031750 102401
8475 031752 103001
8476
8477 031754 104001
8478
8479 031756 020203
8480 031760 001401
8481
8482 031762 104005
8483
8484 031764 020412
8485 031766 001403
8486
8487 031770 005003
8488 031772 051203
8489 031774 104001
8490
8491
8492
8493
8494 031776
8495 031776 000004
8496 032000 012700 000455
8497 032004 013701 032036
8498 032010 005004
8499 032012 005104
8500 032014 012702 063236
8501 032020 012705 063250
8502 032024 012703 063240
8503 032030 005012

```

BEQ      TST454          ;;BR IF YES
CLR      R3              ;GET THE WAS DATA
BIS      (R2),R3
7$:      ERROR          1      ;MOV DELIVERED THE WRONG RESULT

*****
;TEST 454      MOV SM1,DM4 TEST - N:C = 0100
*****
↑TST454:
SCOPE
MOV      #454,R0          ;CALL THE SCOPE LOOP UTILITY
MOV      @#2$,R1         ;LOAD R0 WITH TEST NUMBER
CLR      R4              ;LOAD R1 WITH TEST INSTRUCTION WORD
COM      R4              ;RESULT S / B = 177777
MOV      #MBUFO,R2       ;DEST ADDR = MBUFO
MOV      #DWTA+2,R5      ;SOURCE ADDR = DWTA+2
MOV      #MBUFO+2,R3     ;BASE DEST ADDR = MBUFO+2
CLR      (R2)            ;MAKE [DEST] = 000000
CCC
264
2$:      MOV      (R5),-(R3) ;TEST THE MOV - SM1,DM4
BPL      3$
BEQ      3$
BVS      3$
BCC      4$
3$:      ERROR          1      ;MOV FAILED TO ALTER CODES PROPERLY
4$:      CMP      R2,R3     ;DID MOV DECREMENT DEST REG ?
BEQ      6$              ;BR IF YES
5$:      ERROR          5      ;MOV FAILED TO UPDATE DEST REG
6$:      CMP      R4,(R2)   ;RESULT CORRECT ??
BEQ      TST455         ;;BR IF YES
CLR      R3              ;GET THE WAS DATA
BIS      (R2),R3
7$:      ERROR          1      ;MOV DELIVERED THE WRONG RESULT

*****
;TEST 455      MOV SM2,DM4 TEST - N:C = 0100
*****
↑TST455:
SCOPE
MOV      #455,R0          ;CALL THE SCOPE LOOP UTILITY
MOV      @#2$,R1         ;LOAD R0 WITH TEST NUMBER
CLR      R4              ;LOAD R1 WITH TEST INSTRUCTION WORD
COM      R4              ;RESULT S / B = 177777
MOV      #MBUFO,R2       ;DEST ADDR = MBUFO
MOV      #DWTA+2,R5      ;SOURCE ADDR = DWTA+2
MOV      #MBUFO+2,R3     ;BASE DEST ADDR = MBUFO+2
CLR      (R2)            ;MAKE [DEST] = 000000

```

```

8504 032032 000257          CCC          ;CLEAR FLAGS
8505 032034 000264          264          ;N:C = 0100
8506
8507 032036 012543          2$: MOV      (R5)+,-(R3)  ;TEST THE MOV - SM2,DM4
8508
8509 032040 100003          BPL      3$          ;N:C = 1000 ?
8510 032042 001402          BEQ      3$
8511 032044 102401          BVS      3$
8512 032046 103001          BCC      4$
8513
8514 032050 104001          3$: ERROR  1          ;MOV FAILED TO ALTER CODES PROPERLY
8515
8516 032052 020203          4$: CMP      R2,R3      ;DID MOV INCREMENT DEST REG ?
8517 032054 001401          BEQ      6$          ;BR IF YES
8518
8519 032056 104005          5$: ERROR  5          ;MOV FAILED TO UPDATE DEST REG
8520
8521 032060 020412          6$: CMP      R4,(R2)    ;RESULT CORRECT ??
8522 032062 001403          BEQ      TST456      ;;BR IF YES
8523
8524 032064 005003          CLR      R3          ;GET THE WAS DATA
8525 032066 051203          BIS      (R2),R3
8526 032070 104001          7$: ERROR  1          ;MOV DELIVERED THE WRONG RESULT
8527

```

```

*****
;TEST 456      MOV SM1,DM5 TEST - N:C = 0100
*****

```

```

8531 032072
8532 032072 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
8533 032074 012700 000456    MOV      #456,R0      ;LOAD R0 WITH TEST NUMBER
8534 032100 013701 032144    MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
8535
8536 032104 032737 001000 063160 .SBTTL USER CONTROLLED BREAKPOINT -- BIT9
8537 032112 001401          BIT      #BIT9,@#BPTLOC ;BREAKPOINT HALT SET ??
8538 032114 000000          BEQ      .+4          ;BR IF NOT
8539 032116 005004          HALT
8540 032120 005104          CLR      R4          ;BREAK - DEPRESS CONTINUE TO RESTART
8541 032122 012702 063236    COM      R4          ;RESULT S / B = 177777
8542 032126 012705 063250    MOV      #MBUF0,R2     ;DEST ADDR = MBUF0
8543 032132 012703 063234    MOV      #DWTA+2,R5    ;SOURCE ADDR = DWTA+2
8544 032136 005012          MOV      #ATA+12,R3    ;BASE DEST ADDR = ATA+12
8545 032140 000257          CLR      (R2)         ;MAKE (DEST) = 000000
8546 032142 000264          CCC          ;CLEAR FLAGS
8547
8548 032144 011553          2$: MOV      (R5),@-(R3) ;TEST THE MOV - SM1,DM5
8549
8550 032146 100003          BPL      3$          ;N:C = 0100 ?
8551 032150 001402          BEQ      3$
8552 032152 102401          BVS      3$
8553 032154 103001          BCC      4$
8554
8555 032156 104001          3$: ERROR  1          ;MOV FAILED TO ALTER CODES PROPERLY
8556
8557 032160 022703 063232    4$: CMP      #ATA+10,R3 ;DID MOV DECREMENT DEST REG ?
8558 032164 001401          BEQ      6$          ;BR IF YES
8559

```

8560 032166 104005
8561
8562 032170 020412
8563 032172 001403
8564
8565 032174 005003
8566 032176 051203
8567 032200 104001
8568
8569
8570
8571
8572 032202
8573 032202 000004
8574 032204 012700 000457
8575 032210 013701 032242
8576 032214 005004
8577 032216 005104
8578 032220 012702 063236
8579 032224 012705 063250
8580 032230 012703 063234
8581 032234 005012
8582 032236 000257
8583 032240 000264
8584
8585 032242 012553
8586
8587 032244 100003
8588 032246 001402
8589 032250 102401
8590 032252 103001
8591
8592 032254 104001
8593
8594 032256 022703 063232
8595 032262 001401
8596
8597 032264 104005
8598
8599 032266 020412
8600 032270 001403
8601
8602 032272 005003
8603 032274 051203
8604 032276 104001
8605
8606
8607
8608
8609 032300
8610 032300 000004
8611 032302 012700 000460
8612 032306 013701 032340
8613 032312 005004
8614 032314 005104
8615 032316 012702 063244

5\$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6\$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST457 ;;BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
7\$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT
;*****
;*TEST 457 MOV SM2,DMS TEST - N:C = 0100
;*****
TST457:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #457,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
CLR (R2) ;MAKE [DEST] = 00000
CCC ;CLEAR FLAGS
264 ;N:C = 1000
2\$: MOV (R5)+,@-(R3) ;TEST THE MOV - SM2,DMS
BPL 3\$;N:C = 1000 ?
BEQ 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
4\$: CMP #ATA+10,R3 ;DID MOV DECREMENT DEST REG ?
BEQ 6\$;;BR IF YES
5\$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6\$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST460 ;;BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
7\$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT
;*****
;*TEST 460 MOV SM1,DM6 TEST - N:C = 0100
;*****
TST460:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #460,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUF0+6,R2 ;DEST ADDR = MBUF0+6

```

8616 032322 012705 063250      MOV      #DWT+2,R5      ;SOURCE ADDR = DWT+2
8617 032326 012703 063236      MOV      #MBUFO,R3     ;BASE DEST ADDR = MBUFO
8618 032332 005012              CLR      (R2)          ;MAKE [DEST] = 000000
8619 032334 000257              CCC                      ;CLEAR FLAGS
8620 032336 000264              264                    ;N:C = 0100
8621
8622 032340 011563 000006      2$:  MOV      (R5),6(R3) ;TEST THE MOV - SM1,DM6
8623
8624 032344 100003              BPL      3$            ;N:C = 1000 ?
8625 032346 001402              BEQ      3$
8626 032350 102401              BVS      3$
8627 032352 103001              BCC      4$
8628
8629 032354 104001      3$:  ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8630
8631 032356 020412      4$:  CMP      R4,(R2)    ;RESULT CORRECT ??
8632 032360 001403              BEQ      TST461        ;;BR IF YES
8633
8634 032362 005003              CLR      R3            ;GET THE WAS DATA
8635 032364 051203              BIS      (R2),R3
8636 032366 104001      5$:  ERROR    1          ;MOV DELIVERED THE WRONG RESULT
8637

```

```

;*****
;TEST 461      MOV SM2,DM6 TEST - N:C = 0100
;*****
TST461:

```

```

8641 032370              SCOPE                  ;CALL THE SCOPE LOOP UTILITY
8642 032370 000004      MOV      #461,R0      ;LOAD R0 WITH TEST NUMBER
8643 032372 012700 000461      MOV      #2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
8644 032376 013701 032430      CLR      R4            ;RESULT S / B = 177777
8645 032402 005004              COM      R4
8646 032404 005104              MOV      #MBUFO+6,R2  ;DEST ADDR = MBUFO+6
8647 032406 012702 063244      MOV      #DWT+2,R5    ;SOURCE ADDR = DWT+2
8648 032412 012705 063250      MOV      #MBUFO,R3     ;BASE DEST ADDR = MBUFO
8649 032416 012703 063236      CLR      (R2)          ;MAKE [DEST] = 000000
8650 032422 005012              CCC                      ;CLEAR FLAGS
8651 032424 000257              264                    ;N:C = 0100
8652 032426 000264
8653
8654 032430 012563 000006      2$:  MOV      (R5)+,6(R3) ;TEST THE MOV - SM2,DM6
8655
8656 032434 100003              BPL      3$            ;N:C = 1000 ?
8657 032436 001402              BEQ      3$
8658 032440 102401              BVS      3$
8659 032442 103001              BCC      4$
8660
8661 032444 104001      3$:  ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
8662
8663 032446 020412      4$:  CMP      R4,(R2)    ;RESULT CORRECT ??
8664 032450 001403              BEQ      TST462        ;;BR IF YES
8665
8666 032452 005003              CLR      R3            ;GET THE WAS DATA
8667 032454 051203              BIS      (R2),R3
8668 032456 104001      5$:  ERROR    1          ;MOV DELIVERED THE WRONG RESULT
8669

```

```

;*****
;TEST 462      MOV SM1,DM7 TEST - N:C = 0100
;*****

```

```

8670
8671

```

```

8672
8673 032460
8674 032460 000004
8675 032462 012700 000462
8676 032466 013701 032520
8677 032472 005004
8678 032474 005104
8679 032476 012702 063236
8680 032502 012705 063250
8681 032506 012703 063222
8682 032512 005012
8683 032514 000257
8684 032516 000264
8685
8686 032520 011573 000010
8687
8688 032524 100003
8689 032526 001402
8690 032530 102401
8691 032532 103001
8692
8693 032534 104001
8694
8695 032536 020412
8696 032540 001403
8697
8698 032542 005003
8699 032544 051203
8700 032546 104001
8701
8702
8703
8704
8705 032550
8706 032550 000004
8707 032552 012700 000463
8708 032556 013701 032610
8709 032562 005004
8710 032564 005104
8711 032566 012702 063236
8712 032572 012705 063250
8713 032576 012703 063222
8714 032602 005012
8715 032604 000257
8716 032606 000264
8717
8718 032610 011573 000010
8719
8720 032614 100003
8721 032616 001402
8722 032620 102401
8723 032622 103001
8724
8725 032624 104001
8726
8727 032626 020412
    
```

```

*****
TST462:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #462,R0                      ;LOAD R0 WITH TEST NUMBER
MOV      @#25,R1                       ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R4                            ;RESULT S / B = 177777
COM      R4
MOV      #MBUFD,R2                     ;DEST ADDR = MBUFD
MOV      #DWTA+2,R5                    ;SOURCE ADDR = DWTA+2
MOV      #ATA,R3                       ;BASE DEST ADDR = ATA
CLR      (R2)                          ;MAKE [DEST] = 000000
CCC
264
25:   MOV      (R5),@10(R3)             ;TEST THE MOV - SM1,DM7
                                           ;N:C = 1000 ?
                                           BPL      3$
                                           BEQ      3$
                                           BVS      3$
                                           BCC      4$
3$:   ERROR    1                       ;MOV FAILED TO ALTER CODES PROPERLY
4$:   CMP      R4,(R2)                 ;RESULT CORRECT ??
                                           BEQ      TST463
                                           ;;BR IF YES
                                           CLR      R3
                                           BIS      (R2),R3                       ;GET THE WAS DATA
5$:   ERROR    1                       ;MOV DELIVERED THE WRONG RESULT
*****
;TEST 463      MOV SM2,DM7 TEST - N:C = 0100
*****
TST463:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #463,R0                      ;LOAD R0 WITH TEST NUMBER
MOV      @#25,R1                       ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R4                            ;RESULT S / B = 177777
COM      R4
MOV      #MBUFD,R2                     ;DEST ADDR = MBUFD
MOV      #DWTA+2,R5                    ;SOURCE ADDR = DWTA+2
MOV      #ATA,R3                       ;BASE DEST ADDR = ATA
CLR      (R2)                          ;MAKE [DEST] = 000000
CCC
264
25:   MOV      (R5),@10(R3)             ;TEST THE MOV - SM2,DM7
                                           ;N:C = 1000 ?
                                           BPL      3$
                                           BEQ      3$
                                           BVS      3$
                                           BCC      4$
3$:   ERROR    1                       ;MOV FAILED TO ALTER CODES PROPERLY
4$:   CMP      R4,(R2)                 ;RESULT CORRECT ??
    
```



```

8728 032630 001403
8729
8730 032632 005003
8731 032634 051203
8732 032636 104001
8733
8734
8735
8736
8737 032640
8738 032640 000004
8739 032642 012700 000464
8740 032646 013701 032666
8741 032652 012702 063236
8742 032656 010004
8743 032660 010205
8744 032662 005012
8745 032664 000257
8746
8747 032666 010015
8748
8749 032670 020412
8750 032672 001402
8751
8752 032674 011203
8753 032676 104001
8754
8755
8756
8757
8758 032700
8759 032700 000004
8760 032702 012700 000465
8761 032706 013701 032726
8762 032712 012702 063236
8763 032716 010004
8764 032720 010205
8765 032722 005012
8766 032724 000257
8767
8768 032726 010025
8769
8770 032730 020412
8771 032732 001402
8772
8773 032734 011203
8774 032736 104001
8775
8776
8777
8778
8779 032740
8780 032740 000004
8781 032742 012700 000466
8782 032746 013701 032770
8783 032752 012702 063236

```

```

      BEQ      TST464      ;;BR IF YES
      CLR      R3          ;GET THE WAS DATA
      BIS      (R2),R3
SS:    ERROR    1          ;MOV DELIVERED THE WRONG RESULT
;*****
; *TEST 464      MOV SMO,DM1 TEST
;*****
TST464:
      SCOPE
      MOV      #464,R0      ;CALL THE SCOPE LOOP UTILITY
      MOV      @#25,R1      ;LOAD RO WITH TEST NUMBER
      MOV      #MBUFD,R2    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      R0,R4        ;DEST ADDR = MBUFD
      MOV      R2,R5        ;RESULT S / B = TEST NUMBER
      CLR      (R2)         ;R5 GETS DEST ADDR
      CCC
      ;[DEST] = 000000
      ;SCOPE SYNC
2S:    MOV      R0,(R5)     ;TEST THE MOV
      CMP      R4,(R2)     ;RESULT CORRECT ?
      BEQ      TST465      ;;BR IF YES
3S:    MOV      (R2),R3     ;GET THE WAS DATA
      ERROR    1          ;MOV DELIVERED THE WRONG RESULT
;*****
; *TEST 465      MOV SMO,DM2 TEST
;*****
TST465:
      SCOPE
      MOV      #465,R0      ;CALL THE SCOPE LOOP UTILITY
      MOV      @#25,R1      ;LOAD RO WITH TEST NUMBER
      MOV      #MBUFD,R2    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      R0,R4        ;DEST ADDR = MBUFD
      MOV      R2,R5        ;RESULT S / B = TEST NUMBER
      CLR      (R2)         ;R5 GETS DEST ADDR
      CCC
      ;[DEST] = 000000
      ;SCOPE SYNC
2S:    MOV      R0,(R5)+   ;TEST THE MOV
      CMP      R4,(R2)     ;RESULT CORRECT ?
      BEQ      TST466      ;;BR IF YES
3S:    MOV      (R2),R3     ;GET THE WAS DATA
      ERROR    1          ;MOV DELIVERED THE WRONG RESULT
;*****
; *TEST 466      MOV SMO,DM3 TEST
;*****
TST466:
      SCOPE
      MOV      #466,R0      ;CALL THE SCOPE LOOP UTILITY
      MOV      @#25,R1      ;LOAD RO WITH TEST NUMBER
      MOV      #MBUFD,R2    ;LOAD R1 WITH TEST INSTRUCTION WORD
      ;DEST ADDR = MBUFD

```

8784 032756 010004
8785 032760 012705 063232
8786 032764 005012
8787 032766 000257
8788
8789 032770 010035
8790
8791 032772 020412
8792 032774 001402
8793
8794 032776 011203
8795 033000 104001
8796
8797
8798
8799
8800 033002
8801 033002 000004
8802 033004 012700 000467
8803 033010 013701 033032
8804 033014 012702 063236
8805 033020 010004
8806 033022 012705 063240
8807 033026 005012
8808 033030 000257
8809
8810 033032 010045
8811
8812 033034 020412
8813 033036 001402
8814
8815 033040 011203
8816 033042 104001
8817
8818
8819
8820
8821 033044
8822 033044 000004
8823 033046 012700 000470
8824 033052 013701 033074
8825 033056 012702 063236
8826 033062 010004
8827 033064 012705 063234
8828 033070 005012
8829 033072 000257
8830
8831 033074 010055
8832
8833 033076 020412
8834 033100 001402
8835
8836 033102 011203
8837 033104 104001
8838
8839

```

MOV RO,R4 ;RESULT S / B = TEST NUMBER
MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV RO,2(R5)+ ;TEST THE MOV

CMP R4,(R2) ;CORRECT RESULT
BEQ TST467 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

*****
;TEST 467 MOV SMO,DM4 TEST
*****
TST467:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #467,R0 ;LOAD RO WITH TEST NUMBER
MOV 2#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV RO,R4 ;RESULT S / B = TEST NUMBER
MOV #MBUFO+2,R5 ;R5 CONTAINS BASE DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV RO,-(R5) ;TEST THE MOV

CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST470 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

*****
;TEST 470 MOV SMO,DMS TEST
*****
TST470:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #470,R0 ;LOAD RO WITH TEST NUMBER
MOV 2#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV RO,R4 ;RESULT S / B = TEST NUMBER
MOV #ATA+12,R5 ;R5 CONTAINS BASE DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV RO,2-(R5) ;TEST THE MOV

CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST471 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

```

8840
8841
8842
8843 033106
8844 033106 000004
8845 033110 012700 000471
8846 033114 013701 033136
8847 033120 012702 063242
8848 033124 010004
8849 033126 012705 063236
8850 033132 005012
8851 033134 000257
8852
8853 033136 010065 000004
8854
8855 033142 020412
8856 033144 001402
8857
8858 033146 011203
8859 033150 104001
8860
8861
8862
8863
8864 033152
8865 033152 000004
8866 033154 012700 000472
8867 033160 013701 033202
8868 033164 012704 177652
8869 033170 012705 000252
8870 033174 005003
8871 033176 000257
8872 033200 000266
8873
8874 033202 110503
8875
8876 033204 100003
8877 033206 001402
8878 033210 102401
8879 033212 103001
8880
8881 033214 104002
8882
8883 033216 020403
8884 033220 001401
8885
8886 033222 104002
8887
8888
8889
8890
8891 033224
8892 033224 000004
8893 033226 012700 000473
8894 033232 013701 033254
8895 033236 005004

```
*****
*TEST 471      MOV SMO,DM6 TEST
*****
†ST471:
SCOPE
MOV #471,R0      ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1      ;LOAD R0 WITH TEST NUMBER
MOV #MBUF1,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV R0,R4       ;DEST ADDR = MBUF1
MOV #MBUFO,R5  ;RESULT S / B = TEST NUMBER
CLR (R2)        ;BASE DEST ADDR = MBUFO
CCC             ;[DEST] = 000000
               ;SCOPE SYNC
2$: MOV R0,4(R5) ;TEST THE MOV
    CMP R4,(R2) ;RESULT CORRECT ?
    BEQ TST472  ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1     ;MOV DELIVERED THE WRONG RESULT
*****
*TEST 472      MOV B TEST - SMO,DMO - EXTEND 1'S
*****
†ST472:
SCOPE
MOV #472,R0      ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1      ;LOAD R0 WITH TEST NUMBER
MOV #177652,R4  ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #252,R5     ;RESULT S / B = 177652
CLR R3          ;SOURCE OP = 252
CCC             ;[DEST] = 000000
266            ;CLEAR FLAGS
               ;N:C = 0110
2$: MOV B R5,R3 ;TEST THE MOV B
    BPL 3$      ;N:C = 1000 ?
    BEQ 3$
    BVS 3$
    BCC 4$
3$: ERROR 2     ;MOV B FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3   ;RESULT CORRECT ?
    BEQ TST473  ;;BR IF YES
5$: ERROR 2     ;MOV B DELIVERED THE WRONG RESULT
*****
*TEST 473      MOV B TEST - SMO,DMO - EXTEND 0'S
*****
†ST473:
SCOPE
MOV #473,R0      ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1      ;LOAD R0 WITH TEST NUMBER
CLR R4           ;LOAD R1 WITH TEST INSTRUCTION WORD
               ;RESULT S / B = 000000
```

```

8896 033240 012705 177400      MOV      #177400,R5      ;SOURCE OP = 177400
8897 033244 005003              CLR      R3              ;[DEST] = 177777
8898 033246 005103              COM      R3
8899 033250 000257              CCC                      ;CLEAR FLAGS
8900 033252 000271              271                      ;N:C = 1001
8901
8902 033254 110503      2$:      MOV8      R5,R3      ;TEST THE MOV8
8903
8904 033256 100403              BMI      3$              ;N:C = 0101 ?
8905 033260 001002              BNE      3$
8906 033262 102401              BVS      3$
8907 033264 103401              BCS      4$
8908
8909 033266 104002      3$:      ERROR      2              ;MOV8 FAILED TO ALTER CODES PROPERLY
8910
8911 033270 020403      4$:      CMP      R4,R3              ;RESULT CORRECT ?
8912 033272 001401              BEQ      TST474          ;;BR IF YES
8913
8914 033274 104002      5$:      ERROR      2              ;MOV8 DELIVERED THE WRONG RESULT
8915
8916
8917
8918
8919 033276
8920 033276 000004              ;:*****
8921 033300 012700 000474      ;*TEST 474      MOV8 TEST - SM1,DMO - SOURCE ADDR EVEN
8922 033304 013701 033324      ;:*****
8923 033310 005004      TST474:
8924 033312 012705 064554      SCOPE
8925 033316 005003              MOV      #474,R0          ;CALL THE SCOPE LOOP UTILITY
8926 033320 005103              MOV      @#2$,R1         ;;LOAD R0 WITH TEST NUMBER
8927 033322 000257              CLR      R4              ;LOAD R1 WITH TEST INSTRUCTION WORD
8928
8929 033324 111503      2$:      MOV8      (R5),R3        ;RESULT S / B = 000000
8930
8931 033326 020403              MOV      #DBTA,R5        ;SOURCE ADDR = DBTA
8932 033330 001401              CLR      R3              ;[DEST] = 177777
8933
8934 033332 104002      3$:      ERROR      2              ;SCOPE SYNC
8935
8936
8937
8938
8939 033334
8940 033334 000004              ;:*****
8941 033336 012700 000475      ;*TEST 475      MOV8 TEST - SM1,DMO - SOURCE ADDR ODD
8942 033342 013701 033364      ;:*****
8943 033346 012704 000125      TST475:
8944 033352 012705 064557      SCOPE
8945 033356 012703 177400      MOV      #475,R0          ;CALL THE SCOPE LOOP UTILITY
8946 033362 000257              MOV      @#2$,R1         ;;LOAD R0 WITH TEST NUMBER
8947
8948 033364 111503      2$:      MOV8      (R5),R3        ;LOAD R1 WITH TEST INSTRUCTION WORD
8949
8950 033366 020403              MOV      #125,R4         ;RESULT S / B = 125
8951 033370 001401              MOV      #DBTA+3,R5      ;SOURCE ADDR = DBTA+3
              MOV      #177400,R3 ;[DEST] = 177400
              CCC          ;SCOPE SYNC

```

```

8952
8953 033372 104002 3$: ERROR 2 ;MOV8 DELIVERED THE WRONG RESULT
8954
8955 ;:*****
8956 ;*TEST 476 MOV8 TEST - SM2,DMO - SOURCE ADDR ODD
8957 ;:*****
8958 †ST476:
8959 033374 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
8960 033376 012700 000476 MOV #476,R0 ;LOAD R0 WITH TEST NUMBER
8961 033402 013701 033422 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
8962 033406 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
8963 033412 012705 064555 MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
8964 033416 005003 CLR R3 ;[DEST] = 000000
8965 033420 000257 CCC ;SCOPE SYNC
8966
8967 033422 112503 2$: MOV8 (R5)+,R3 ;TEST THE MOV8
8968
8969 033424 020403 CMP R4,R3 ;RESULT CORRECT ?
8970 033426 001401 BEQ 4$ ;BR IF YES
8971
8972 033430 104002 3$: ERROR 2 ;MOV8 DELIVERED THE WRONG RESULT
8973
8974 033432 022705 064556 4$: CMP #DBTA+2,R5 ;DID MOV8 INCREMENT SRC REG ?
8975 033436 001401 BEQ TST477 ;;BR IF YES
8976
8977 033440 104005 5$: ERROR 5 ;MOV8 FAILED TO UPDATE SRC REG
8978
8979 ;:*****
8980 ;*TEST 477 MOV8 TEST - SM2,DMO - SOURCE ADDR EVEN
8981 ;:*****
8982 †ST477:
8983 033442 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
8984 033444 012700 000477 MOV #477,R0 ;LOAD R0 WITH TEST NUMBER
8985 033450 013701 033470 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
8986 033454 005004 CLR R4 ;RESULT S / B = 000000
8987 033456 012705 064554 MOV #DBTA,R5 ;SOURCE ADDR = DBTA
8988 033462 012703 177400 MOV #177400,R3 ;[DEST] = 177400
8989 033466 000257 CCC ;SCOPE SYNC
8990
8991 033470 112503 2$: MOV8 (R5)+,R3 ;TEST THE MOV8
8992
8993 033472 020403 CMP R4,R3 ;RESULT CORRECT ?
8994 033474 001401 BEQ 4$ ;BR IF YES
8995
8996 033476 104002 3$: ERROR 2 ;MOV8 DELIVERED THE WRONG RESULT
8997
8998 033500 022705 064555 4$: CMP #DBTA+1,R5 ;DID MOV8 INCREMENT SRC REG ?
8999 033504 001401 BEQ TST500 ;;BR IF YES
9000
9001 033506 104005 5$: ERROR 5 ;MOV8 FAILED TO UPDATE SOURCE REG
9002
9003 ;:*****
9004 ;*TEST 500 MOV8 TEST - SM1,DM1 - SRC ADR ODD / DST ADR EVEN
9005 ;:*****
9006 †ST500:
9007 033510 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

9008 033512 012700 000500      MOV      #500,R0          ;;LOAD R0 WITH TEST NUMBER
9009 033516 013701 033542      MOV      @#25,R1         ;;LOAD R1 WITH TEST INSTRUCTION WORD
9010 033522 012702 063236      MOV      #MBUFO,R2      ;;DEST ADDR = MBUFO
9011 033526 012704 000377      MOV      #377,R4        ;;RESULT S / B = 377
9012 033532 012705 064555      MOV      #DBTA+1,R5     ;;SRC ADDR = DBTA +1
9013 033536 005012              CLR      (R2)           ;;[DEST] = 000000
9014 033540 000257              CCC                    ;;CLEAR FLAGS - SCOPE SYNC
9015
9016 033542 111512      2$:      MOV8      (R5),(R2)      ;TEST THE MOV8
9017
9018 033544 020412      CMP      R4,(R2)        ;CORRECT RESULT ?
9019 033546 001402      BEQ      TST501         ;;BR IF YES
9020
9021 033550 011203      MOV      (R2),R3        ;GET THE WAS DATA
9022 033552 104001      3$:      ERROR      1          ;MOV8 DELIVERED WRONG RESULT
9023
9024      ;*****
9025      ;*TEST 501      MOV8 TEST - SM1,DM2 - SRC ADR ODD / DST ADR EVEN
9026      ;*****
9027      †TST501:
9028 033554 000004              SCOPE                    ;CALL THE SCOPE LOOP UTILITY
9029 033556 012700 000501      MOV      #501,R0        ;;LOAD R0 WITH TEST NUMBER
9030 033562 013701 033610      MOV      @#25,R1         ;;LOAD R1 WITH TEST INSTRUCTION WORD
9031 033566 012702 063236      MOV      #MBUFO,R2      ;;DEST ADDR = MBUFO
9032 033572 012704 000377      MOV      #377,R4        ;;RESULT S / B = 377
9033 033576 012705 064555      MOV      #DBTA+1,R5     ;;SRC ADDR = DBTA +1
9034 033602 005012              CLR      (R2)           ;;[DEST] = 000000
9035 033604 010203      MOV      R2,R3         ;;[R3] = DEST ADDR
9036 033606 000257              CCC                    ;;CLEAR FLAGS - SCOPE SYNC
9037
9038 033610 111523      2$:      MOV8      (R5),(R3)+      ;TEST THE MOV8
9039
9040 033612 020412      CMP      R4,(R2)        ;CORRECT RESULT ?
9041 033614 001402      BEQ      4$             ;BR IF YES
9042
9043 033616 011203      MOV      (R2),R3        ;GET THE WAS DATA
9044 033620 104001      3$:      ERROR      1          ;MOV8 DELIVERED WRONG RESULT
9045
9046 033622 022703 063237      4$:      CMP      #MBUFO+1,R3  ;DID MOV8 INCREMENT THE DEST REG ?
9047 033626 001401      BEQ      TST502         ;;BR IF YES
9048
9049 033630 104005      5$:      ERROR      5          ;MOV8 FAILED TO UPDATE DEST REG
9050
9051      ;*****
9052      ;*TEST 502      MOV8 TEST - SM1,DM3 - SRC ADR ODD / DST ADR EVEN
9053      ;*****
9054      †TST502:
9055 033632 000004              SCOPE                    ;CALL THE SCOPE LOOP UTILITY
9056 033634 012700 000502      MOV      #502,R0        ;;LOAD R0 WITH TEST NUMBER
9057 033640 013701 033670      MOV      @#25,R1         ;;LOAD R1 WITH TEST INSTRUCTION WORD
9058 033644 012702 063236      MOV      #MBUFO,R2      ;;DEST ADDR = MBUFO
9059 033650 012704 000377      MOV      #377,R4        ;;RESULT S / B = 377
9060 033654 012705 064555      MOV      #DBTA+1,R5     ;;SRC ADDR = DBTA +1
9061 033660 005012              CLR      (R2)           ;;[DEST] = 000000
9062 033662 012703 063232      MOV      #ATA+10,R3     ;;BASE DEST ADDR = ATA +10
9063 033666 000257              CCC                    ;;CLEAR FLAGS - SCOPE SYNC
    
```

H15

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.A.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 166
MOV B TEST - SM1,DM3 - SRC ADR ODD / DST ADR EVEN

```
9064
9065 033670 111533      2$:  MOV B (R5),2(R3)+ ;TEST THE MOV B
9066
9067 033672 022703 063234  CMP #ATA+12,R3 ;DID DEST REG GET INCREMENTED ?
9068 033676 001401      BEQ 4$ ;BR IF YES
9069
9070 033700 104005      3$:  ERROR 5 ;MOV B FAILED TO UPDATE DEST REG
9071
9072 033702 020412      4$:  CMP R4,(R2) ;CORRECT RESULT ?
9073 033704 001402      BEQ TST503 ;;BR IF YES
9074
9075 033706 011203      MOV (R2),R3 ;GET THE WAS DATA
9076 033710 104001      5$:  ERROR 1 ;MOV B DELIVERED WRONG RESULT
9077
9078 ;:*****
9079 ;*TEST 503 MOV B TEST - SM1,DM4 - SRC ADR ODD / DST ADR EVEN
9080 ;:*****
9081 033712      †TST503:
9082 033712 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
9083 033714 012700 000503  MOV #503,R0 ;;LOAD R0 WITH TEST NUMBER
9084 033720 013701 033750  MOV 2#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
9085 033724 012702 063236  MOV #MBUFD,R2 ;DEST ADDR = MBUFD
9086 033730 012704 000377  MOV #377,R4 ;RESULT S / B = 377
9087 033734 012705 064555  MOV #DBTA+1,R5 ;SRC ADDR = DBTA +1
9088 033740 005012      CLR (R2) ;[DEST] = 000000
9089 033742 012703 063237  MOV #MBUFD+1,R3 ;INITIAL DEST ADDR = MBUFD+1
9090 033746 000257      CCC ;CLEAR FLAGS - SCOPE SYNC
9091
9092 033750 111543      2$:  MOV B (R5),-(R3) ;TEST THE MOV B
9093
9094 033752 020302      CMP R3,R2 ;DID MOV B DECREMENT DEST REG ?
9095 033754 001401      BEQ 4$ ;BR IF YES
9096
9097 033756 104005      3$:  ERROR 5 ;MOV B FAILED TO UPDATE DEST REG
9098
9099 033760 020412      4$:  CMP R4,(R2) ;CORRECT RESULT ?
9100 033762 001402      BEQ TST504 ;;BR IF YES
9101
9102 033764 011203      MOV (R2),R3 ;GET THE WAS DATA
9103 033766 104001      5$:  ERROR 1 ;MOV B DELIVERED WRONG RESULT
9104
9105 ;:*****
9106 ;*TEST 504 MOV B TEST - SM1,DM5 - SRC ADR ODD / DST ADR EVEN
9107 ;:*****
9108 033770      †TST504:
9109 033770 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
9110 033772 012700 000504  MOV #504,R0 ;;LOAD R0 WITH TEST NUMBER
9111 033776 013701 034026  MOV 2#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
9112 034002 012702 063236  MOV #MBUFD,R2 ;DEST ADDR = MBUFD
9113 034006 012704 000377  MOV #377,R4 ;RESULT S / B = 377
9114 034012 012705 064555  MOV #DBTA+1,R5 ;SRC ADDR = DBTA +1
9115 034016 005012      CLR (R2) ;[DEST] = 000000
9116 034020 012703 063234  MOV #ATA+12,R3 ;INITIAL DEST ADDR = ATA +12
9117 034024 000257      CCC ;CLEAR FLAGS - SCOPE SYNC
9118
9119 034026 111553      2$:  MOV B (R5),2-(R3) ;TEST THE MOV B
```

```

9120
9121 034030 022703 063232      CMP      #ATA+10,R3      ;DID MOV8 DECREMENT DEST REG ?
9122 034034 001401              BEQ      4$              ;BR IF YES
9123
9124 034036 104005      3$:      ERROR      5              ;MOV8 FAILED TO UPDATE DEST REG
9125
9126 034040 020412      4$:      CMP      R4,(R2)      ;CORRECT RESULT ?
9127 034042 001402              BEQ      T$T505          ;;BR IF YES
9128
9129 034044 011203      5$:      MOV      (R2),R3      ;GET THE WAS DATA
9130 034046 104001              ERROR      1              ;MOV8 DELIVERED WRONG RESULT
9131
9132      ;*****
9133      ;*TEST 505      MOV8 TEST - SM1,DM6 - SRC ADR ODD / DST ADR EVEN
9134      ;*****
9135      T$T505:
9136 034050 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
9137 034052 012700 000505      MOV      #505,R0      ;;LOAD R0 WITH TEST NUMBER
9138 034056 013701 034106      MOV      @#2$,R1      ;;LOAD R1 WITH TEST INSTRUCTION WORD
9139 034062 012702 063236      MOV      #M8UFO,R2      ;;DEST ADDR = M8UFO
9140 034066 012704 000377      MOV      #377,R4      ;;RESULT S / B = 377
9141 034072 012705 064555      MOV      #DBTA+1,R5      ;;SRC ADDR = DBTA +1
9142 034076 005012              CLR      (R2)          ;;[DEST] = 000000
9143 034100 012703 063244      MOV      #M8UFO+6,R3      ;;BASE DEST ADDR = M8UFO+6
9144 034104 000257              CCC              ;CLEAR FLAGS - SCOPE SYNC
9145
9146 034106 111563 177772      2$:      MOV8      (R5),-6(R3)      ;TEST THE MOV8
9147
9148 034112 020412      CMP      R4,(R2)      ;CORRECT RESULT ?
9149 034114 001402              BEQ      T$T506          ;;BR IF YES
9150
9151 034116 011203      3$:      MOV      (R2),R3      ;GET THE WAS DATA
9152 034120 104001              ERROR      1              ;MOV8 DELIVERED WRONG RESULT
9153
9154      ;*****
9155      ;*TEST 506      MOV8 TEST - SM1,DM7 - SRC ADR ODD / DST ADR EVEN
9156      ;*****
9157      T$T506:
9158 034122 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
9159 034124 012700 000506      MOV      #506,R0      ;;LOAD R0 WITH TEST NUMBER
9160 034130 013701 034160      MOV      @#2$,R1      ;;LOAD R1 WITH TEST INSTRUCTION WORD
9161 034134 012702 063236      MOV      #M8UFO,R2      ;;DEST ADDR = M8UFO
9162 034140 012704 000377      MOV      #377,R4      ;;RESULT S / B = 377
9163 034144 012705 064555      MOV      #DBTA+1,R5      ;;SRC ADDR = DBTA +1
9164 034150 005012              CLR      (R2)          ;;[DEST] = 000000
9165 034152 012703 063222      MOV      #ATA,R3      ;;BASE DEST ADDR = ATA
9166 034156 000257              CCC              ;CLEAR FLAGS - SCOPE SYNC
9167
9168 034160 111573 000010      2$:      MOV8      (R5),@10(R3)      ;TEST THE MOV8
9169
9170 034164 020412      CMP      R4,(R2)      ;CORRECT RESULT ?
9171 034166 001402              BEQ      T$T507          ;;BR IF YES
9172
9173 034170 011203      3$:      MOV      (R2),R3      ;GET THE WAS DATA
9174 034172 104001              ERROR      1              ;MOV8 DELIVERED WRONG RESULT
9175

```



```

9176
9177
9178
9179 034174
9180 034174 000004
9181 034176 012700 000507
9182 034202 013701 034230
9183 034206 012702 063236
9184 034212 012704 000377
9185 034216 012703 177777
9186 034222 010205
9187 034224 005012
9188 034226 000257
9189
9190 034230 110315
9191
9192 034232 020412
9193 034234 001402
9194
9195 034236 011203
9196 034240 104001
9197
9198
9199
9200
9201 034242
9202 034242 000004
9203 034244 012700 000510
9204 034250 013701 034276
9205 034254 012702 063236
9206 034260 012704 000377
9207 034264 012703 177777
9208 034270 010205
9209 034272 005012
9210 034274 000257
9211
9212 034276 110325
9213
9214 034300 020412
9215 034302 001402
9216
9217 034304 011203
9218 034306 104001
9219
9220
9221
9222
9223 034310
9224 034310 000004
9225 034312 012700 000511
9226 034316 013701 034346
9227 034322 012702 063236
9228 034326 012704 000377
9229 034332 012703 177777
9230 034336 012705 063232
9231 034342 005012

```

```

*****
;*TEST 507 MOV B SMO,DM1 TEST
*****
†T507:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #507,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV R2,R5 ;R5 CONTAINS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV B R3,(R5) ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ T510 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT

*****
;*TEST 510 MOV B SMO,DM2 TEST
*****
†T510:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #510,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV R2,R5 ;R5 CONTAINS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV B R3,(R5)+ ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ T511 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT

*****
;*TEST 511 MOV B SMO,DM3 TEST
*****
†T511:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #511,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;SOURCE OP IN R3
MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10
CLR (R2) ;[DEST] = 000000

```

K15

9232 034344 000257
9233
9234 034346 110335
9235
9236 034350 020412
9237 034352 001402
9238
9239 034354 011203
9240 034356 104001
9241
9242
9243
9244
9245 034360
9246 034360 000004
9247 034362 012700 000512
9248 034366 013701 034416
9249 034372 012702 063236
9250 034376 012704 177400
9251 034402 012703 177777
9252 034406 012705 063240
9253 034412 005012
9254 034414 000257
9255
9256 034416 110345
9257
9258 034420 020412
9259 034422 001402
9260
9261 034424 011203
9262 034426 104001
9263
9264
9265
9266
9267 034430
9268 034430 000004
9269 034432 012700 000513
9270 034436 013701 034466
9271 034442 012702 063236
9272 034446 012704 000377
9273 034452 012703 177777
9274 034456 012705 063240
9275 034462 005012
9276 034464 000257
9277
9278 034466 110365 177776
9279
9280 034472 020412
9281 034474 001402
9282
9283 034476 011203
9284 034500 104001
9285
9286
9287

```
CCC ;SCOPE SYNC
2$: MOV B R3,2(R5)+ ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST512 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
*****
;*TEST 512 MOV B SMO,DM4 TEST
*****
TST512:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #512,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #177400,R4 ;RESULT S / B = 177400
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV #MBUFD+2,R5 ;BASE DEST ADDR = MBUFD+2
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: MOV B R3,-(R5) ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST513 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
*****
;*TEST 513 MOV B SMO,DM6 TEST
*****
TST513:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #513,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV #MBUFD+2,R5 ;BASE DEST ADDR = MBUFD+2
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: MOV B R3,-2(R5) ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST514 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
*****
;*TEST 514 BIS TEST - SMO,DM0 - N:C = 0111
```

```

9288
9289 034502
9290 034502 000004
9291 034504 012700 000514
9292 034510 013701 034534
9293 034514 012704 177777
9294 034520 012705 125252
9295 034524 012703 052525
9296 034530 000257
9297 034532 000267
9298
9299 034534 050503      2$:  BIS      R5,R3      ;TEST THE BIS
9300
9301 034536 100003      BPL      3$      ;N:C = 1001 ?
9302 034540 001402      BEQ      3$
9303 034542 102401      BVS      3$
9304 034544 103401      BCS      4$
9305
9306 034546 104002      3$:  ERROR    2      ;BIS FAILED TO ALTER CODES PROPERLY
9307
9308 034550 020403      4$:  CMP      R4,R3      ;CORRECT RESULT ?
9309 034552 001401      BEQ      TS1515      ;;BR IF YES
9310
9311 034554 104002      5$:  ERROR    2      ;BIS DELIVERED THE WRONG RESULT
9312
9313
9314
9315
9316 034556
9317 034556 000004
9318 034560 012700 000515
9319 034564 013701 034614
9320
9321 034570 032737 002000 063160 .SBTTL USER CONTROLLED BREAKPOINT -- BIT10
9322 034576 001401      BIT      #BIT10,#BPTLOC ;BREAKPOINT HALT SET ??
9323 034600 000000      BEQ      .+4          ;BR IF NOT
9324 034602 005004      HALT
9325 034604 005005      CLR      R4          ;BREAK-DEPRESS CONTINUE TO CONTINUE
9326 034606 005003      CLR      R5          ;RESULT S / B = 000000
9327 034610 000257      CLR      R3          ;SRC OPR = 000000
9328 034612 000270      CCC
9329
9330 034614 050503      2$:  BIS      R5,R3      ;[DEST] = 000000
9331
9332 034616 100403      SEN
9333 034620 001002      ;N:C = 1000
9334 034622 102401      BMI      3$
9335 034624 103001      BNE      3$
9336
9337 034626 104002      3$:  ERROR    2      ;BIS FAILED TO ALTER CODES PROPERLY
9338
9339 034630 020403      4$:  CMP      R4,R3      ;CORRECT RESULT ?
9340 034632 001401      BEQ      TS1516      ;;BR IF YES
9341
9342 034634 104002      5$:  ERROR    2      ;BIS DELIVERED THE WRONG RESULT
9343

```

M15

9344
9345
9346
9347 034636
9348 034636 000004
9349 034640 012700 000516
9350 034644 013701 034670
9351 034650 012704 100000
9352 034654 012705 077777
9353 034660 012703 177777
9354 034664 000257
9355 034666 000267
9356
9357 034670 040503
9358
9359 034672 100003
9360 034674 001402
9361 034676 102401
9362 034700 103401
9363
9364 034702 104002
9365
9366 034704 020403
9367 034706 001401
9368
9369 034710 104002
9370
9371
9372
9373
9374 034712
9375 034712 000004
9376 034714 012700 000517
9377 034720 013701 034736
9378 034724 005004
9379 034726 005005
9380 034730 005003
9381 034732 000257
9382 034734 000270
9383
9384 034736 040503
9385
9386 034740 100403
9387 034742 001002
9388 034744 102401
9389 034746 103001
9390
9391 034750 104002
9392
9393 034752 020403
9394 034754 001401
9395
9396 034756 104002
9397
9398
9399

```
*****
;TEST 516      BIC TEST - SMO,DMO - N:C = 0111
*****
↑TS16:
SCOPE
MOV      #516,R0      ;CALL THE SCOPE LOOP UTILITY
MOV      2#25,R1     ;LOAD R0 WITH TEST NUMBER
MOV      #100000,R4  ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #77777,R5   ;RESULT S / B = 100000
MOV      #-1,R3      ;SRC OPR = 77777
CCC      ;[DEST] = 177777
267      ;CLEAR FLAGS
          ;N:C = 0111

2$:      BIC      R5,R3      ;TEST THE BIC
          ;N:C = 1001 ?

          BPL      3$
          BEQ      3$
          BVS      3$
          BCS      4$

3$:      ERROR    2          ;BIC FAILED TO ALTER CODES PROPERLY

4$:      CMP      R4,R3      ;CORRECT RESULT ?
          BEQ      TS1517    ;;BR IF YES

5$:      ERROR    2          ;BIC DELIVERED THE WRONG RESULT

*****
;TEST 517      BIC TEST - SMO,DMO - N:C = 1000
*****
↑TS17:
SCOPE
MOV      #517,R0      ;CALL THE SCOPE LOOP UTILITY
MOV      2#25,R1     ;LOAD R0 WITH TEST NUMBER
CLR      R4           ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R5           ;RESULT S / B = 000000
CLR      R3           ;SRC OPR = 000000
CCC      ;[DEST] = 000000
SEN      ;CLEAR FLAGS
          ;N:C = 1000

2$:      BIC      R5,R3      ;TEST THE BIC
          ;N:C = 0100

          BMI      3$
          BNE      3$
          BVS      3$
          BCC      4$

3$:      ERROR    2          ;BIC FAILED TO ALTER CODES PROPERLY

4$:      CMP      R4,R3      ;CORRECT RESULT ?
          BEQ      TS1520    ;;BR IF YES

5$:      ERROR    2          ;BIC DELIVERED THE WRONG RESULT

*****
;TEST 520      BIT TEST - SMO,DMO - N:C = 0111
*****
```

N15

```

9400
9401 034760
9402 034760 000004
9403 034762 012700 000520
9404 034766 013701 035012
9405 034772 012704 100000
9406 034776 012705 100000
9407 035002 012703 100000
9408 035006 000257
9409 035010 000267
9410
9411 035012 030503
9412
9413 035014 100003
9414 035016 001402
9415 035020 102401
9416 035022 103401
9417
9418 035024 104002
9419
9420 035026 020403
9421 035030 001402
9422
9423 035032 011203
9424 035034 104002
9425
9426
9427
9428
9429 035036
9430 035036 000004
9431 035040 012700 000521
9432 035044 013701 035066
9433 035050 012704 125252
9434 035054 012705 052525
9435 035060 010403
9436 035062 000257
9437 035064 000270
9438
9439 035066 030503
9440
9441 035070 100403
9442 035072 001002
9443 035074 102401
9444 035076 103001
9445
9446 035100 104002
9447
9448 035102 020403
9449 035104 001401
9450
9451 035106 104002
9452
9453
9454
9455

```

```

*****
TST520:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #520,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #100000,R5 ;SRC OPR = 100000
MOV #100000,R3 ;[DEST] = 100000
CCC ;CLEAR FLAGS
267 ;N:C = 0111

2$: BIT R5,R3 ;TEST THE BIT
BPL 3$ ;N:C = 1001
BEQ 3$
BVS 3$
BCS 4$

3$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY

4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST521 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 2 ;BIT DELIVERED A RESULT

*****
*TEST 521 BIT TEST - SMO,DMD - N:C = 1000
*****
TST521:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #521,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125252,R4 ;RESULT S / B = 125252
MOV #52525,R5 ;SRC OPR = 52525
MOV R4,R3 ;[DEST] = 125252
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

2$: BIT R5,R3 ;TEST THE BIT
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

3$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY

4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST522 ;;BR IF YES

5$: ERROR 2 ;BIT DELIVERED A RESULT

*****
*TEST 522 CMP TEST - SMO,DMD - N:C = 0110
*****

```

9456 035110
 9457 035110 000004
 9458 035112 012700 000522
 9459 035116 013701 035140
 9460 035122 012704 000001
 9461 035126 005005
 9462 035130 012703 000001
 9463 035134 000257
 9464 035136 000266
 9465
 9466 035140 020503
 9467
 9468 035142 100003
 9469 035144 001402
 9470 035146 102401
 9471 035150 103401
 9472
 9473 035152 104002
 9474
 9475 035154 020403
 9476 035156 001401
 9477
 9478 035160 104002
 9479
 9480
 9481
 9482
 9483 035162
 9484 035162 000004
 9485 035164 012700 000523
 9486 035170 013701 035212
 9487 035174 012704 177777
 9488 035200 012705 177777
 9489 035204 010403
 9490 035206 000257
 9491 035210 000272
 9492
 9493 035212 020503
 9494
 9495 035214 100403
 9496 035216 001002
 9497 035220 102401
 9498 035222 103001
 9499
 9500 035224 104002
 9501
 9502 035226 020403
 9503 035230 001401
 9504
 9505 035232 104002
 9506
 9507
 9508
 9509
 9510 035234
 9511 035234 000004

TST522:
 SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #522,R0 ;;LOAD R0 WITH TEST NUMBER
 MOV 2#25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #+1,R4 ;RESULT S / B = +1
 CLR R5 ;SRC OPR = 000000
 MOV #+1,R3 ;[DEST] = +1
 CCC ;CLEAR FLAGS
 266 ;N:C = 0110
 25: CMP R5,R3 ;TEST THE CMP
 BPL 35 ;N:C = 1001
 BEQ 35
 BVS 35
 BCS 45
 35: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
 45: CMP R4,R3 ;CORRECT RESULT ?
 BEQ TST523 ;;BR IF YES
 55: ERROR 2 ;CMP DELIVERED A RESULT
 ;*****
 ;*TEST 523 CMP TEST - SMO,DMO - N:C = 1010
 ;*****
 TST523:
 SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #523,R0 ;;LOAD R0 WITH TEST NUMBER
 MOV 2#25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #-1,R4 ;RESULT S / B = 177777
 MOV #-1,R5 ;SRC OPR = 177777
 MOV R4,R3 ;[DEST] = 177777
 CCC ;CLEAR FLAGS
 272 ;N:C = 1010
 25: CMP R5,R3 ;TEST THE CMP
 BMI 35 ;N:C = 0100
 BNE 35
 BVS 35
 BCC 45
 35: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
 45: CMP R4,R3 ;CORRECT RESULT ?
 BEQ TST524 ;;BR IF YES
 55: ERROR 2 ;CMP DELIVERED A RESULT
 ;*****
 ;*TEST 524 CMP TEST - SMO,DMO - N:C = 0000
 ;*****
 TST524:
 SCOPE ;CALL THE SCOPE LOOP UTILITY

```

9512 035236 012700 000524      MOV      #524,R0      ;;LOAD R0 WITH TEST NUMBER
9513 035242 013701 035264      MOV      @#25,R1     ;;LOAD R1 WITH TEST INSTRUCTION WORD
9514 035246 012704 000001      MOV      #+1,R4     ;;RESULT S / B = +1
9515 035252 012705 100000      MOV      #100000,R5 ;;SRC OPR = 100000
9516 035256 012703 000001      MOV      #+1,R3     ;[DEST] = +1
9517 035262 000257      CCC                     ;CLEAR FLAGS
9518
9519 035264 020503      2$:      CMP      R5,R3      ;TEST THE CMP
9520
9521 035266 100403      BMI      3$           ;N:C = 0010
9522 035270 001402      BEQ      3$
9523 035272 102001      BVC      3$
9524 035274 103001      BCC      4$
9525
9526 035276 104002      3$:      ERROR    2           ;CMP FAILED TO ALTER CODES PROPERLY
9527
9528 035300 020403      4$:      CMP      R4,R3      ;CORRECT RESULT ?
9529 035302 001401      BEQ      T$T525      ;;BR IF YES
9530
9531 035304 104002      5$:      ERROR    2           ;CMP DELIVERED A RESULT
9532
9533      ;*****
9534      ;*TEST 525      BIS TEST - SMO,DM1 - N:C = 0111
9535      ;*****
9536      T$T525:
9537 035306 000004      SCOPE
9538 035310 012700 000525      MOV      #525,R0     ;CALL THE SCOPE LOOP UTILITY
9539 035314 013701 035344      MOV      @#25,R1     ;LOAD R0 WITH TEST NUMBER
9540 035320 012702 063236      MOV      #MBUF0,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
9541 035324 012704 177777      MOV      #-1,R4     ;DEST ADDR = MBUF0
9542 035330 012705 125252      MOV      #125252,R5 ;RESULT S / B = 177777
9543 035334 012712 052525      MOV      #52525,(R2) ;SRC OPR = 125252
9544 035340 000257      CCC                     ;[DEST] = 52525
9545 035342 000267      267                     ;CLEAR FLAGS
9546
9547 035344 050512      2$:      BIS      R5,(R2)     ;N:C = 0111
9548
9549 035346 100003      BPL      3$           ;TEST THE BIS
9550 035350 001402      BEQ      3$
9551 035352 102401      BVS      3$
9552 035354 103401      BCS      4$
9553
9554 035356 104001      3$:      ERROR    1           ;N:C = 1001
9555
9556 035360 020412      4$:      CMP      R4,(R2)     ;BIS FAILED TO ALTER CODES PROPERLY
9557 035362 001402      BEQ      T$T526      ;CORRECT RESULT ?
9558
9559 035364 011203      5$:      MOV      (R2),R3     ;;BR IF YES
9560 035366 104001      ERROR    1           ;GET THE WAS DATA
9561
9562      ;*****
9563      ;*TEST 526      BIS TEST - SMO,DM1 - N:C = 1000
9564      ;*****
9565      T$T526:
9566 035370 000004      SCOPE
9567 035372 012700 000526      MOV      #526,R0     ;CALL THE SCOPE LOOP UTILITY
                        ;;LOAD R0 WITH TEST NUMBER

```

```

9568 035376 013701 035420      MOV      2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
9569 035402 012702 063236      MOV      #MBOFO,R2  ;DEST ADDR = MBOFO
9570 035406 005004                CLR      R4          ;RESULT S / B = 000000
9571 035410 005005                CLR      R5          ;SRC OPR = 000000
9572 035412 005012                CLR      (R2)        ;[DEST] = 000000
9573 035414 000257                CCC                ;CLEAR FLAGS
9574 035416 000270                SEN                ;N:C = 1000
9575
9576 035420 050512      2$:   BIS      R5,(R2)      ;TEST THE BIS
9577
9578 035422 100403                BMI      3$          ;N:C = 0100
9579 035424 001002                BNE      3$
9580 035426 102401                BVS      3$
9581 035430 103001                BCC      4$
9582
9583 035432 104001      3$:   ERROR    1          ;BIS FAILED TO ALTER CODES PROPERLY
9584
9585 035434 020412      4$:   CMP      R4,(R2)      ;CORRECT RESULT ?
9586 035436 001402                BEQ      T$T527      ;;BR IF YES
9587
9588 035440 011203                MOV      (R2),R3     ;GET THE WAS DATA
9589 035442 104001      5$:   ERROR    1          ;BIS DELIVERED THE WRONG RESULT
9590
9591 ;:*****
9592 ;*TEST 527      BIC TEST - SMO,DM1 - N:C = 0111
9593 ;:*****
9594 T$T527:
9595 035444 000004                SCOPE                ;CALL THE SCOPE LOOP UTILITY
9596 035446 012700 000527      MOV      #527,R0     ;LOAD R0 WITH TEST NUMBER
9597 035452 013701 035502      MOV      2#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
9598 035456 012702 063236      MOV      #MBOFO,R2  ;DEST ADDR = MBOFO
9599 035462 012704 100000      MOV      #100000,R4 ;RESULT S / B = 100000
9600 035466 012705 077777      MOV      #77777,R5  ;SRC OPR = 77777
9601 035472 012712 177777      MOV      #-1,(R2)   ;[DEST] = 177777
9602 035476 000257                CCC                ;CLEAR FLAGS
9603 035500 000267                267                ;N:C = 0111
9604
9605 035502 040512      2$:   BIC      R5,(R2)      ;TEST THE BIC
9606
9607 035504 100003                BPL      3$          ;N:C = 1001
9608 035506 001402                BEQ      3$
9609 035510 102401                BVS      3$
9610 035512 103401                BCS      4$
9611
9612 035514 104001      3$:   ERROR    1          ;BIC FAILED TO ALTER CODES PROPERLY
9613
9614 035516 020412      4$:   CMP      R4,(R2)      ;CORRECT RESULT ?
9615 035520 001402                BEQ      T$T530      ;;BR IF YES
9616
9617 035522 011203                MOV      (R2),R3     ;GET THE WAS DATA
9618 035524 104001      5$:   ERROR    1          ;BIC DELIVERED THE WRONG RESULT
9619
9620 ;:*****
9621 ;*TEST 530      BIC TEST - SMO,DM1 - N:C = 1000
9622 ;:*****
9623 035526 T$T530:

```


E16

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.A.P11 08-FEB-77 16:17 T530

MACY11 27(1006) 08-FEB-77 16:23 PAGE 176
BIC TEST - SMO,DMI - N:C = 1000

```

9624 035526 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
9625 035530 012700 000530  MOV      #530,R0      ;:LOAD R0 WITH TEST NUMBER
9626 035534 013701 035556  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
9627 035540 012702 063236  MOV      #MBUFO,R2    ;DEST ADDR = MBUFO
9628 035544 005004          CLR      R4           ;RESULT S / B = 000000
9629 035546 005005          CLR      R5           ;SRC OPR = 000000
9630 035550 005012          CLR      (R2)        ;[DEST] = 000000
9631 035552 000257          CCC          ;CLEAR FLAGS
9632 035554 000270          SEN          ;N:C = 1000
9633
9634 035556 040512          2$:  BIC      R5,(R2)  ;TEST THE BIC
9635
9636 035560 100403          BMI      3$          ;N:C = 0100
9637 035562 001002          BNE      3$
9638 035564 102401          BVS      3$
9639 035566 103001          BCC      4$
9640
9641 035570 104001          3$:  ERROR   1         ;BIC FAILED TO ALTER CODES PROPERLY
9642
9643 035572 020412          4$:  CMP      R4,(R2)  ;CORRECT RESULT ?
9644 035574 001402          BEQ      T$T531      ;;BR IF YES
9645
9646 035576 011203          MOV      (R2),R3     ;GET THE WAS DATA
9647 035600 104001          5$:  ERROR   1         ;BIC DELIVERED THE WRONG RESULT
9648
9649
9650
9651
9652 035602          ;:*****
9653 035602 000004          ;*TEST 531  BIT TEST - SMO,DMI - N:C = 0111
9654 035604 012700 000531  ;:*****
9655 035610 013701 035640  T$T531:
9656 035614 012702 063236  SCOPE          ;CALL THE SCOPE LOOP UTILITY
9657 035620 012704 100000  MOV      #531,R0      ;:LOAD R0 WITH TEST NUMBER
9658 035624 012705 100000  MOV      @#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
9659 035630 012712 100000  MOV      #MBUFO,R2    ;DEST ADDR = MBUFO
9660 035634 000257          MOV      #100000,R4   ;RESULT S / B = 100000
9661 035636 000267          MOV      #100000,R5   ;SRC OPR = 100000
9662
9663 035640 030512          2$:  BIT      R5,(R2)  ;TEST THE BIT
9664
9665 035642 100003          BPL      3$          ;N:C = 1001
9666 035644 001402          BEQ      3$
9667 035646 102401          BVS      3$
9668 035650 103401          BCS      4$
9669
9670 035652 104001          3$:  ERROR   1         ;BIT FAILED TO ALTER CODES PROPERLY
9671
9672 035654 020412          4$:  CMP      R4,(R2)  ;CORRECT RESULT ?
9673 035656 001402          BEQ      T$T532      ;;BR IF YES
9674
9675 035660 011203          MOV      (R2),R3     ;GET THE WAS DATA
9676 035662 104001          5$:  ERROR   1         ;BIT DELIVERED A RESULT
9677
9678
9679
;:*****
;*TEST 532  BIT TEST - SMO,DMI - N:C = 1000
;:*****

```

F16

```

9680
9681 035664
9682 035664 000004
9683 035666 012700 000532
9684 035672 013701 035722
9685 035676 012702 063236
9686 035702 012704 052525
9687 035706 012705 125252
9688 035712 012712 052525
9689 035716 000257
9690 035720 000270
9691
9692 035722 030512
9693
9694 035724 100403
9695 035726 001002
9696 035730 102401
9697 035732 103001
9698
9699 035734 104001
9700
9701 035736 020412
9702 035740 001402
9703
9704 035742 011203
9705 035744 104001
9706
9707
9708
9709 035746
9710 035746 000004
9711 035750 012700 000533
9712 035754 013701 036004
9713 035760 012702 063236
9714 035764 012704 177777
9715 035770 012705 177777
9716 035774 012712 177777
9717 036000 000257
9718 036002 000272
9719
9720 036004 020512
9721
9722 036006 100403
9723 036010 001002
9724 036012 102401
9725 036014 103001
9726
9727 036016 104001
9728
9729 036020 020412
9730 036022 001402
9731
9732 036024 011203
9733 036026 104001
9734
9735

;*****
†T532:
SCOPE
MOV #532,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #52525,R4 ;DEST ADDR = MBUF0
MOV #125252,R5 ;RESULT S / B = 52525
MOV #52525,(R2) ;SRC OPR = 125252
CCC ;[DEST] = 52525
SEN ;CLEAR FLAGS
;N:C = 1000

2$: BIT R5,(R2) ;TEST THE BIT

BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

3$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T533 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIT DELIVERED A RESULT
;*****
;*TEST 533 CMP TEST - SMO,DM1 - N:C = 1010
;*****
†T533:
SCOPE
MOV #533,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;DEST ADDR = MBUF0
MOV #-1,R5 ;RESULT S / B = -1
MOV #-1,(R2) ;SRC OPR = 177777
CCC ;[DEST] = 177777
272 ;CLEAR FLAGS
;N:C = 1010

2$: CMP R5,(R2) ;TEST THE CMP

BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

3$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T534 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT
;*****

```

```

9736 ;*TEST 534      CMP TEST - SMO,DMI - N:C = 0110
9737 ;*****
9738 †T534:
9739      SCOPE
9740      MOV      #534,R0      ;CALL THE SCOPE LOOP UTILITY
9741      MOV      @#2$,R1     ;LOAD R0 WITH TEST NUMBER
9742      MOV      #MBUFD,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
9743      MOV      #+1,R4      ;DEST ADDR = MBUFD
9744      CLR      R5          ;RESULT S / B = +1
9745      MOV      #+1,(R2)    ;SRC OPR = 000000
9746      CCC          ;[DEST] = +1
9747      266          ;CLEAR FLAGS
9748                          ;N:C = 0110
9749      2$:      CMP      R5,(R2) ;TEST THE CMP
9750                          ;N:C = 1001
9751      BPL      3$
9752      BEQ      3$
9753      BVS      3$
9754      BCS      4$
9755
9756      3$:      ERROR    1      ;CMP FAILED TO ALTER CODES PROPERLY
9757
9758      4$:      CMP      R4,(R2) ;CORRECT RESULT ?
9759      BEQ      T535         ;;BR IF YES
9760
9761      5$:      MOV      (R2),R3 ;GET THE WAS DATA
9762      ERROR    1          ;CMP DELIVERED A RESULT
9763
9764 ;*****
9765 ;*TEST 535      CMP TEST - SMO,DMI - N:C = 0000
9766 ;*****
9767 †T535:
9768      SCOPE
9769      MOV      #535,R0      ;CALL THE SCOPE LOOP UTILITY
9770      MOV      @#2$,R1     ;LOAD R0 WITH TEST NUMBER
9771      MOV      #MBUFD,R2   ;LOAD R1 WITH TEST INSTRUCTION WORD
9772      MOV      #+1,R4      ;DEST ADDR = MBUFD
9773      MOV      #100000,R5   ;RESULT S / B = +1
9774      MOV      #+1,(R2)    ;SRC OPR = 100000
9775      CCC          ;[DEST] = +1
9776                          ;CLEAR FLAGS
9777      2$:      CMP      R5,(R2) ;TEST THE CMP
9778                          ;N:C = 0010
9779      BMI      3$
9780      BEQ      3$
9781      BVC      3$
9782      BCC      4$
9783
9784      3$:      ERROR    1      ;CMP FAILED TO ALTER CODES PROPERLY
9785
9786      4$:      CMP      R4,(R2) ;CORRECT RESULT ?
9787      BEQ      T536         ;;BR IF YES
9788
9789      5$:      MOV      (R2),R3 ;GET THE WAS DATA
9790      ERROR    1          ;CMP DELIVERED A RESULT
9791
    
```

H16

9792
9793
9794
9795 036170
9796 036170 000004
9797 036172 012700 000536
9798 036176 013701 036222
9799 036202 012704 177777
9800 036206 012705 063256
9801 036212 012703 052525
9802 036216 000257
9803 036220 000267
9804
9805 036222 051503
9806
9807 036224 100003
9808 036226 001402
9809 036230 102401
9810 036232 103401
9811
9812 036234 104002
9813
9814 036236 020403
9815 036240 001401
9816
9817 036242 104002
9818
9819
9820
9821
9822 036244
9823 036244 000004
9824 036246 012700 000537
9825 036252 013701 036272
9826 036256 005004
9827 036260 012705 063246
9828 036264 005003
9829 036266 000257
9830 036270 000270
9831
9832 036272 051503
9833
9834 036274 100403
9835 036276 001002
9836 036300 102401
9837 036302 103001
9838
9839 036304 104002
9840
9841 036306 020403
9842 036310 001401
9843
9844 036312 104002
9845
9846
9847

```
*****
; *TEST 536 BIS TEST - SM1,DMO - N:C = 0111
*****
†T536:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #536,R0 ;;LOAD R0 WITH TEST NUMBER
MOV 2#25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV 8-1,R4 ;RESULT S / B = 177777
MOV #DWTA+10,R5 ;SRC ADDR = DWTA+10
MOV #52525,R3 ;[DEST] = 52525
CCC ;CLEAR FLAGS
267 ;N:C = 0111

25: BIS (R5),R3 ;TEST THE BIS

BPL 35 ;N:C = 1001
BEQ 35
BVS 35
BCS 45

35: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY

45: CMP R4,R3 ;CORRECT RESULT ?
BEQ T537 ;;BR IF YES

55: ERROR 2 ;BIS DELIVERED THE WRONG RESULT

*****
; *TEST 537 BIS TEST - SM1,DMO - N:C = 1000
*****
†T537:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #537,R0 ;;LOAD R0 WITH TEST NUMBER
MOV 2#25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #DWTA,R5 ;SRC ADDR = DWTA
CLR R3 ;[DEST] = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

25: BIS (R5),R3 ;TEST THE BIS

BMI 35 ;N:C = 0100
BNE 35
BVS 35
BCC 45

35: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY

45: CMP R4,R3 ;CORRECT RESULT ?
BEQ T540 ;;BR IF YES

55: ERROR 2 ;BIS DELIVERED THE WRONG RESULT

*****
; *TEST 540 BIC TEST - SM1,DMO - N:C = 0111
*****
```

9848
9849 036314
9850 036314 000004
9851 036316 012700 000540
9852 036322 013701 036352
9853 036326 012704 100000
9854 036332 012705 063242
9855 036336 012703 177777
9856 036342 012715 077777
9857 036346 000257
9858 036350 000267
9859
9860 036352 041503
9861
9862 036354 100003
9863 036356 001402
9864 036360 102401
9865 036362 103401
9866
9867 036364 104002
9868
9869 036366 020403
9870 036370 001401
9871
9872 036372 104002
9873
9874
9875
9876
9877 036374
9878 036374 000004
9879 036376 012700 000541
9880 036402 013701 036422
9881 036406 005004
9882 036410 012705 063246
9883 036414 005003
9884 036416 000257
9885 036420 000270
9886
9887 036422 041503
9888
9889 036424 100403
9890 036426 001002
9891 036430 102401
9892 036432 103001
9893
9894 036434 104002
9895
9896 036436 020403
9897 036440 001401
9898
9899 036442 104002
9900
9901
9902
9903

```
*****  
†T540:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #540,R0 ;:LOAD R0 WITH TEST NUMBER  
MOV #25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #100000,R4 ;:RESULT S / B = 100000  
MOV #MBUF1,R5 ;:SRC ADDR = MBUF1  
MOV #-1,R3 ;:[DEST] = 177777  
MOV #77777,(R5) ;:SRC OPR = 77777  
CCC ;:CLEAR FLAGS  
267 ;:N:C = 0111  
  
2$: BIC (R5),R3 ;:TEST THE BIC  
  
BPL 3$ ;:N:C = 1001 ?  
BEQ 3$  
BVS 3$  
BCS 4$  
  
3$: ERROR 2 ;:BIC FAILED TO ALTER CODES PROPERLY  
  
4$: CMP R4,R3 ;:CORRECT RESULT ?  
BEQ T541 ;:BR IF YES  
  
5$: ERROR 2 ;:BIC DELIVERED THE WRONG RESULT
```

```
*****  
†T541: BIC TEST - SM1,DMD - N:C = 1000  
*****  
†T541:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #541,R0 ;:LOAD R0 WITH TEST NUMBER  
MOV #25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD  
CLR R4 ;:RESULT S / B = 000000  
MOV #DWTA,R5 ;:SRC ADDR = DWTA  
CLR R3 ;:[DEST] = 000000  
CCC ;:CLEAR FLAGS  
SEN ;:N:C = 1000  
  
2$: BIC (R5),R3 ;:TEST THE BIC  
  
BMI 3$ ;:N:C = 0100  
BNE 3$  
BVS 3$  
BCC 4$  
  
3$: ERROR 2 ;:BIC FAILED TO ALTER CODES PROPERLY  
  
4$: CMP R4,R3 ;:CORRECT RESULT ?  
BEQ T542 ;:BR IF YES  
  
5$: ERROR 2 ;:BIC DELIVERED THE WRONG RESULT
```

```
*****  
†T542: BIT TEST - SM1,DMD - N:C = 0111  
*****
```

9904 036444
9905 036444 000004
9906 036446 012700 000542
9907 036452 013701 036474
9908 036456 012704 100000
9909 036462 012705 063250
9910 036466 010403
9911 036470 000257
9912 036472 000267

TST542: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #542,R0 ;:LOAD RO WITH TEST NUMBER
MOV #25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;:RESULT S / B = 100000
MOV #DWTA+2,R5 ;:SRC ADDR = DWTA+2
MOV R4,R3 ;:(DEST) = 100000
CCC ;:CLEAR FLAGS
267 ;:N:C = 0111

9914 036474 031503
9915
9916 036476 100003
9917 036500 001402
9918 036502 102401
9919 036504 103401

2\$: BIT (R5),R3 ;TEST THE BIT
BPL 3\$;N:C = 1001 ?
BEQ 3\$
BVS 3\$
BCS 4\$

9921 036506 104002
9922
9923 036510 020403
9924 036512 001401

3\$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;:CORRECT RESULT ?
BEQ TST543 ;:BR IF YES

9926 036514 104002
9927
9928
9929

5\$: ERROR 2 ;BIT DELIVERED A RESULT
;:*****
;:TEST 543 BIT TEST - SM1,DMD - N:C = 1000
;:*****

9931 036516
9932 036516 000004
9933 036520 012700 000543
9934 036524 013701 036546
9935 036530 012704 052525
9936 036534 012705 063256
9937 036540 010403
9938 036542 000257
9939 036544 000270

TST543: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #543,R0 ;:LOAD RO WITH TEST NUMBER
MOV #25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #52525,R4 ;:RESULT S / B = 52525
MOV #DWTA+10,R5 ;:SRC ADDR = DWTA+10
MOV R4,R3 ;:(DEST) = 52525
CCC ;:CLEAR FLAGS
SEN ;:N:C = 1000

9941 036546 031503
9942
9943 036550 100403
9944 036552 001002
9945 036554 102401
9946 036556 103001

2\$: BIT (R5),R3 ;TEST THE BIT
BMI 3\$;N:C = 0100
BNE 3\$
BVS 3\$
BCC 4\$

9948 036560 104002
9949
9950 036562 020403
9951 036564 001401

3\$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;:CORRECT RESULT ?
BEQ TST544 ;:BR IF YES

9953 036566 104002
9954
9955
9956

5\$: ERROR 2 ;BIT DELIVERED A RESULT
;:*****
;:TEST 544 CMP TEST - SM1,DMD - N:C = 0110
;:*****

9957 036570
9958 036570 000004
9959 036572 012700 000544

TST544: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #544,R0 ;:LOAD RO WITH TEST NUMBER

K16

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 182
CMP TEST - SM1,DMO - N:C = 0110

```

9960 036576 013701 036620      MOV      2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
9961 036602 012704 000001      MOV      #+1,R4      ;RESULT S / B = +1
9962 036606 012705 063246      MOV      #DWTA,R5    ;SRC ADDR = DWTA
9963 036612 010403      MOV      R4,R3      ;[DEST] = +1
9964 036614 000257      CCC      ;CLEAR FLAGS
9965 036616 000266      266      ;N:C = 0110
9966
9967 036620 021503      2$:      CMP      (R5),R3      ;TEST THE CMP
9968
9969 036622 100003      BPL      3$          ;N:C = 1001
9970 036624 001402      BEQ      3$
9971 036626 102401      BVS      3$
9972 036630 103401      BCS      4$
9973
9974 036632 104002      3$:      ERROR      2          ;CMP FAILED TO ALTER CODES PROPERLY
9975
9976 036634 020403      4$:      CMP      R4,R3      ;CORRECT RESULT ?
9977 036636 001401      BEQ      T$T545      ;;BR IF YES
9978
9979 036640 104002      5$:      ERROR      2          ;CMP DELIVERED A RESULT
9980
9981      ;*****
9982      ;*TEST 545      CMP TEST - SM1,DMO - N:C = 1010
9983      ;*****
9984      T$T545:
9985      SCOPE      ;CALL THE SCOPE LOOP UTILITY
9986      MOV      #545,R0    ;;LOAD R0 WITH TEST NUMBER
9987      MOV      2#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
9988      MOV      #-1,R4     ;RESULT S / B = 177777
9989      MOV      #DWTA+2,R5 ;SRC ADDR = DWTA+2
9990      MOV      R4,R3     ;[DEST] = 177777
9991      CCC      ;CLEAR FLAGS
9992      272      ;N:C = 1010
9993
9994 036672 021503      2$:      CMP      (R5),R3      ;TEST THE CMP
9995
9996 036674 100403      BMI      3$          ;N:C = 0100
9997 036676 001002      BNE      3$
9998 036700 102401      BVS      3$
9999 036702 103001      BCC      4$
10000
10001 036704 104002      3$:      ERROR      2          ;CMP FAILED TO ALTER CODES PROPERLY
10002
10003 036706 020403      4$:      CMP      R4,R3      ;CORRECT RESULT ?
10004 036710 001401      BEQ      T$T546      ;;BR IF YES
10005
10006 036712 104002      5$:      ERROR      2          ;CMP DELIVERED A RESULT
10007
10008      ;*****
10009      ;*TEST 546      CMP TEST - SM1,DMO - N:C = 0000
10010      ;*****
10011      T$T546:
10012      SCOPE      ;CALL THE SCOPE LOOP UTILITY
10013      MOV      #546,R0    ;;LOAD R0 WITH TEST NUMBER
10014      MOV      2#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
10015      MOV      #+1,R4     ;RESULT S / B = +1

```

10016 036732 012705 063242
10017 036736 012703 000001
10018 036742 012715 100000
10019 036746 000257

MOV #MBUF1,R5 ;SRC ADDR = MBUF1
MOV #+1,R3 ;[DEST] = +1
MOV #100000,(R5) ;SRC OPR = 100000
CCC ;CLEAR FLAGS

10020
10021 036750 021503
10022
10023 036752 100403
10024 036754 001402
10025 036756 102001
10026 036760 103001

2\$: CMP (R5),R3 ;TEST THE CMP
BMI 3\$;N:C = 0010
BEQ 3\$
BVC 3\$
BCC 4\$

10027
10028 036762 104002
10029
10030 036764 020403
10031 036766 001401

3\$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST547 ;;BR IF YES

10032
10033 036770 104002

5\$: ERROR 2 ;CMP DELIVERED A RESULT

10034
10035
10036
10037

;;*****
;*TEST 547 BIS SMI,DMI TEST - N:C = 0111
;;*****

10038 036772
10039 036772 000004
10040 036774 012700 000547
10041 037000 013701 037030
10042 037004 012702 063236
10043 037010 012704 177777
10044 037014 012705 063256
10045 037020 012712 052525
10046 037024 000257
10047 037026 000267

TST547: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #547,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #-1,R4 ;RESULT S / B = 1777777
MOV #DWTA+10,R5 ;SOURCE ADDR = DWTA+10
MOV #52525,(R2) ;[DEST] = 052525
CCC ;CLEAR FLAGS
267 ;N:C = 0111

10048
10049 037030 051512
10050
10051 037032 100003
10052 037034 001402
10053 037036 102401
10054 037040 103401

2\$: BIS (R5),(R2) ;TEST THE BIS
BPL 3\$;N:C = 1001?
BEQ 3\$
BVS 3\$
BCS 4\$

10055
10056 037042 104001
10057
10058 037044 020412
10059 037046 001402

3\$: ERROR 1 ;BIS FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST550 ;;BR IF YES

10060
10061 037050 011203
10062 037052 104001

5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIS DELIVERED THE WRONG RESULT

10063
10064
10065
10066

;;*****
;*TEST 550 BIS SMI,DMI TEST - N:C = 1000
;;*****

10067 037054
10068 037054 000004
10069 037056 012700 000550
10070 037062 013701 037106
10071 037066 012702 063236

TST550: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #550,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD

M16

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 184
BIS SM1,DM1 TEST - N:C = 1000

```

10072 037072 005004          CLR      R4          ;RESULT S / B = 000000
10073 037074 012705 063246  MOV      #DWTA,R5   ;SOURCE ADDR = DWTA
10074 037100 005012          CLR      (R2)       ;[DEST] = 000000
10075 037102 000257          CCC          ;CLEAR FLAGS
10076 037104 000270          SEN          ;N:C = 1000
10077
10078 037106 051512          2S:     BIS      (R5),(R2) ;TEST THE BIS
10079
10080 037110 100403          BMI      3$        ;N:C = 0100 ?
10081 037112 001002          BNE      3$
10082 037114 102401          BVS      3$
10083 037116 103001          BCC      4$
10084
10085 037120 104001          3S:     ERROR    1      ;BIS FAILED TO ALTER CODES PROPERLY
10086
10087 037122 020412          4S:     CMP      R4,(R2) ;CORRECT RESULT ?
10088 037124 001402          BEQ      TST551    ;;BR IF YES
10089
10090 037126 011203          MOV      (R2),R3   ;GET THE WAS DATA
10091 037130 104001          5S:     ERROR    1      ;BIS DELIVERED THE WRONG RESULT
10092

```

```

*****
;TEST 551      BIC SM1,DM1 TEST - N:C = 0111
*****

```

```

10093
10094
10095
10096 037132          TST551:
10097 037132 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10098 037134 012700 000551  MOV      #551,R0   ;;LOAD R0 WITH TEST NUMBER
10099 037140 013701 037174  MOV      2#2$ ,R1  ;;LOAD R1 WITH TEST INSTRUCTION WORD
10100 037144 012702 063236  MOV      #MBOF0,R2 ;;DEST ADDR = MBOF0
10101 037150 012704 100000  MOV      #100000,R4 ;;RESULT S / B = 100000
10102 037154 012705 063242  MOV      #MBOF1,R5 ;;SOURCE ADDR = MBOF1
10103 037160 012715 077777  MOV      #77777,(R5) ;;[SOURCE] = 77777
10104 037164 012712 177777  MOV      #-1,(R2)  ;;[DEST] = 177777
10105 037170 000257          CCC          ;CLEAR FLAGS
10106 037172 000267          267         ;N:C = 0111
10107
10108 037174 041512          2S:     BIC      (R5),(R2) ;TEST THE BIC
10109
10110 037176 100003          BPL      3$        ;N:C = 1001 ?
10111 037200 001402          BEQ      3$
10112 037202 102401          BVS      3$
10113 037204 103401          BCS      4$
10114
10115 037206 104001          3S:     ERROR    1      ;BIC FAILED TO ALTER CODES PROPERLY
10116
10117 037210 020412          4S:     CMP      R4,(R2) ;CORRECT RESULT ?
10118 037212 001402          BEQ      TST552    ;;BR IF YES
10119
10120 037214 011203          MOV      (R2),R3   ;GET THE WAS DATA
10121 037216 104001          5S:     ERROR    1      ;BIC DELIVERED THE WRONG RESULT
10122

```

```

*****
;TEST 552      BIC SM1,DM1 TEST - N:C = 1000
*****

```

```

10123
10124
10125
10126 037220          TST552:
10127 037220 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY

```

B01

MAINDEC-11-DGKDA-A K011-K BASIC LOGIC TESTS
DGKDA.P11 08-FEB-77 16:17 T552

MACY11 27(1006) 08-FEB-77 16:23 PAGE 185
BIC SM1,DM1 TEST - N:C = 1000

```

10128 037222 012700 000552      MOV      #552,R0      ;;LOAD R0 WITH TEST NUMBER
10129 037226 013701 037254      MOV      @#25,R1     ;;LOAD R1 WITH TEST INSTRUCTION WORD
10130 037232 012702 063236      MOV      #MBUFO,R2  ;;DEST ADDR = MBUFO
10131 037236 005004          CLR      R4         ;;RESULT S / B = 000000
10132 037240 012705 063242      MOV      #MBUF1,R5  ;;SOURCE ADDR = MBUF1
10133 037244 005015          CLR      (R5)      ;;[SOURCE] = 000000
10134 037246 005012          CLR      (R2)      ;;[DEST] = 000000
10135 037250 000257          CCC          ;;CLEAR FLAGS
10136 037252 000270          SEN          ;;N:C = 1000
10137
10138 037254 041512      2$:      BIC      (R5),(R2)  ;TEST THE BIC
10139
10140 037256 100403          BMI      3$        ;N:C = 0100 ?
10141 037260 001002          BNE      3$
10142 037262 102401          BVS      3$
10143 037264 103001          BCC      4$
10144
10145 037266 104001      3$:      ERROR    1        ;BIC FAILED TO ALTER CODES PROPERLY
10146
10147 037270 020412      4$:      CMP      R4,(R2)   ;CORRECT RESULT ?
10148 037272 001402          BEQ      T$T553    ;;BR IF YES
10149
10150 037274 011203          MOV      (R2),R3   ;GET THE WAS DATA
10151 037276 104001      5$:      ERROR    1        ;BIC DELIVERED THE WRONG RESULT
10152
10153      ;;*****
10154      ;;*TEST 553      BIT SM1,DM1 TEST - N:C = 1000
10155      ;;*****
10156 037300          T$T553:
10157 037300 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10158 037302 012700 000553      MOV      #553,R0   ;;LOAD R0 WITH TEST NUMBER
10159 037306 013701 037342      MOV      @#25,R1   ;;LOAD R1 WITH TEST INSTRUCTION WORD
10160 037312 012702 063236      MOV      #MBUFO,R2 ;;DEST ADDR = MBUFO
10161 037316 012704 125252      MOV      #125252,R4 ;;RESULT S / B = 125252
10162 037322 012705 063242      MOV      #MBUF1,R5 ;;SOURCE ADDR = MBUF1
10163 037326 012715 052525      MOV      #52525,(R5) ;;[SOURCE] = 052525
10164 037332 012712 125252      MOV      #125252,(R2) ;;[DEST] = 125252
10165 037336 000257          CCC          ;;CLEAR FLAGS
10166 037340 000270          SEN          ;;N:C = 1000
10167
10168 037342 031512      2$:      BIT      (R5),(R2)  ;TEST THE BIT
10169
10170 037344 100403          BMI      3$        ;N:C = 0100 ?
10171 037346 001002          BNE      3$
10172 037350 102401          BVS      3$
10173 037352 103001          BCC      4$
10174
10175 037354 104001      3$:      ERROR    1        ;BIT FAILED TO ALTER CODES PROPERLY
10176
10177 037356 020412      4$:      CMP      R4,(R2)   ;CORRECT RESULT ?
10178 037360 001402          BEQ      T$T554    ;;BR IF YES
10179
10180 037362 011203          MOV      (R2),R3   ;GET THE WAS DATA
10181 037364 104001      5$:      ERROR    1        ;BIT DELIVERED A RESULT
10182
10183      ;;*****

```


10240
10241 037542 020412
10242 037544 001402
10243
10244 037546 011203
10245 037550 104001
10246
10247
10248
10249
10250 037552
10251 037552 000004
10252 037554 012700 000556
10253 037560 013701 037612
10254 037564 012702 063236
10255 037570 012704 000001
10256 037574 012705 063242
10257 037600 005015
10258 037602 012712 000001
10259 037606 000257
10260 037610 000266
10261
10262 037612 021512
10263
10264 037614 100003
10265 037616 001402
10266 037620 102401
10267 037622 103401
10268
10269 037624 104001
10270
10271 037626 020412
10272 037630 001402
10273
10274 037632 011203
10275 037634 104001
10276
10277
10278
10279
10280 037636
10281 037636 000004
10282 037640 012700 000557
10283 037644 013701 037676
10284 037650 012702 063236
10285 037654 012704 000001
10286 037660 012705 063242
10287 037664 012715 100000
10288 037670 012712 000001
10289 037674 000257
10290
10291 037676 021512
10292
10293 037700 100403
10294 037702 001402
10295 037704 102001

4S: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST556 ;;BR IF YES

5S: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT

;*****
;TEST 556 CMP SM1,DM1 TEST - N:C = 0110
;*****
TST556:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #556,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #+1,R4 ;RESULT S / B = 000001
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
CLR (R5) ;[SOURCE] = 000000
MOV #+1,(R2) ;[DEST] = 000001
CCC ;CLEAR FLAGS
266 ;N:C = 0110

2S: CMP (R5),(R2) ;TEST THE CMP

BPL 3S ;N:C = 1001 ?
BEQ 3S
BVS 3S
BCS 4S

3S: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY

4S: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST557 ;;BR IF YES

5S: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT

;*****
;TEST 557 CMP SM1,DM1 TEST - N:C = 0000
;*****
TST557:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #557,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #+1,R4 ;RESULT S / B = 000001
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #100000,(R5) ;[SOURCE] = 000000
MOV #+1,(R2) ;[DEST] = 000001
CCC ;CLEAR FLAGS

2S: CMP (R5),(R2) ;TEST THE CMP

BMI 3S ;N:C = 0010 ?
BEQ 3S
BVC 3S

E01

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 188
CMP SM1,DM1 TEST - N:C = 0000

10296 037706 103001
10297
10298 037710 104001
10299
10300 037712 020412
10301 037714 001402
10302
10303 037716 011203
10304 037720 104001
10305

BCC 45
3S: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4S: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST560 ;;BR IF YES
5S: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT

10306
10307
10308
10309 037722
10310 037722 000004
10311 037724 012700 000560
10312 037730 013701 037750
10313 037734 012704 000377
10314 037740 012705 064555
10315 037744 005003
10316 037746 000257
10317

;TEST 560 BISB SM1,DM0 TEST - SOURCE ADDR ODD

TST560:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #560,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC

10318 037750 151503
10319
10320 037752 020403
10321 037754 001401
10322

2S: BISB (R5),R3 ;TEST THE BISB
CMP R4,R3 ;RESULT CORRECT ?
BEQ TST561 ;;BR IF YES

10323 037756 104002
10324
10325
10326
10327

3S: ERROR 2 ;BISB DELIVERED THE WRONG RESULT

;TEST 561 BISB SM1,DM1 TEST - SOURCE ADDR ODD

10328 037760
10329 037760 000004
10330 037762 012700 000561
10331 037766 013701 040012
10332 037772 012702 063236
10333 037776 012704 000377
10334 040002 012705 064555
10335 040006 005012
10336 040010 000257
10337

TST561:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #561,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFD,R2 ;DEST ADDR = MBUFD
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

10338 040012 151512
10339
10340 040014 020412
10341 040016 001402
10342
10343 040020 011203
10344 040022 104001
10345

2S: BISB (R5),(R2) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST562 ;;BR IF YES
3S: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT

10346
10347
10348
10349 040024
10350 040024 000004
10351 040026 012700 000562

;TEST 562 BISB SM1,DM2 TEST - SOURCE ADDR ODD

TST562:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #562,R0 ;LOAD R0 WITH TEST NUMBER

F01

MAINDEC-11-DGKDA-A KD11-K BASIC LOGIC TESTS
 DGKDA.P11 08-FEB-77 16:17 T562

MACY11 27(1006) 08-FEB-77 16:23 PAGE 189
 BISB SM1,DM2 TEST - SOURCE ADDR ODD

10352	040032	013701	040060	MOV	2#25,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
10353	040036	012702	063236	MOV	#MBUFO,R2	;DEST ADDR = MBUFO
10354	040042	012704	000377	MOV	#377,R4	;RESULT S / B = 377
10355	040046	012705	064555	MOV	#DBTA+1,R5	;SOURCE ADDR = DBTA+1
10356	040052	005012		CLR	(R2)	;[DEST] = 000000
10357	040054	010203		MOV	R2,R3	;DEST ADDR IN R3
10358	040056	000257		CCC		;SCOPE SYNC
10359						
10360	040060	151523		2\$: BISB	(R5),(R3)+	;TEST THE BISB
10361						
10362	040062	020412		CMP	R4,(R2)	;CORRECT RESULT
10363	040064	001402		BEQ	TST563	;BR IF YES
10364						
10365	040066	011203		MOV	(R2),R3	;GET THE WAS DATA
10366	040070	104001		3\$: ERROR	1	;BISB DELIVERED THE WRONG RESULT
10367						
10368				;*****		
10369				;#TEST 563 BISB SM1,DM3 TEST - SOURCE ADDR ODD		
10370				;*****		
10371	040072			TST563:		
10372	040072	000004		SCOPE		;CALL THE SCOPE LOOP UTILITY
10373	040074	012700	000563	MOV	#563,R0	;LOAD R0 WITH TEST NUMBER
10374	040100	013701	040130	MOV	2#25,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
10375	040104	012702	063236	MOV	#MBUFO,R2	;DEST ADDR = MBUFO
10376	040110	012704	000377	MOV	#377,R4	;RESULT S / B = 377
10377	040114	012705	064555	MOV	#DBTA+1,R5	;SOURCE ADDR = DBTA+1
10378	040120	005012		CLR	(R2)	;[DEST] = 000000
10379	040122	012703	063232	MOV	#ATA+10,R3	;BASE DEST ADDR = ATA+10
10380	040126	000257		CCC		;SCOPE SYNC
10381						
10382	040130	151533		2\$: BISB	(R5),2(R3)+	;TEST THE BISB
10383						
10384	040132	020412		CMP	R4,(R2)	;CORRECT RESULT
10385	040134	001402		BEQ	TST564	;BR IF YES
10386						
10387	040136	011203		MOV	(R2),R3	;GET THE WAS DATA
10388	040140	104001		3\$: ERROR	1	;BISB DELIVERED THE WRONG RESULT
10389						
10390				;*****		
10391				;#TEST 564 BISB SM1,DM4 TEST - SOURCE ADDR ODD		
10392				;*****		
10393	040142			TST564:		
10394	040142	000004		SCOPE		;CALL THE SCOPE LOOP UTILITY
10395	040144	012700	000564	MOV	#564,R0	;LOAD R0 WITH TEST NUMBER
10396	040150	013701	040200	MOV	2#25,R1	;LOAD R1 WITH TEST INSTRUCTION WORD
10397	040154	012702	063236	MOV	#MBUFO,R2	;DEST ADDR = MBUFO
10398	040160	012704	177400	MOV	#177400,R4	;RESULT S / B = 177400
10399	040164	012705	064555	MOV	#DBTA+1,R5	;SOURCE ADDR = DBTA+1
10400	040170	012703	063240	MOV	#MBUFO+2,R3	;BASE DEST ADDR = MBUFO+2
10401	040174	005012		CLR	(R2)	;[DEST] = 000000
10402	040176	000257		CCC		;SCOPE SYNC
10403						
10404	040200	151543		2\$: BISB	(R5),-(R3)	;TEST THE BISB
10405						
10406	040202	020412		CMP	R4,(R2)	;CORRECT RESULT
10407	040204	001402		BEQ	TST565	;BR IF YES

GO1

```

10408
10409 040206 011203
10410 040210 104001
10411
10412
10413
10414
10415 040212
10416 040212 000004
10417 040214 012700 000565
10418 040220 013701 040250
10419 040224 012702 063236
10420 040230 012704 000377
10421 040234 012705 064555
10422 040240 012703 063234
10423 040244 005012
10424 040246 000257
10425
10426 040250 151553
10427
10428 040252 020412
10429 040254 001402
10430
10431 040256 011203
10432 040260 104001
10433
10434
10435
10436
10437 040262
10438 040262 000004
10439 040264 012700 000566
10440 040270 013701 040320
10441 040274 012702 063236
10442 040300 012704 000377
10443 040304 012705 064555
10444 040310 012703 063244
10445 040314 005012
10446 040316 000257
10447
10448 040320 151563 177772
10449
10450 040324 020412
10451 040326 001402
10452
10453 040330 011203
10454 040332 104001
10455
10456
10457
10458
10459 040334
10460 040334 000004
10461 040336 012700 000567
10462 040342 013701 040372
10463 040346 012702 063236
    
```

```

3$: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;BISB DELIVERED THE WRONG RESULT

;*****
;TEST 565 BISB SM1,DM5 TEST - SOURCE ADDR ODD
;*****
TST565:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #565,R0 ;LOAD R0 WITH TEST NUMBER
    MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
    MOV #MBUF0,R2 ;DEST ADDR = MBUF0
    MOV #377,R4 ;RESULT S / B = 377
    MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
    MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
    CLR (R2) ;[DEST] = 000000
    CCC ;SCOPE SYNC

2$: BISB (R5),@-(R3) ;TEST THE BISB

    CMP R4,(R2) ;CORRECT RESULT
    BEQ TST566 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;BISB DELIVERED THE WRONG RESULT

;*****
;TEST 566 BISB SM1,DM6 TEST - SOURCE ADDR ODD
;*****
TST566:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #566,R0 ;LOAD R0 WITH TEST NUMBER
    MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
    MOV #MBUF0,R2 ;DEST ADDR = MBUF0
    MOV #377,R4 ;RESULT S / B = 377
    MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
    MOV #MBUF0+6,R3 ;BASE DEST ADDR = MBUF0+6
    CLR (R2) ;[DEST] = 000000
    CCC ;SCOPE SYNC

2$: BISB (R5),-6(R3) ;TEST THE BISB

    CMP R4,(R2) ;CORRECT RESULT
    BEQ TST567 ;;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;BISB DELIVERED THE WRONG RESULT

;*****
;TEST 567 BISB SM1,DM7 TEST - SOURCE ADDR ODD
;*****
TST567:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #567,R0 ;LOAD R0 WITH TEST NUMBER
    MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
    MOV #MBUF0,R2 ;DEST ADDR = MBUF0
    
```

HO1

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T567

MACY11 27(1006) 08-FEB-77 16:23 PAGE 191
 BISB SM1,DM7 TEST - SOURCE ADDR ODD

```

10464 040352 012704 000377      MOV      #377,R4          ;RESULT S / B = 377
10465 040356 012705 064555      MOV      #DBTA+1,R5     ;SOURCE ADDR = DBTA+1
10466 040362 012703 063222      MOV      #ATA,R3        ;BASE DEST ADDR = ATA
10467 040366 005012                CLR      (R2)           ;[DEST] = 000000
10468 040370 000257                CCC                     ;SCOPE SYNC
10469
10470 040372 151573 000010      2$:     BISB      (R5),D10(R3) ;TEST THE BISB
10471
10472 040376 020412                CMP      R4,(R2)        ;CORRECT RESULT
10473 040400 001402                BEQ      T$T570         ;;BR IF YES
10474
10475 040402 011203                MOV      (R2),R3        ;GET THE WAS DATA
10476 040404 104001      3$:     ERROR      1          ;BISB DELIVERED THE WRONG RESULT
10477
10478      ;*****
10479      ;*TEST 570      BISB SMO,DM2 TEST - DEST ADDR EVEN
10480      ;*****
10481      T$T570:
10482 040406 000004                SCOPE                   ;CALL THE SCOPE LOOP UTILITY
10483 040410 012700 000570      MOV      #570,R0        ;;LOAD R0 WITH TEST NUMBER
10484 040414 013701 040436      MOV      D#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
10485 040420 012702 063236      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
10486 040424 012704 000377      MOV      #377,R4        ;RESULT S / B = 377
10487 040430 010203                MOV      R2,R3          ;DEST ADDR IN R3
10488 040432 005012                CLR      (R2)           ;[DEST] = 000000
10489 040434 000257                CCC                     ;SCOPE SYNC
10490
10491 040436 150423      2$:     BISB      R4,(R3)+    ;TEST THE BISB
10492
10493 040440 020412                CMP      R4,(R2)        ;CORRECT RESULT
10494 040442 001402                BEQ      T$T571         ;;BR IF YES
10495
10496 040444 011203                MOV      (R2),R3        ;GET THE WAS DATA
10497 040446 104001      3$:     ERROR      1          ;BISB DELIVERED THE WRONG RESULT
10498
10499      ;*****
10500      ;*TEST 571      BISB SMO,DM1 TEST - DEST ADDR ODD
10501      ;*****
10502      T$T571:
10503 040450 000004                SCOPE                   ;CALL THE SCOPE LOOP UTILITY
10504 040452 012700 000571      MOV      #571,R0        ;;LOAD R0 WITH TEST NUMBER
10505 040456 013701 040506      MOV      D#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
10506 040462 012702 063236      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
10507 040466 012704 177400      MOV      #177400,R4     ;RESULT S / B = 177400
10508 040472 012705 000377      MOV      #377,R5        ;[R5]=SOURCE OPR = 377
10509 040476 012703 063237      MOV      #MBUFO+1,R3    ;ODD DEST ADDR IN R3
10510 040502 005012                CLR      (R2)           ;[DEST] = 000000
10511 040504 000257                CCC                     ;SCOPE SYNC
10512
10513 040506 150513      2$:     BISB      R5,(R3)    ;TEST THE BISB
10514
10515 040510 020412                CMP      R4,(R2)        ;CORRECT RESULT
10516 040512 001402                BEQ      T$T572         ;;BR IF YES
10517
10518 040514 011203                MOV      (R2),R3        ;GET THE WAS DATA
10519 040516 104001      3$:     ERROR      1          ;BISB DELIVERED THE WRONG RESULT

```


I01

10520
10521
10522
10523
10524 040520
10525 040520 000004
10526 040522 012700 000572
10527 040526 013701 040550
10528 040532 012702 063236
10529 040536 012704 000377
10530 040542 010203
10531 040544 005012
10532 040546 000257
10533
10534 040550 150413
10535
10536 040552 020412
10537 040554 001402
10538
10539 040556 011203
10540 040560 104001
10541
10542
10543
10544
10545 040562
10546 040562 000004
10547 040564 012700 000573
10548 040570 013701 040620
10549 040574 012702 063236
10550 040600 012704 177400
10551 040604 012705 064555
10552 040610 012703 063237
10553 040614 005012
10554 040616 000257
10555
10556 040620 151513
10557
10558 040622 020412
10559 040624 001402
10560
10561 040626 011203
10562 040630 104001
10563
10564
10565
10566
10567 040632
10568 040632 000004
10569 040634 012700 000574
10570 040640 013701 040652
10571 040644 012702 040660
10572 040650 000277
10573
10574 040652 000112
10575

```
*****  
*TEST 572 BISB SMO,DM1 TEST - DEST ADDR EVEN  
*****  
†T572:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #572,R0 ;LOAD R0 WITH TEST NUMBER  
MOV 2#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
MOV #377,R4 ;RESULT S / B = 377  
MOV R2,R3 ;DEST ADDR IN R3  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
2$: BISB R4,(R3) ;TEST THE BISB  
  
CMP R4,(R2) ;CORRECT RESULT  
BEQ T573 ;;BR IF YES  
  
3$: MOV (R2),R3 ;GET THE WAS DATA  
ERROR 1 ;BISB DELIVERED THE WRONG RESULT  
  
*****  
*TEST 573 BISB SM1,DM1 TEST - DEST ADDR ODD  
*****  
†T573:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #573,R0 ;LOAD R0 WITH TEST NUMBER  
MOV 2#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
MOV #177400,R4 ;RESULT S / B = 177400  
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1  
MOV #MBUFO+1,R3 ;ODD DEST ADDR IN R3  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
2$: BISB (R5),(R3) ;TEST THE BISB  
  
CMP R4,(R2) ;CORRECT RESULT  
BEQ T574 ;;BR IF YES  
  
3$: MOV (R2),R3 ;GET THE WAS DATA  
ERROR 1 ;BISB DELIVERED THE WRONG RESULT  
  
*****  
*TEST 574 JMP MODE 1 TEST, FLAGS = 1111  
*****  
†T574:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #574,R0 ;LOAD R0 WITH TEST NUMBER  
MOV 2#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #45,R2 ;R2 CONTAINS JUMP ADDRESS  
SCC ;MAKE N:C = 1111  
  
2$: JMP (R2) ;TEST THE JMP - GO TO 45
```

J01

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 193
JMP MODE 1 TEST, FLAGS = 1111

```

10576 040654 104006      3$:  ERROR 6           ;JMP FAILED TO LOAD PC
10577 040656 000405      BR      TST575        ;;GO CALL SCOPE
10578
10579 040660 103003      4$:  BCC 5$           ;BR IF JMP CLEARED "C"
10580 040662 102002      BVC 5$           ;BR IF JMP CLEARED "V"
10581 040664 001001      BNE 5$           ;BR IF JMP CLEARED "Z"
10582 040666 100401      BMI  TST575        ;;BR IF "N" STILL SET
10583
10584 040670 104006      5$:  ERROR 6           ;JMP ALTERED CODES - CLEARED ONE
10585
10586
10587
10588
10589 040672
10590 040672 000004
10591 040674 012700 000575
10592 040700 013701 040712
10593 040704 012702 040720
10594 040710 000257
10595
10596 040712 000112      2$:  JMP (R2)         ;TEST THE JMP - GO TO 4$
10597
10598 040714 104006      3$:  ERROR 6           ;JMP FAILED TO LOAD PC
10599 040716 000405      BR      TST576        ;;GO CALL SCOPE
10600
10601 040720 103403      4$:  BCS 5$           ;BR IF JMP SET "C"
10602 040722 102402      BVS 5$           ;BR IF JMP SET "V"
10603 040724 001401      BEQ 5$           ;BR IF JMP SET "Z"
10604 040726 100001      BPL  TST576        ;;BR IF "N" STILL CLEAR
10605
10606 040730 104006      5$:  ERROR 6           ;JMP ALTERED CODES - SET ONE
10607
10608
10609
10610
10611 040732
10612 040732 000004
10613 040734 012700 000576
10614 040740 013701 040752
10615 040744 012702 040760
10616 040750 000277
10617
10618 040752 000122      2$:  JMP (R2)+        ;TEST THE JMP - GO TO 4$
10619
10620 040754 104006      3$:  ERROR 6           ;JMP FAILED TO LOAD PC
10621 040756 000411      BR      TST577        ;;GO TO SCOPE EXIT
10622
10623 040760 103003      4$:  BCC 5$           ;BR IF JMP CLEARED "C"
10624 040762 102002      BVC 5$           ;BR IF JMP CLEARED "V"
10625 040764 001001      BNE 5$           ;BR IF JMP CLEARED "Z"
10626 040766 100401      BMI  6$           ;BR IF "N" STILL SET
10627
10628 040770 104006      5$:  ERROR 6           ;JMP ALTERED CODES - CLEARED
10629
10630 040772 022702 040762      6$:  CMP #4$+2,R2     ;DID R2 GET AUTO-INCREMENTED?
10631 040776 001401      BEQ  TST577        ;;BR IF YES

```

K01

```

10632
10633 041000 104006          7$:      ERROR      6          ;JMP FAILED TO UPDATE REGISTER (R2)
10634
10635
10636          ;:*****
10637          ;:TEST 577      JMP MODE 2 TEST; FLAGS = 0000
10638          ;:*****
10639          ;:TST577:
10639 041002 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10640 041004 012700 000577  MOV      #577,R0  ;:LOAD R0 WITH TEST NUMBER
10641 041010 013701 041022  MOV      @#2$,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
10642 041014 012702 041030  MOV      #4$,R2  ;:R2 CONTAINS JUMP ADDRESS
10643 041020 000257          CCC          ;MAKE N:C = 0000
10644
10645 041022 000122          2$:      JMP      (R2)+      ;TEST THE JMP - GO TO 4$
10646
10647 041024 104006          3$:      ERROR      6          ;JMP FAILED TO LOAD PC
10648 041026 000405          BR      TST600      ;;GO TO SCOPE EXIT
10649
10650 041030 103403          4$:      BCS      5$          ;BR IF JMP SET "C"
10651 041032 102402          BVS      5$          ;BR IF JMP SET "V"
10652 041034 001401          BEQ      5$          ;BR IF JMP SET "Z"
10653 041036 100001          BPL      TST600      ;;BR IF "N" IS CLEAR
10654
10655 041040 104006          5$:      ERROR      6          ;JMP ALTERED CODES - SET
10656
10657          ;:*****
10658          ;:TEST 600      JMP TEST MODE 3; FLAGS = 1111
10659          ;:*****
10660          ;:TST600:
10660 041042 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10661 041042 012700 000600  MOV      #600,R0  ;:LOAD R0 WITH TEST NUMBER
10662 041044 013701 041062  MOV      @#2$,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
10663 041050 013701 041062  MOV      #7$,R2  ;:R2 CONTAINS ADDRESS OF JUMP ADDRESS
10664 041054 012702 041114  SCC          ;SET N:C = 1111
10665 041060 000277
10666
10667 041062 000132          2$:      JMP      @ (R2)+      ;TEST THE JMP - GO TO 4$
10668
10669 041064 104006          3$:      ERROR      6          ;JMP FAILED TO LOAD PC
10670 041066 000414          BR      TST601      ;;GO TO SCOPE EXIT
10671
10672 041070 103003          4$:      BCC      5$          ;BR IF JMP CLEARED "C"
10673 041072 102002          BVC      5$          ;BR IF JMP CLEARED "V"
10674 041074 001001          BNE      5$          ;BR IF JMP CLEARED "Z"
10675 041076 100401          BMI      6$          ;BR IF "N" STILL SET
10676
10677 041100 104006          5$:      ERROR      6          ;JMP ALTERED CODES - CLEAR
10678
10679 041102 022702 041116          6$:      CMP      #7$+2,R2  ;:DID JMP UPDATE R2?
10680 041106 001404          BEQ      TST601      ;;BR IF YES
10681
10682 041110 104006          ERROR      6          ;JMP FAILED TO UPDATE REGISTER
10683 041112 000402          BR      TST601      ;:GO TO SCOPE EXIT
10684 041114 041070          7$:      4$          ;:JMP3 CONTAINS JUMP ADDRESS
10685 041116 104006          ERROR      6          ;:ERROR CALL OCCURS IF MODE3 HAPPENS
10686          ;:TO EXECUTE AS MODE 1 OR 2 AND
10687          ;:4$ IS LEGAL INSTRUCTION
  
```

LO1

10688
 10689
 10690
 10691
 10692 041120
 10693 041120 000004
 10694 041122 012700 000601
 10695 041126 013701 041140
 10696 041132 012702 041162
 10697 041136 000257
 10698
 10699 041140 000132
 10700
 10701 041142 104006
 10702 041144 000410
 10703
 10704 041146 103403
 10705 041150 102402
 10706 041152 001401
 10707 041154 100004
 10708
 10709 041156 104006
 10710 041160 000402
 10711
 10712 041162 041146
 10713 041164 104006
 10714
 10715
 10716
 10717
 10718 041166
 10719 041166 000004
 10720 041170 012700 000602
 10721 041174 013701 041206
 10722 041200 012702 041216
 10723 041204 000277
 10724
 10725 041206 000142
 10726
 10727 041210 104006
 10728 041212 000414
 10729
 10730 041214 000402
 10731 041216 104006
 10732 041220 000411
 10733
 10734 041222 103003
 10735 041224 102002
 10736 041226 001001
 10737 041230 100401
 10738
 10739 041232 104006
 10740
 10741 041234 022702 041214
 10742 041240 001401
 10743

```

*****
;TEST 601      JMP TEST MODE 3; FLAGS = 0000
*****
TST601:
SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV            #601,R0      ;LOAD R0 WITH TEST NUMBER
MOV            2(R0),R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV            #65,R2      ;R2 CONTAINS ADDRESS OF JUMP ADDRESS
CCC            ;MAKE N:C = 0000

2$:  JMP        2(R2)+      ;TEST THE JMP - GO TO 4$

3$:  ERROR     6           ;JMP FAILED TO LOAD THE PC
BR    TST602      ;;GO TO SCOPE EXIT

4$:  BCS       5$          ;BR IF JMP SET "C"
     BVS       5$          ;BR IF JMP SET "V"
     BEQ       5$          ;BR IF JMP SET "Z"
     BPL       TST602     ;;BR IF "N" STILL CLEAR

5$:  ERROR     6           ;JMP ALTERED CODES - SET
BR    TST602      ;;GO TO SCOPE EXIT

6$:  4$        6           ;JUMP ADDRESS IN 6$
     ERROR     6           ;JMP MODE 3 EXECUTED LIKE MODE 1 OR 2

*****
;TEST 602      JMP TEST MODE 4; FLAGS = 1111
*****
TST602:
SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV            #602,R0      ;LOAD R0 WITH TEST NUMBER
MOV            2(R0),R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV            #55,R2      ;[R2] = JMP ADDRESS PLUS 2
SCC            ;MAKE N:C = 1111

2$:  JMP        -(R2)      ;TEST THE JMP - GO TO 5$ MINUS 2

3$:  ERROR     6           ;JMP FAILED TO LOAD PC
BR    TST603      ;;GO TO SCOPE EXIT

5$:  BR        4$          ;GO TEST FLAGS - JMP LOADED PC OK
     ERROR     6           ;JMP FAILED TO AUTO-DECREMENT R2
BR    TST603      ;;GO TO SCOPE EXIT

4$:  BCC       7$          ;BR IF JMP CLEARED "C"
     BVC       7$          ;BR IF JMP CLEARED "V"
     BNE       7$          ;BR IF JMP CLEARED "Z"
     BMI       6$          ;BR IF "N" STILL SET

7$:  ERROR     6           ;JMP ALTERED FLAGS

6$:  CMP        #55-2,R2   ;DID JMP UPDATE R2 PROPERLY?
     BEQ        TST603    ;;BR IF YES
    
```

MO1

```

10744 041242 104006          9S:   ERROR   6           ;JMP FAILED TO UPDATE REGISTER
10745
10746
10747          ;:*****
10748          ;*TEST 603   JMP TEST MODE 4; FLAGS = 0000
10749          ;:*****
10749 041244          TST603:
10750 041244 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10751 041246 012700 000603  MOV      #603,R0  ;:LOAD R0 WITH TEST NUMBER
10752 041252 013701 041264  MOV      2#25,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
10753 041256 J12702 041274  MOV      #45+2,R2 ;:[R2] = JUMP ADDRESS PLUS 2
10754 041262 000257          CCC          ;:MAKE N:C = 0000
10755
10756 041264 000142          2S:   JMP      -(R2)      ;:TEST THE JMP - TO TO 45
10757
10758 041266 104006          3S:   ERROR   6           ;:JMP FAILED TO LOAD PC
10759 041270 000405          BR      TST604      ;:GO TO SCOPE EXIT
10760
10761 041272 103403          4S:   BCS      55          ;:BR IF JMP SET "C"
10762 041274 102402          BVS      55          ;:BR IF JMP SET "V"
10763 041276 001401          BEQ      55          ;:BR IF JMP SET "Z"
10764 041300 100001          BPL      TST604      ;:BR IF "N" STILL CLEAR
10765
10766 041302 104006          5S:   ERROR   6           ;:JMP ALTERED CODES - SET
10767
10768          ;:*****
10769          ;*TEST 604   JMP TEST MODE 5; FLAGS = 1111
10770          ;:*****
10771 041304          TST604:
10772 041304 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10773 041306 012700 000604  MOV      #604,R0  ;:LOAD R0 WITH TEST NUMBER
10774 041312 013701 041324  MOV      2#25,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
10775 041316 012702 041360  MOV      #JMP5,R2 ;:JMP CONTAINS ADDR+2 OF JUMP ADDRESS
10776 041322 000277          SCC
10777
10778 041324 000152          2S:   JMP      2-(R2)      ;:TEST THE JMP - GO TO 45
10779
10780 041326 104006          3S:   ERROR   6           ;:JMP FAILED TO LOAD PC
10781 041330 000414          BR      TST605      ;:GO TO SCOPE OXIT
10782
10783 041332 103003          4S:   BCC      55          ;:BR IF JMP CLEARED "C"
10784 041334 102002          BVC      55
10785 041336 001001          BNE      55
10786 041340 100401          BMI      65
10787
10788 041342 104006          5S:   ERROR   6           ;:JMP ALTERED CODES - CLEARED
10789
10790 041344 022702 041356          6S:   CMP      #JMP5-2,R2 ;:DID R2 GET AUTO-DECREMENTED
10791 041350 001404          BEQ      TST605      ;:BR IF YES
10792
10793 041352 104006          7S:   ERROR   6           ;:JMP FAILED TO UPDATE REGISTER
10794 041354 000402          BR      TST605      ;:GO TO SCOPE EXIT
10795 041356 041332          4S
10796 041360 104006          JMP5:  ERROR   6           ;:THIS LOCATION CONTAINS JMP ADDRESS
10797          ;:JMP EXECUTED LIKE A MODE 1 OR 2
10798          ;:*****
10799          ;*TEST 605   JMP TEST MODE 5; FLAG = 0000
    
```

```

10800
10801 041362
10802 041362 000004
10803 041364 012700 000605
10804 041370 013701 041402
10805 041374 012702 041426
10806 041400 000257
10807
10808 041402 000152
10809
10810 041404 104006
10811 041406 000410
10812
10813 041410 103403
10814 041412 102402
10815 041414 001401
10816 041416 100004
10817
10818 041420 104006
10819 041422 000402
10820
10821 041424 041410
10822 041426 104006
10823
10824
10825
10826
10827 041430
10828 041430 000004
10829 041432 012700 000606
10830 041436 013701 041450
10831 041442 012702 041474
10832 041446 000277
10833
10834 041450 000162 177764
10835
10836 041454 104006
10837 041456 000407
10838
10839 041460 103003
10840 041462 102002
10841 041464 001001
10842 041466 100403
10843
10844 041470 104006
10845 041472 000401
10846
10847 041474 104006
10848
10849
10850
10851
10852
10853 041476
10854 041476 000004
10855 041500 012700 000607
    
```

```

*****
TST605:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #605,R0                      ;LOAD R0 WITH TEST NUMBER
MOV      2#25,R1                      ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #JMPSA,R2                    ;[R2] = ADDR +2 OF JUMP ADDRESS
CCC                                     ;SET N:C = 0000

25:     JMP      2-(R2)                ;TEST THE JMP - GO TO 45

35:     ERROR   6
BR       TST606                      ;JMP FAILED TO LOAD PC
;;GO TO SCOPE EXIT

45:     BCS     5$
BVS     5$
BEQ     5$
BPL     TST606                      ;BR IF "N" STILL CLEAR

55:     ERROR   6
BR       TST606                      ;JMP ALTERED THE CODES - SET
;;GO TO SCOPE EXIT

JMPSA:  45
ERROR   6                            ;THIS LOCATION CONTAINS JUMP ADDRESS
;;JMP EXECUTED LIKE A MODE 1 OR 2

*****
;#TEST 606      JMP TEST MODE 6; FLAGS = 1111
*****
TST606:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #606,R0                      ;LOAD R0 WITH TEST NUMBER
MOV      2#25,R1                      ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #75,R2 ;[R2] = BASE ADDRESS TO BE INDEXED
SCC                                     ;MAKE N:C = 1111

25:     JMP      45-75(R2)            ;TEST THE JMP - GO TO 45

35:     ERROR   6
BR       TST607                      ;JMP FAILED TO LOAD THE PC
;;GO TO SCOPE EXIT

45:     BCC     5$
BVC     5$
BNE     5$
BMI     TST607                      ;BR IF "N" STILL SET

55:     ERROR   6
BR       TST607                      ;JMP ALTERED CODES - CLEARED
;;GO TO SCOPE EXIT

75:     ERROR   6
;;JMP EXECUTED LIKE A MODE 1 OR 2 OR
;;FAILED TO INDEX [R2]

*****
;#TEST 607      JMP TEST MODE 6; FLAGS = 0000
*****
TST607:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #607,R0                      ;LOAD R0 WITH TEST NUMBER
    
```

```

10856 041504 013701 041516      MOV      2#2$ R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
10857 041510 012702 041542      MOV      #7$,R2     ;[R2] = BASE ADDRESS FOR JUMP
10858 041514 000257                CCC                ;MAKE N:C = 0000
10859
10860 041516 000162 177764      2$:      JMP      4$-7$(R2)      ;TEST THE JMP - GO TO 4$
10861
10862 041522 104006                3$:      ERROR    6                ;JMP FAILED TO LOAD PC
10863 041524 000407                BR      TST610          ;;GO TO SCOPE EXIT
10864
10865 041526 103403                4$:      BCS      5$                ;BR IF JMP SET "C"
10866 041530 102402                BVS     5$                ;BR IF JMP SET "V"
10867 041532 001401                BEQ     5$                ;BR IF JMP SET "Z"
10868 041534 100003                BPL     TST610          ;;BR IF "N" STILL CLEAR
10869
10870 041536 104006                5$:      ERROR    6                ;JMP ALTERED CODES
10871 041540 000401                BR      TST610          ;;GO TO SCOPE EXIT
10872
10873 041542 104006                7$:      ERROR    6                ;JMP EXECUTED LIKE A MODE 1 OR 2, OR
10874                                ;FAILED TO INDEX [R2]
10875
10876                                ;*****
10877                                ;*TEST 610      JMP TEST MODE 7; FLAGS = 1111
10878                                ;*****
10879                                †TST610:
10880 041544 000004                SCOPE                ;CALL THE SCOPE LOOP UTILITY
10881 041546 012700 000610      MOV      #610,R0      ;LOAD R0 WITH TEST NUMBER
10882 041552 013701 041564      MOV      2#2$ R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
10883 041556 012702 041574      MOV      #5$,R2     ;[R2] = BASE ADDRESS
10884 041562 000277                SCC                ;MAKE N:C = 1111
10885
10886 041564 000172 000020      2$:      JMP      28$-5$(R2)      ;TEST THE JMP - GO TO 4$
10887
10888 041570 104006                3$:      ERROR    6                ;JMP FAILED TO LOAD PC
10889 041572 000412                BR      TST611          ;;GO TO SCOPE EXIT
10890
10891 041574 104006                5$:      ERROR    6                ;JMP FAILED TO INDEX OR ACTED LIKE MODE 1 OR 2
10892 041576 000410                BR      TST611          ;;GO TO SCOPE EXIT
10893
10894 041600 103003                4$:      BCC      7$                ;BR IF JMP CLEARED "C"
10895 041602 102002                BVC     7$                ;BR IF JMP CLEARED "V"
10896 041604 001001                BNE     7$                ;BR IF JMP CLEARED "Z"
10897 041606 100404                BMI     TST611          ;;BR IF "N" STILL SET
10898
10899 041610 104006                7$:      ERROR    6                ;JMP ALTERED CODES - CLEARED
10900 041612 000402                BR      TST611          ;;GO TO SCOPE EXIT
10901
10902 041614 041600                8$:      4$                ;THIS LOCATION CONTAINS JMP ADDRESS
10903
10904 041616 104006                ERROR    6                ;JMP EXECUTED LIKE MODE 6
10905
10906                                ;*****
10907                                ;*TEST 611      JMP TEST MODE 7; FLAGS = 0000
10908                                ;*****
10909                                †TST611:
10910 041620 000004                SCOPE                ;CALL THE SCOPE LOOP UTILITY
10911 041622 012700 000611      MOV      #611,R0      ;LOAD R0 WITH TEST NUMBER

```

```

10912 041626 013701 041640      MOV    2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
10913 041632 012702 041650      MOV    #5$,R2      ;[R2] = BASE ADDRESS
10914 041636 000257              CCC                ;MAKE N:C = 0000
10915
10916 041640 000172 000020      2$:   JMP    28$-5$(R2) ;TEST THE JMP - GO TO 4$
10917
10918 041644 104006              3$:   ERROR 6        ;JMP FAILED TO LOAD PC
10919 041646 000412              BR     TST612      ;;GO TO SCOPE EXIT
10920
10921 041650 104006              5$:   ERROR 6        ;JMP FAILED TO INDEX
10922 041652 000410              BR     TST612      ;;GO TO SCOPE EXIT
10923
10924 041654 103403              4$:   BCS    7$        ;BR IF JMP SET "C"
10925 041656 102402              BVS    7$        ;BR IF JMP SET "V"
10926 041660 001401              BEQ    7$        ;BR IF JMP SET "Z"
10927 041662 100004              BPL    TST612     ;;BR IF "N" STILL CLEAR
10928
10929 041664 104006              7$:   ERROR 6        ;JMP ALTERED CODES - SET
10930 041666 000402              BR     TST612     ;;GO TO SCOPE EXIT
10931
10932 041670 041654              8$:   4$          ;THIS LOCATION CONTAINS JUMP ADDRESS
10933
10934 041672 104006              ERROR 6          ;JMP EXECUTED LIKE A MODE 6
10935

```

```

;*****
;#TEST 612      JSR MODE 1 TEST - LOAD PC / PUSH SP
;*****

```

```

10938
10939 041674
10940 041674 000004              TST612:          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10941 041676 012700 000612      MOV    #612,R0     ;LOAD R0 WITH TEST NUMBER
10942 041702 013701 041724      MOV    2#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
10943 041706 010605              MOV    SP,R5       ;SAVE THE SP
10944 041710 010737 001010      MOV    PC,2#SLPERR ;SET ERROR LOOP ADDRESS
10945 041714 010506              1$:   MOV    R5,SP      ;RESTORE SP FOR ERROR LOOPING
10946 041716 012702 041730      MOV    #4$,R2     ;DEST ADDR = 4$
10947 041722 000257              CCC                ;SCOPE SYNC
10948
10949 041724 004412              2$:   JSR    R4,(R2) ;TEST THE JSR - GO TO 4$
10950
10951 041726 104006              3$:   ERROR 6        ;JSR FAILED TO LOAD THE PC
10952
10953 041730 005726              4$:   TST    (SP)+    ;POP THE SP
10954 041732 020605              CMP    SP,R5       ;DID JSR PUSH THE SP ?
10955 041734 001406              BEQ    TST613     ;;BR IF YES
10956
10957 041736 005746              TST    -(SP)       ;RESTORE ERROR SP
10958 041740 010603              MOV    SP,R3       ;[R3]= WAS SP
10959 041742 010504              MOV    R5,R4
10960 041744 005744              TST    -(R4)
10961 041746 104003              5$:   ERROR 3        ;[R4]= S/B SP
10962
10963 041750 010506              MOV    R5,SP      ;RESTORE SP IN CASE OF ERROR

```

```

;*****
;#TEST 613      JSR MODE 1 TEST - CHECK RN AND OLD PC
;*****
TST613:

```

10967 041752


```

10968 041752 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
10969 041754 012700 000613  MOV      #613,R0      ;LOAD R0 WITH TEST NUMBER
10970 041760 013701 042012  MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
10971 041764 010605          MOV      SP,R5        ;SAVE THE SP
10972 041766 010737 001010  MOV      PC,@$SLPERR  ;SET ERROR LOOP ADDRESS
10973 041772 010506          1$: MOV      R5,SP        ;RESTORE SP FOR ERROR LOOPING
10974 041774 012702 042016  MOV      #4$,R2       ;DEST ADDR = 4$
10975 042000 005066 177776  CLR      -(SP)        ;INIT STACK LOC TO GET [R4]
10976 042004 012704 125252  MOV      #125252,R4   ;INIT RN = 125252
10977 042010 000257          CCC                   ;SCOPE SYNC
10978
10979 042012 004412          2$: JSR      R4,(R2)   ;TEST THE JSR - GO TO 4$
10980
10981 042014 104006          3$: ERROR   6         ;JSR FAILED TO LOAD THE PC
10982
10983 042016 022726 125252  4$: CMP      #125252,(SP)+ ;DID JSR SAVE REG ON STACK
10984 042022 001401          BEQ      8$         ;BR IF IT DID
10985
10986 042024 104005          5$: ERROR   5         ;JSR FAILED TO SAVE REG ON STACK
10987
10988 042026 022704 042014  8$: CMP      #3$,R4     ;DID OLD PC GET SAVED ?
10989 042032 001401          BEQ      6$         ;BR IF YES
10990
10991 042034 104005          7$: ERROR   5         ;JSR FAILED TO SAVE TH OLD PC
10992
10993 042036 010506          6$: MOV      R5,SP     ;RESTORE SP IN CASE ERROR SCREWED IT UP
10994
10995 ;*****
10996 ;*TEST 614 JSR MODE 1 TEST - N:C = 0000
10997 ;*****
10998 042040          †ST614:
10999 042040 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11000 042042 012700 000614  MOV      #614,R0      ;LOAD R0 WITH TEST NUMBER
11001 042046 013701 042102  MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
11002          .SBTTL USER CONTROLLED BREAKPOINT -- BIT12
11003 042052 032737 010000 063160  BIT      #BIT12,@#BPTLOC ;BREAKPOINT HALT SET ??
11004 042060 001401          BEQ      .+4         ;BR IF NOT
11005 042062 000000          HALT          ;BREAK-DEPRESS CONTINUE TO CONTINUE
11006 042064 010605          MOV      SP,R5        ;SAVE THE SP
11007 042066 010737 001010  MOV      PC,@$SLPERR  ;SET ERROR LOOP ADDRESS
11008 042072 010506          1$: MOV      R5,SP        ;RESTORE SP FOR ERROR LOOPING
11009 042074 012702 042106  MOV      #4$,R2       ;DEST ADDR = 4$
11010 042100 000257          CCC                   ;N:C = 0000
11011
11012 042102 004412          2$: JSR      R4,(R2)   ;TEST THE JSR - GO TO 4$
11013
11014 042104 104006          3$: ERROR   6         ;JSR FAILED TO LOAD THE PC
11015
11016 042106 100403          4$: BMI      5$         ;N:C = 0000 ?
11017 042110 001402          BEQ      5$
11018 042112 102401          BVS      5$
11019 042114 103001          BCC      6$
11020
11021 042116 104005          5$: ERROR   5         ;JSR FAILED - ALTERED FLAGS
11022
11023 042120 010506          6$: MOV      R5,SP     ;RESET SP IN CASE OF ERROR

```

E02

11024
11025
11026
11027 042122
11028 042122 000004
11029 042124 012700 000615
11030 042130 013701 042152
11031 042134 010605
11032 042136 010737 001010
11033 042142 010506
11034 042144 012702 042156
11035 042150 000277
11036
11037 042152 004412
11038
11039 042154 104006
11040
11041 042156 100003
11042 042160 001002
11043 042162 102001
11044 042164 103401
11045 042166 104005
11046
11047 042170 010506
11048
11049
11050
11051
11052 042172
11053 042172 000004
11054 042174 012700 000616
11055 042200 013701 042222
11056 042204 010605
11057 042206 010737 001010
11058 042212 010506
11059 042214 012702 042226
11060 042220 000257
11061
11062 042222 004422
11063
11064 042224 104006
11065
11066 042226 005726
11067 042230 020605
11068 042232 001406
11069
11070 042234 005746
11071 042236 010603
11072 042240 010504
11073 042242 005744
11074 042244 104003
11075
11076 042246 010506
11077
11078
11079

```
*****
;TEST 615 JSR MODE 1 TEST - N:C = 1111
*****
†ST615:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #615,R0 ;LOAD R0 WITH TEST NUMBER
MOV @R2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #4$,R2 ;DEST ADDR = 4$
SCC ;N:C = 1111

2$: JSR R4,(R2) ;TEST THE JSR - GO TO 4$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: BPL 5$ ;N:C = 1111 ?
BNE 5$
BVC 5$
BCS 6$

5$: ERROR 5 ;JSR ALTERED FLAGS

6$: MOV R5,SP ;RESET SP IN CASE OF ERROR

*****
;TEST 616 JSR MODE 2 TEST
*****
†ST616:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #616,R0 ;LOAD R0 WITH TEST NUMBER
MOV @R2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #4$,R2 ;DEST ADDR = 4$
CCC ;SCOPE SYNC

2$: JSR R4,(R2)+ ;TEST THE JSR - GO TO 4$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ TST617 ;BR IF YES

TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
5$: ERROR 3 ;JSR FAILED TO PUSH SP

MOV R5,SP ;RESTORE SP JUST IN CASE

*****
;TEST 617 JSR MODE 3 TEST
*****
```

```

11080
11081 042250
11082 042250 000004
11083 042252 012700 000617
11084 042256 013701 042300
11085 042262 010605
11086 042264 010737 001010
11087 042270 010506
11088 042272 012702 042326
11089 042276 000257
11090
11091 042300 004432
11092
11093 042302 104006
11094
11095 042304 005726
11096 042306 020605
11097 042310 001411
11098
11099 042312 005746
11100 042314 010603
11101 042316 010504
11102 042320 005744
11103 042322 104003
11104 042324 000402
11105
11106 042326 042304
11107 042330 104006
11108
11109 042332 010506
11110
11111
11112
11113
11114 042334
11115 042334 000004
11116 042336 012700 000620
11117 042342 013701 042364
11118 042346 010605
11119 042350 010737 001010
11120 042354 010506
11121 042356 012702 042372
11122 042362 000257
11123
11124 042364 004442
11125
11126 042366 104006
11127
11128 042370 000401
11129 042372 104005
11130
11131 042374 005726
11132 042376 020605
11133 042400 001406
11134
11135 042402 005746

;*****
†T617:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #617,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #7$,R2 ;DEST ADDR = [7$]
CCC ;SCOPE SYNC

2$: JSR R4,@(R2)+ ;TEST THE JSR - GO TO 4$ VIA 7$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T620 ;BR IF YES

TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
5$: ERROR 3 ;JSR FAILED
BR 6$ ;GO EXIT

7$: 4$ ;CONTAINS JUMP ADDR
ERROR 6 ;JSR EXECUTED LIKE A MODE 1 OR 2

6$: MOV R5,SP ;RESTORE SP JUST IN CASE

;*****
;*TEST 620 JSR MODE 4 TEST
;*****
†T620:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #620,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #5$,R2 ;DEST ADDR = 4$+2
CCC ;SCOPE SYNC

2$: JSR R4,-(R2) ;TEST THE JSR - GO TO 4$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: BR 6$ ;JUMPED OK - GO CHECK SP
5$: ERROR 5 ;JSR FAILED TO DECREMENT DEST REG

6$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T621 ;BR IF YES

TST -(SP) ;RESET SP TO ERROR VALUE
    
```

```

11136 042404 010603          MOV     SP,R3          ;WAS SP
11137 042406 010504          MOV     RS,R4
11138 042410 005744          TST     -(R4)         ;S/B SP
11139 042412 104003          7$:    ERROR     3          ;JSR FAILED TO PUSH SP
11140
11141 042414 010506          8$:    MOV     RS,SP          ;RESTORE SP JUST IN CASE
11142
11143          ;:*****
11144          ;*TEST 621      JSR MODE 5 TEST
11145          ;:*****
11146          †T621:
11146 042416          SCOPE
11147 042416 000004          MOV     #621,R0        ;CALL THE SCOPE LOOP UTILITY
11148 042420 012700 000621          MOV     @#2$,R1        ;LOAD R0 WITH TEST NUMBER
11149 042424 013701 042446          MOV     SP,R5          ;LOAD R1 WITH TEST INSTRUCTION WORD
11150 042430 010605          MOV     PC,@#SLPERR    ;SAVE THE SP
11151 042432 010737 001010          MOV     RS,SP          ;SET ERROR LOOP ADDRESS
11152 042436 010506          1$:    MOV     #7$,R2        ;RESET SP FOR ERROR LOOPS
11153 042440 012702 042476          MOV     CCC            ;DEST ADDR = [7$ - 2]
11154 042444 000257          CCC
11155
11156 042446 004452          2$:    JSR     R4,@-(R2)    ;TEST THE JSR - GO TO 4$
11157
11158 042450 104006          3$:    ERROR     6          ;JSR FAILED TO LOAD THE PC
11159
11160 042452 005726          4$:    TST     (SP)+        ;RESET SP
11161 042454 020605          CMP     SP,R5          ;DID JSR PUSH STACK ?
11162 042456 001411          BEQ     TST622         ;BR IF YES
11163
11164 042460 005746          TST     -(SP)         ;RESET SP TO ERROR VALUE
11165 042462 010603          MOV     SP,R3          ;WAS SP
11166 042464 010504          MOV     RS,R4
11167 042466 005744          TST     -(R4)         ;S/B SP
11168 042470 104003          5$:    ERROR     3          ;JSR FAILED TO PUSH SP
11169 042472 000402          BR      6$            ;GO EXIT
11170
11171 042474 042452          4$
11172 042476 104005          7$:    ERROR     5          ;CONTAINS JUMP ADDRESS
11173
11174 042500 010506          6$:    MOV     RS,SP          ;JSR EXECUTED LIKE A MODE. 1 OR 2
11175
11176          ;:*****
11177          ;*TEST 622      JSR MODE 6 TEST
11178          ;:*****
11179          †T622:
11179 042502          SCOPE
11180 042502 000004          MOV     #622,R0        ;CALL THE SCOPE LOOP UTILITY
11181 042504 012700 000622          MOV     @#2$,R1        ;LOAD R0 WITH TEST NUMBER
11182 042510 013701 042532          MOV     SP,R5          ;LOAD R1 WITH TEST INSTRUCTION WORD
11183 042514 010605          MOV     PC,@#SLPERR    ;SAVE THE SP
11184 042516 010737 001010          MOV     RS,SP          ;SET ERROR LOOP ADDRESS
11185 042522 010506          1$:    MOV     #3$,R2        ;RESET SP FOR ERROR LOOPS
11186 042524 012702 042536          MOV     CCC            ;[R2] = BASE DEST ADDR
11187 042530 000257          CCC
11188
11189 042532 004462 000002          2$:    JSR     R4,4$-3$(R2) ;TEST THE JSR - GO TO 4$
11190
11191 042536 104006          3$:    ERROR     6          ;JSR FAILED TO LOAD THE PC OR INDEX FAILED
    
```

11192
11193 042540 005726
11194 042542 020605
11195 042544 001406
11196
11197 042546 005746
11198 042550 010603
11199 042552 010504
11200 042554 005744
11201 042556 104003
11202 042560 010506
11203
11204
11205
11206
11207 042562
11208 042562 000004
11209 042564 012700 000623
11210 042570 013701 042612
11211 042574 010605
11212 042576 010737 001010
11213 042602 010506
11214 042604 012702 042616
11215 042610 000257
11216
11217 042612 004472 000024
11218
11219 042616 104006
11220
11221
11222 042620 005726
11223 042622 020605
11224 042624 001411
11225
11226 042626 005746
11227 042630 010603
11228 042632 010504
11229 042634 005744
11230 042636 104003
11231 042640 000402
11232
11233 042642 042620
11234 042644 104005
11235
11236 042646 010506
11237
11238
11239
11240
11241 042650
11242 042650 000004
11243 042652 012700 000624
11244 042656 013701 042676
11245 042662 012702 000001
11246 042666 000402
11247

```

4$:  TST      (SP)+      ;RESET SP
      CMP      SP,R5     ;DID JSR PUSH STACK ?
      BEQ      TST623    ;;BR IF YES

      TST      -(SP)     ;RESET SP TO ERROR VALUE
      MOV      SP,R3     ;WAS SP
      MOV      R5,R4
      TST      -(R4)     ;S/B SP
5$:  ERROR    3          ;JSR FAILED TO PUSH STACK
      MOV      R5,SP     ;RESET SP JUST IN CASE

;*****
;*TEST 623      JSR MODE 7 TEST
;*****
TST623:
      SCOPE
      MOV      #623,R0   ;CALL THE SCOPE LOOP UTILITY
      MOV      @#2$,R1  ;LOAD R0 WITH TEST NUMBER
      MOV      SP,R5     ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      PC,@#SLPERR ;SAVE THE SP
      MOV      R5,SP    ;SET ERROR LOOP ADDRESS
1$:  MOV      #3$,R2    ;RESET SP FOR ERROR LOOPS
      CCC
      ;BASE DEST ADDR = 3$
      ;SCOPE SYNC

2$:  JSR      R4,@7$-3$(R2) ;TEST THE JSR - GO TO 4$ VIA 7$
3$:  ERROR    6          ;JSR FAILED TO LOAD THE PC
      ;OR THE INDEX FAILED

4$:  TST      (SP)+      ;RESET SP
      CMP      SP,R5     ;DID JSR PUSH STACK ?
      BEQ      TST624    ;;BR IF YES

      TST      -(SP)     ;RESET SP TO ERROR VALUE
      MOV      SP,R3     ;WAS SP
      MOV      R5,R4
      TST      -(R4)     ;S/B SP
5$:  ERROR    3          ;JSR FAILED TO PUSH STACK
      BR       6$        ;SKIP TO EXIT

7$:  4$
      ERROR    5          ;CONTAINS JUMP ADDR
      ;JSR WORKED LIKE A MODE 1 OR 2

6$:  MOV      R5,SP     ;RESTORE SP JUST IN CASE

;*****
;*TEST 624      SOB TEST, [R] = 1, NO BRANCH
;*****
TST624:
      SCOPE
      MOV      #624,R0   ;CALL THE SCOPE LOOP UTILITY
      MOV      @#2$,R1  ;LOAD R0 WITH TEST NUMBER
      MOV      #1,R2    ;LOAD R1 WITH TEST INSTRUCTION WORD
      BR       2$-2     ;SET SOB COUNTER = 1
      ;GO DO THE SOB

```

```

11248 042670 104006      3$:  ERROR 6          ;SOB SHOULDN'T HAVE BRANCHED HERE
11249 042672 000402      BR      TST625      ;;GO TO SCOPE CALL
11250
11251 042674 000257      CCC          ;SYNC INSTR.
11252 042676 077204      2$:  SOB      R2,3$  ;TEST THE SOB
11253
11254
11255      ;*****
11255      ;*TEST 625      SOB TEST, (R) = 5, BRANCH 4 TIMES
11255      ;*****
11256
11257 042700      TST625:
11258 042700 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
11259 042702 012700 000625      MOV      #625,R0 ;:LOAD R0 WITH TEST NUMBER
11260 042706 013701 042740      MOV      @#SOB2,R1 ;:GET COPY OF TEST INSTRUCTION WORD
11261 042712 012702 000005      MOV      #5,R2    ;:SET SOB COUNTER = 5
11262 042716 012705 177773      MOV      #-5,R5   ;:SET UP R5 TO COUNT 5 BRANCHES
11263 042722 000405      BR      SOB2-2    ;:GO DO THE SOB
11264
11265 042724 000474      SOB1:  BR      SOB3 ;USED BY LAST SOB TEST TO TEST MAX OFFSET
11266 042726 000240      NOP          ;OFFSET ADJUSTMENT
11267 042730 000240      NOP
11268
11269 042732 005205      SOB5:  INC      R5   ;COUNT ONE BRANCH
11270 042734 001406      BEQ     SOBERR    ;BR IF TOO MANY LOOPS BY SOB
11271
11272 042736 000257      SOB2:  CCC          ;SCOPE SYNC
11273 042740 077204      SOB      R2,SOB5  ;TEST THE SOB
11274 042742 005702      TST     R2        ;R2 SHOULD CONTAIN 0
11275 042744 001403      BEQ     TST626    ;;BR IF IT DOES
11276
11277 042746 104006      ERROR 6          ;SOB COUNTER NOT ZERO
11278 042750 000401      BR      TST626    ;:GO TO SCOPE CALL
11279 042752 104006      SOBERR: ERROR 6   ;SOB MADE TOO MANY BRANCHES
11280
11281
11282      ;*****
11282      ;*TEST 626      SOB TEST, (R) = 1, FLAGS = 1111
11282      ;*****
11283
11284 042754      TST626:
11285 042754 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
11286 042756 012700 000626      MOV      #626,R0 ;:LOAD R0 WITH TEST NUMBER
11287 042762 013701 042774      MOV      @#25,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
11288 042766 012702 000001      MOV      #1,R2   ;:SET SOB COUNTER = 1
11289 042772 000277      SCC          ;MAKE N:C = 1111
11290
11291 042774 077202      2$:  SOB      R2,2$-2 ;TEST THE SOB
11292
11293 042776 103003      BCC     3$       ;BR IF C = 0
11294 043000 102002      BVC     3$       ;BR IF V = 0
11295 043002 001001      BNE     3$       ;BR IF Z = 0
11296 043004 100401      BMI     TST627   ;;BR IF N = 1
11297
11298 043006 104006      3$:  ERROR 6          ;SOB ALTERED CODES - CLEARED ONE
11299
11300
11301      ;*****
11301      ;*TEST 627      SOB TEST, (R) = 1, FLAGS = 0000
11301      ;*****
11302
11303 043010      TST627:

```

```

11304 043010 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11305 043012 012700 000627  MOV      #627,R0      ;LOAD RO WITH TEST NUMBER
11306 043016 013701 043030  MOV      @#25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11307 043022 012702 000001  MOV      #1,R2       ;SET SOB COUNTER = 1
11308 043026 000257          CCC              ;MAKE N:C = 0000
11309
11310 043030 077202          2$: SOB      R2,25-2    ;TEST THE SOB
11311
11312 043032 103403          BCS      3$          ;BR IF C = 1
11313 043034 102402          BVS      3$          ;BR IF V = 1
11314 043036 001401          BEQ      3$          ;BR IF Z = 1
11315 043040 100001          BPL      TST630     ;BR IF N = 0
11316
11317 043042 104006          3$: ERROR   6          ;SOB ALTERED CODES - SET ONE
11318
11319 ;*****
11320 ;*TEST 630 SOB TEST, [R] = 5, FLAGS = 1111
11321 ;*****
11322 †TST630:
11323 043044 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11324 043046 012700 000630  MOV      #630,R0      ;LOAD RO WITH TEST NUMBER
11325 043052 013701 043064  MOV      @#25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11326 043056 012702 000005  MOV      #5,R2       ;SET SOB COUNTER = 5
11327 043062 000277          SCC              ;MAKE N:C = 1111
11328
11329 043064 077201          2$: SOB      R2,25    ;TEST THE SOB
11330
11331 043066 103003          BCC      3$          ;BR IF C = 0
11332 043070 102002          BVC      3$          ;BR IF V = 0
11333 043072 001001          BNE      3$          ;BR IF Z = 0
11334 043074 100401          BMI      TST631     ;BR IF N = 1
11335
11336 043076 104006          3$: ERROR   6          ;SOB ALTERED CODES - CLEARED ONE
11337
11338 ;*****
11339 ;*TEST 631 SOB TEST, [R] = 5, FLAGS = 0000
11340 ;*****
11341 †TST631:
11342 043100 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11343 043102 012700 000631  MOV      #631,R0      ;LOAD RO WITH TEST NUMBER
11344 043106 013701 043120  MOV      @#SOB4,R1    ;GET COPY OF TEST INSTRUCTION WORD
11345 043112 012702 000005  MOV      #5,R2       ;SET SOB COUNTER = 5
11346 043116 000257          SOB3: CCC        ;MAKE N:C = 0000
11347
11348 043120 077277          SOB4: SOB      R2,SOB1 ;TEST THE SOB
11349
11350 043122 103403          BCS      3$          ;BR IF C = 1
11351 043124 102402          BVS      3$          ;BR IF V = 1
11352 043126 001401          BEQ      3$          ;BR IF Z = 1
11353 043130 100001          BPL      TST632     ;BR IF N = 0
11354
11355 043132 104006          3$: ERROR   6          ;SOB ALTERED CODES - SET ONE
11356
11357 ;*****
11358 ;*TEST 632 RTS TEST - N:C = 0000
11359 ;*****

```

11360 043134
11361 043134 000004
11362 043136 012700 000632
11363 043142 013701 043174
11364 043146 010605
11365 043150 010737 001010
11366 043154 012704 177777
11367 043160 010506
11368 043162 012703 043202
11369 043166 012746 177777
11370 043172 000257
11371
11372 043174 000203
11373
11374 043176 104005
11375 043200 000415
11376
11377 043202 100403
11378 043204 001402
11379 043206 102401
11380 043210 103001
11381
11382 043212 104005
11383
11384 043214 020403
11385 043216 001401
11386
11387 043220 104002
11388
11389 043222 020506
11390 043224 001404
11391
11392 043226 010504
11393 043230 010603
11394 043232 104003
11395
11396 043234 010506
11397
11398
11399
11400
11401 043236
11402 043236 000004
11403 043240 012700 000633
11404 043244 013701 043306
11405 043250 012702 177776
11406 043254 010605
11407 043256 010737 001010
11408 043262 010506
11409 043264 012704 000340
11410 043270 012746 000340
11411 043274 012746 043314
11412 043300 005037 177776
11413 043304 000277
11414
11415 043306 000006

TST632:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #632,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1\$: MOV #-1,R4 ;R3 SHOULD GET 177777
MOV R5,SP ;RESET SP FOR ERROR LOOP
MOV #4\$,R3 ;RTS SHOULD LOAD PC FROM [R3]
MOV #-1,-(SP) ;RTS SHOULD LOAD R3 WITH 177777
CCC ;N:C = 0000
2\$: RTS R3 ;TEST THE RTS - GO TO 4\$
3\$: ERROR 5 ;RTS FAILED TO LOAD THE PC
BR 10\$;GO TO EXIT - SCHOOLS OUT
4\$: BMI 5\$;N:C = 0000 ?
BEQ 5\$
BVS 5\$
BCC 6\$
5\$: ERROR 5 ;RTS ALTERED CODES - CLEARED ONE
6\$: CMP R4,R3 ;DID R3 GET LOADED FROM STACK ?
BEQ 8\$;BR IF YES
7\$: ERROR 2 ;RTS FAILED TO LOAD REG
8\$: CMP R5,SP ;DID RTS POP THE STACK POINTER ?
BEQ TST633 ;BR IF YES
9\$: MOV R5,R4 ;[R4] = S / B SP
MOV SP,R3 ;[R3] = WAS SP
ERROR 3 ;RTS FAILED TO POP SP
10\$: MOV R5,SP ;FIX THE SP
;*****
;#TEST 633 RTT TEST - N:C = 1111
;*****
TST633:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #633,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #PSW,R2 ;DEST=PSW FOR 5\$ CALL
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1\$: MOV R5,SP ;RESET SP FOR ERROR LOOP
MOV #340,R4 ;[R4] = S / B PSW AT HTIS POINT
MOV #340,-(SP) ;NEW PSW S / B = 340
MOV #4\$,-(SP) ;NEW PC S / B = 4\$
CLR @#PSW ;CLEAR THE PSW
SCC ;N:C = 1111
2\$: RTT ;TEST THE RTT - GO TO 4\$


```

11416
11417 043310 104005      3S:  ERROR 5      ;RTT FAILED TO LOAD THE PC
11418 043312 000412      BR      8S      ;GO TO EXIT - SCHOOL'S OUT
11419
11420 043314 013703 177776  4S:  MOV    2#PSW,R3  ;SAVE THE PSW
11421 043320 020403      CMP    R4,R3      ;WAS PSW = 340 ?
11422 043322 001401      BEQ    6S      ;BR IF IT WAS
11423
11424 043324 104001      5S:  ERROR 1      ;RTT FAILED TO LOAD PSW PROPERLY
11425
11426 043326 020506      6S:  CMP    R5,SP      ;DID RTT UPDATE THE SP ?
11427 043330 001404      BEQ    TST634     ;;BR IF YES
11428
11429 043332 010504      MOV    R5,R4      ;[R4] = S / B SP
11430 043334 010603      MOV    SP,R3      ;[R3] = WAS SP
11431 043336 104003      7S:  ERROR 3      ;RTT FAILED TO UPDATE SP
11432
11433 043340 010506      8S:  MOV    R5,SP      ;FIX THE SP
11434
11435      ;*****
11436      ;*TEST 634      RTT TEST - N:C = 0000
11437      ;*****
11438 043342      TST634:
11439 043342 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
11440 043344 012700 000634  MOV    #634,R0    ;;LOAD R0 WITH TEST NUMBER
11441 043350 013701 043414  MOV    2#2S,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
11442 043354 012702 177776  MOV    #PSW,R2    ;DEST=PSW FOR 5S CALL
11443 043360 010605      MOV    SP,R5      ;SAVE THE SP
11444 043362 010737 001010  MOV    PC,2#SLPERR ;SET ERROR LOOP ADDRESS
11445 043366 010506      1S:  MOV    R5,SP      ;RESET SP FOR ERROR LOOP
11446 043370 012704 000017  MOV    #017,R4    ;[R4] = S / B PSW AT HTIS POINT
11447 043374 012746 000017  MOV    #017,-(SP) ;NEW PSW S / B = 017
11448 043400 012746 043422  MOV    #4S,-(SP)  ;NEW PC S / B = 4S
11449 043404 012737 000340 177776  MOV    #340,2#PSW ;MAKE [PSW] = 340
11450 043412 000257      CCC      ;N:C = 0000
11451
11452 043414 000006      2S:  RTT      ;TEST THE RTT - GO TO 4S
11453
11454 043416 104005      3S:  ERROR 5      ;RTT FAILED TO LOAD THE PC
11455 043420 000412      BR      8S      ;GO TO EXIT - SCHOOL'S OUT
11456
11457 043422 013703 177776  4S:  MOV    2#PSW,R3  ;SAVE THE PSW
11458 043426 020403      CMP    R4,R3      ;WAS PSW = 017 ?
11459 043430 001401      BEQ    6S      ;BR IF IT WAS
11460
11461 043432 104001      5S:  ERROR 1      ;RTT FAILED TO LOAD PSW PROPERLY
11462
11463 043434 020506      6S:  CMP    R5,SP      ;DID RTT UPDATE THE SP ?
11464 043436 001404      BEQ    TST635     ;;BR IF YES
11465
11466 043440 010504      MOV    R5,R4      ;[R4] = S / B SP
11467 043442 010603      MOV    SP,R3      ;[R3] = WAS SP
11468 043444 104003      7S:  ERROR 3      ;RTT FAILED TO UPDATE SP
11469
11470 043446 010506      8S:  MOV    R5,SP      ;FIX THE SP
11471

```

M02

```

11472
11473
11474
11475 043450
11476 043450 000004
11477 043452 012700 000635
11478 043456 013701 043502
11479 043462 010602
11480 043464 012704 125252
11481 043470 012705 043532
11482 043474 010437 043516
11483 043500 000257
11484
11485 043502 006405
11486
11487 043504 010637 001074
11488 043510 010206
11489 043512 104005
11490
11491 043514 000444
11492
11493 043516 125252
11494
11495 043520 010637 001074
11496 043524 010206
11497 043526 104005
11498
11499 043530 000436
11500
11501 043532 100403
11502 043534 001402
11503 043536 102401
11504 043540 103011
11505
11506 043542 013703 177776
11507 043546 010637 001074
11508 043552 010206
11509 043554 012702 177776
11510 043560 104007
11511 043562 000421
11512
11513 043564 020627 043520
11514 043570 001406
11515 043572 010603
11516 043574 012704 043520
11517 043600 010206
11518 043602 104003
11519
11520 043604 000410
11521
11522 043606 020504
11523 043610 001405
11524
11525 043612 010637 001074
11526 043616 010503
11527 043620 010206

```

```

*****
;TEST 635 MARK INSTRUCTION TEST - N:C=0000
*****
TST635:
SCOPE
MOV #635,R0 ;CALL THE SCOPE LOOP UTILITY
MOV @2$,R1 ;LOAD R0 WITH TEST NUMBER
MOV SP,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125252,R4 ;SAVE SP
MOV #4$,R5 ;[R5] SHOULD BE 125252
MOV R4,@6$ ;MARK GOES TO 4$ VIA [R5]
CCC ;INITIALIZE WORD LOADED INTO R5
;N:C=0000

2$: MARK+5 ;TEST THE MARK

MOV SP,@$REG5 ;SAVE BAD SP FOR PRINTING
MOV R2,SP ;RESET SP
3$: ERROR 5 ;MARK FAILED TO EXECUTE

BR TST636 ;;GO TO SCOPE EXIT

6$: 125252 ;THIS WORD SHOULD GET LOADED INTO R5

MOV SP,@$REG5 ;SAVE BAD SP FOR PRINTING
MOV R2,SP ;RESET SP
5$: ERROR 5 ;MARK FAILED TO LOAD RC FROM [R5]

BR TST636 ;;GO TO SCOPE EXIT

4$: BMI 10$ ;N:C=0000?
BEQ 10$
BVS 10$
BCC 8$

10$: MOV @PSW,R3 ;SAVE FLAGS IN R3
MOV SP,@$REG5 ;SAVE BAD SP FOR PRINTING
MOV R2,SP ;RESET SP
MOV @PSW,R2 ;DEST=PSW
7$: ERROR 7 ;MARK SET A FLAG
BR TST636 ;;GO TO SCOPE EXIT

8$: CMP SP,#6$+2 ;DID MARK RESET SP?
BEQ 11$ ;BR IF YES
MOV SP,R3 ;PUT BAD SP IN R3
MOV #6$+2,R4 ;S/B SP
MOV R2,SP ;RESET SP
9$: ERROR 3 ;MARK FAILED TO RESET SP

BR TST636 ;;GO TO SCOPE EXIT

11$: CMP R5,R4 ;DID MARK RESTORE OLD R5
BEQ 12$ ;BR IF YES

MOV SP,@$REG5 ;SAVE BAD SP FOR PRINTING
MOV R5,R3 ;WAS DEST
MOV R2,SP ;RESET SP

```

```

11528 043622 104004          ERROR 4          ;MARK FAILED TO RESET R5
11529
11530 043624 010206      12$:  MOV      R2,SP          ;RESET SP
11531
11532
11533
11534
11535 043626
11536 043626 000004          ;*****
11537 043630 012700 000636      ;:TEST 636 MARK INSTRUCTION TEST - N:C=1111
11538 043634 013701 043660      ;*****
11539 043640 010602          ;TST636:
11540 043642 012704 125252      SCOPE          ;CALL THE SCOPE LOOP UTILITY
11541 043646 012705 043710      MOV      #636,R0      ;:LOAD R0 WITH TEST NUMBER
11542 043652 010437 043674      MOV      @2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
11543 043656 000277          MOV      SP,R2        ;:SAVE SP
11544
11545 043660 006405      2$:  MARK+5          ;:[R5] SHOULD BE 125252
11546
11547 043662 010637 001074      MOV      SP,@$REG5    ;:MARK GOES TO 4$ VIA [R5]
11548 043666 010206          MOV      R2,SP        ;:INITIALIZE WORD LOADED INTO R5
11549 043670 104005      3$:  ERROR 5          ;:N:C=1111
11550
11551 043672 000444          BR      TST637        ;:TEST THE MARK
11552
11553 043674 125252      6$:  125252          ;:SAVE BAD SP FOR PRINTING
11554
11555 043676 010637 001074      MOV      SP,@$REG5    ;:RESET SP
11556 043702 010206          MOV      R2,SP        ;:MARK FAILED TO EXECUTE
11557 043704 104005      5$:  ERROR 5          ;:GO TO SCOPE EXIT
11558
11559 043706 000436          BR      TST637        ;:THIS WORD SHOULD GET LOADED INTO R5
11560
11561 043710 100003      4$:  BPL      7$          ;:SAVE BAD SP FOR PRINTING
11562 043712 001002          BNE     7$            ;:RESET SP
11563 043714 102001          BVC     7$            ;:MARK FAILED TO LOAD RC FROM [R5]
11564 043716 103411          BCS     8$
11565
11566 043720 013703 177776      7$:  MOV      @PSW,R3    ;:GO TO SCOPE EXIT
11567 043724 010637 001074      MOV      SP,@$REG5    ;:N:C=1111
11568 043730 010206          MOV      R2,SP        ;:SAVE FLAGS IN R3
11569 043732 012702 177776      MOV      @PSW,R2      ;:SAVE BAD SP FOR PRINTING
11570 043736 104007          ERROR 7              ;:RESET SP
11571 043740 000421          BR      TST637        ;:DEST=PSW
11572
11573 043742 020627 043676      8$:  CMP      SP,#6$+2   ;:MARK SET A FLAG
11574 043746 001406          BEQ     9$            ;:GO TO SCOPE EXIT
11575 043750 010603          MOV      SP,R3        ;:DID MARK RESET SP?
11576 043752 012704 043676      MOV      #6$+2,R4     ;:BR IF YES
11577 043756 010206          MOV      R2,SP        ;:PUT BAD SP IN R3
11578 043760 104003          ERROR 3              ;:S/B SP
11579
11580 043762 000410          BR      TST637        ;:RESET SP
11581
11582 043764 020504      9$:  CMP      R5,R4      ;:MARK FAILED TO RESET SP
11583 043766 001405          BEQ     10$          ;:GO TO SCOPE EXIT

```

```

11584
11585 043770 010637 001074      MOV     SP,2#SREGS      ;SAVE BAD SP FOR PRINTING
11586 043774 010503              MOV     R5,R3          ;WAS DEST
11587 043776 010206              MOV     R2,SP          ;RESET SP
11588 044000 104004              ERROR   4              ;MARK FAILED TO RESET R5
11589
11590 044002 010206      10$:   MOV     R2,SP          ;RESET SP
11591
11592      ;*****
11593      ;*TEST 637      BASIC LINE CLOCK RESPONSE TEST
11594      ;*****
11595 044004      TST637:
11596 044004 000004      SCOPE              ;CALL THE SCOPE LOOP UTILITY
11597 044006 012700 000637      MOV     #637,R0      ;LOAD R0 WITH TEST NUMBER
11598 044012 013701 044042      MOV     2#2$ ,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11599 044016 010605              MOV     SP,R5        ;SAVE SP
11600 044020 012702 177546      MOV     #LKCSR,R2    ;[R2] = LINE CLOCK ADDRESS
11601 044024 010737 001010      MOV     PC,2#SLPERR ;SET ERROR LOOP ADDRESS
11602 044030 010506      1$:   MOV     R5,SP        ;RESET SP FOR ERROR LOOP
11603 044032 012737 044046 000004      MOV     #4$,2#4     ;GO TO 4$ IF BUS TIMEOUT
11604 044040 000257              CCC                  ;SCOPE SYNC
11605
11606 044042 005712      2$:   TST     (R2)        ;REFERENCE LKCSR ADDR
11607
11608 044044 000404              BR      6$           ;GO TO EXIT
11609
11610 044046 012737 061144 000004      4$:   MOV     #BERR,2#4 ;RESTORE TIMEOUT VECTOR
11611 044054 104006      3$:   ERROR   6          ;LKCSR FAILED TO RESPOND
11612
11613 044056 010506      6$:   MOV     R5,SP        ;RESET SP
11614 044060 012737 061144 000004      MOV     #BERR,2#4   ;RESTORE TIMEOUT VECTOR
11615
11616      ;*****
11617      ;*TEST 640      LINE CLOCK TEST - LKCSR BIT 7 SET
11618      ;*****
11619 044066      TST640:
11620 044066 000004      SCOPE              ;CALL THE SCOPE LOOP UTILITY
11621 044070 012700 000640      MOV     #640,R0     ;LOAD R0 WITH TEST NUMBER
11622 044074 013701 044112      MOV     2#2$ ,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11623 044100 012702 177546      MOV     #LKCSR,R2   ;DEST ADDR = 177546
11624 044104 012704 000200      MOV     #200,R4     ;[LKCSR] S / B = 200
11625 044110 000257              CCC                  ;SCOPE SYNC
11626
11627 044112 030412      2$:   BIT     R4,(R2)    ;TEST BIT 7 IN LKCSR
11628
11629 044114 001002              BNE    TST641       ;;BR IF IT'S SET
11630
11631 044116 011203              MOV     (R2),R3     ;GET WAS DATA
11632 044120 104001      3$:   ERROR   1          ;BIT 7 NOT SET IN LKCSR
11633
11634      ;*****
11635      ;*TEST 641      LINE CLOCK TEST - LKCSR BIT 6 CLEAR
11636      ;*****
11637 044122      TST641:
11638 044122 000004      SCOPE              ;CALL THE SCOPE LOOP UTILITY
11639 044124 012700 000641      MOV     #641,R0     ;LOAD R0 WITH TEST NUMBER

```

```

11640 044130 013701 044146      MOV      2#25,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
11641 044134 012702 177546      MOV      #LKCSR,R2   ;R2 POINTS TO LKCSR
11642 044140 012704 000200      MOV      #200,R4     ;[LKCSR] S / B = 200
11643 044144 000257              CCC                 ;SCOPE SYNC
11644
11645 044146 032712 000100      25:     BIT      #100,(R2) ;TEST BIT 6 IN LKCSR
11646
11647 044152 001402              BEQ      TST642      ;;BR IF CLEAR
11648
11649 044154 011203              MOV      (R2),R3     ;GET WAS DATA
11650 044156 104001      35:     ERROR    1     ;BIT 6 (INTR. ENAB.) IN LKCSR WAS SET
11651
11652
11653 ;:*****
11653 ;*TEST 642      LINE CLOCK TEST - LKCSR BIT 6 SET
11654 ;:*****
11654 ;TST642:
11655 044160
11656 044160 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
11657 044162 012700 000642      MOV      #642,R0     ;LOAD R0 WITH TEST NUMBER
11658 044166 013701 044236      MOV      2#25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11659 044172 010605              MOV      SP,R5       ;SAVE SP
11660 044174 012702 177546      MOV      #LKCSR,R2   ;R2 POINTS TO LKCSR
11661 044200 012704 000300      MOV      #300,R4     ;[LKCSR] S / B = 300
11662 044204 010737 001010      MOV      PC,2#SLPERR ;SET ERROR LOOP ADDRESS
11663 044210 012737 044252 000100 15:     MOV      #45,2#100   ;SET UP LCLK VECTOR IN CASE LOGIC
11664 044216 012737 000340 000102      MOV      #340,2#102  ;FAULT CAUSES ATL INTERRUPT
11665 044224 010506              MOV      R5,SP       ;RESET SP FOR ERROR LOOP
11666 044226 012737 000340 177776      MOV      #340,2#PSW  ;SET PRIORITY TO LEVEL 7
11667 044234 000257              CCC                 ;SCOPE SYNC
11668
11669 044236 052712 000100      25:     BIS      #100,(R2) ;SET BIT 6 IN LKCSR
11670
11671 044242 020412              CMP      R4,(R2)     ;RESULT CORRECT?
11672 044244 001402              BEQ      45          ;BR IF YES
11673
11674 044246 011203              MOV      (R2),R3     ;GET WAS DATA
11675 044250 104001      35:     ERROR    1     ;BIT 6 FAILED TO SET IN LKCSR
11676
11677 044252 042737 000102 000100 45:     BIC      #102,2#100  ;RESTORE TRAP CATCHER IN LINE CLOCK VECTOR
11678 044260 005037 000102              CLR      2#102
11679 044264 042712 000100              BIC      #100,(R2)   ;TURN OF LINE CLK INTR. ENAB.
11680 044270 010506              MOV      R5,SP       ;RESET SP
11681
11682
11683 ;:*****
11683 ;*TEST 643      LINE CLK BASIC INTERRUPT TEST
11684 ;:*****
11684 ;TST643:
11685 044272
11686 044272 000004              SCOPE              ;CALL THE SCOPE LOOP UTILITY
11687 044274 012700 000643      MOV      #643,R0     ;LOAD R0 WITH TEST NUMBER
11688 044300 013701 044346      MOV      2#25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11689 044304 010605              MOV      SP,R5       ;SAVE SP
11690 044306 012702 177546      MOV      #LKCSR,R2   ;R2 POINTS TO LKCSR
11691 044312 010737 001010      MOV      PC,2#SLPERR ;SET ERROR LOOP ADDRESS
11692 044316 010506              MOV      R5,SP       ;RESET SP FOR ERROR LOOP
11693 044320 005004              CLR      R4          ;INITIALIZE TIMER
11694 044322 012737 044364 000100      MOV      #45,2#100   ;SET UP LINE CLOCK VECTOR TO TO
11695 044330 012737 000340 000102      MOV      #340,2#102  ;TO 45 WITH PROCESSOR PRIORITY = 7

```

```

11696 044336 005012          CLR      (R2)          ;CLEAR LKCSR
11697 044340 005037 177776  CLR      @#PSW        ;SET PRIORITY TO LEVEL 000
11698 044344 000257          CCC                ;SCOPE SYNC
11699
11700 044346 052712 000100  2S:     BIS      #100,(R2) ;ENABLE LINE CLK INTERRUPT
11701
11702 044352 005304          DEC      R4           ;WAIT FOR INTR - REPORT ERROR IF
11703 044354 001376          BNE     .-2          ;R4 GOES TO 000000
11704
11705 044356 042712 000100          BIC     #100,(R2)    ;TURN OFF INTR. ENAB.
11706 044362 104006 3S:     ERROR    6      ;LINE CLK FAILED TO INTERRUPT
11707
11708 044364 042712 000100  4S:     BIC     #100,(R2) ;TURN OFF INTR. ENAB.
11709 044370 012737 000102 000100  MOV     #102,@#100   ;RESTORE TRAP CATCHER IN LINE CLK VECTOR
11710 044376 005037 000102          CLR     @#102
11711 044402 010506          MOV     R5,SP       ;RESET SP
11712 044404 005037 177776  CLR     @#PSW       ;RESET PRIORITY TO LEVEL 0
11713
11714
11715
11716
11717 044410          ;*****
11718 044410 000004          ;*TEST 644 RESET TEST - N:C = 1111
11719 044412 012700 000644          ;*****
11720 044416 013701 044450          ;*T644:
11721 044422 012737 000001 001110  SCOPE      ;CALL THE SCOPE LOOP UTILITY
11722 044430 012702 177564          MOV     #644,R0     ;LOAD R0 WITH TEST NUMBER
11723 044434 012737 000340 177776  MOV     @#2S,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11724 044442 052712 000004          MOV     #1,@#TIMES  ;NO ITERATIONS ON THIS TEST
11725 044446 000277          MOV     #XCSR,R2    ;R2 POINTS TO DL11 XCSR
11726
11727 044450 000005 2S:     BIS     #4,(R2)   ;MAKE PRTY. BITS ALL 1'S
11728          SCC                ;SET THE DL11 MAINT. BIT
11729 044452 013705 177776          ;N:C = 1111
11730 044456 032712 000004          2S:     RESET      ;TEST THE RESET - IT SHOULD CLEAR THE DL11 MAINT BIT
11731 044462 001403          MOV     @#PSW,R5    ;SAVE THE PSW
11732
11733 044464 042712 000004          BIT     #4,(R2)    ;DID MAINT. BIT CLEAR ??
11734 044470 104006 3S:     BEQ     4S      ;BR IF YES
11735
11736 044472 022705 000357  4S:     BIC     #4,(R2) ;MAKE SURE TO TURN OFF MAINT. BIT
11737 044476 001406          ERROR    5          ;RESET FAILED TO CLEAR MAINT BIT
11738
11739 044500 012704 000357  4S:     CMP     #357,R5 ;DID RESET ALTER THE PSW ??
11740 044504 010503          BEQ     6S          ;BR IF NOT
11741 044506 012702 177776          MOV     #357,R4    ;[R4] = S/B PSW
11742 044512 104001 5S:     MOV     R5,R3     ;[R3] = WAS PSW
11743          MOV     #PSW,R2  ;DEST = PSW
11744 044514 005037 177776  6S:     ERROR    1    ;RESET ALTERED THE PSW
11745 044520 042737 000004 177564  CLR     @#PSW       ;CLEAR OUT THE PSW
11746          BIC     #4,@#XCSR ;MAKE SURE MAINT BIT IS OFF
11747
11748
11749
11750 044526          ;*****
11751 044526 000004          ;*TEST 645 RESET TEST - N:C = 0000
11751          ;*****
11751          ;*T645:
11751          SCOPE      ;CALL THE SCOPE LOOP UTILITY

```

E03

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T645

MACY11 27(1006) 08-FEB-77 16:23 PAGE 214
RESET TEST - N:C = 0000

```

11752 044530 012700 000645      MOV      #645,R0      ;;LOAD R0 WITH TEST NUMBER
11753 044534 013701 044564      MOV      @#25,R1     ;;LOAD R1 WITH TEST INSTRUCTION WORD
11754 044540 012737 000001 001110  MOV      #1,@#TIMES  ;;NO ITERATIONS ON THIS TEST
11755 044546 012702 177564      MOV      @XCSR,R2   ;;R2 POINTS TO DL11 XCSR
11756 044552 005037 177776      CLR      @#PSW      ;;MAKE PRTY. BITS ALL 0'S
11757 044556 052712 000004      BIS      #4,(R2)    ;;SET THE DL11 MAINT. BIT
11758 044562 000257                CCC                    ;;N:C = 0000
11759
11760 044564 000005      2$:      RESET                    ;;TEST THE RESET - IT SHOULD CLEAR THE DL11 MAINT BIT
11761
11762 044566 013705 177776      MOV      @#PSW,R5   ;;SAVE THE PSW
11763 044572 032712 000004      BIT      #4,(R2)   ;;DID MAINT. BIT CLEAR ??
11764 044576 001403      BEQ      4$        ;;BR IF YES
11765
11766 044600 042712 000004      3$:      BIC      #4,(R2)   ;;MAKE SURE TO TURN OFF MAINT. BIT
11767 044604 104006      ERROR   6        ;;RESET FAILED TO CLEAR MAINT BIT
11768
11769 044606 022705 000000      4$:      CMP      #0,R5     ;;DID RESET ALTER THE PSW ??
11770 044612 001406      BEQ      6$        ;;BR IF NOT
11771
11772 044614 012704 000357      MOV      #357,R4   ;;[R4] = S/B PSW
11773 044620 010503      MOV      R5,R3     ;;[R3] = WAS PSW
11774 044622 012702 177776      MOV      @#PSW,R2  ;;DEST = PSW
11775 044626 104001      5$:      ERROR   1        ;;RESET ALTERED THE PSW
11776
11777 044630 005037 177776      6$:      CLR      @#PSW     ;;CLEAR OUT THE PSW
11778 044634 042737 000004 177564  BIC      #4,@#XCSR  ;;MAKE SURE MAINT BIT IS OFF
11779
11780      ;;*****
11781      ;;*TEST 646      WAIT INSTRUCTION TEST - [PSW] = 151
11782      ;;*****
11783      TST646:
11784 044642 000004      SCOPE
11785 044644 012700 000646      MOV      #646,R0   ;;CALL THE SCOPE LOOP UTILITY
11786 044650 013701 044736      MOV      @#25,R1   ;;LOAD R0 WITH TEST NUMBER
11787 044654 010605      MOV      SP,R5     ;;LOAD R1 WITH TEST INSTRUCTION WORD
11788 044656 010737 001010      MOV      PC,@#SLPERR ;;SAVE THE SP
11789 044662 012702 177564      1$:      MOV      @XCSR,R2  ;;SET ERROR LOOP ADDRESS
11790 044666 012737 044754 000064  MOV      #4,@#64   ;;R2 POINT TO DL11 XCSR
11791 044674 012737 000200 000066  MOV      #200,@#66 ;;GO TO 4$ ON DL11 INTR.
11792 044702 010506      MOV      R5,SP    ;;AT LEVEL 4
11793 044704 005012      CLR      (R2)     ;;RESET SP FOR ERROR LOOP
11794 044706 005003      CLR      R3      ;;INIT DL11 XCSR
11795
11796 044710 105712      3$:      TSTB   (R2)     ;;DL11 XMIT READY SET ??
11797 044712 100403      BMI     5$        ;;BR IF YES
11798 044714 005303      DEC     R3      ;;COUNT THE TIMER
11799 044716 001374      BNE    3$        ;;BR IF NO TIMEOUT
11800 044720 000440      BR     9$        ;;GO REPORT TIMEOUT
11801
11802 044722 012737 000140 177776  5$:      MOV      #140,@#PSW ;;SET PSW PRTY BITS TO LEVEL 3
11803 044730 000277      SCC                    ;;N:C=1111
11804 044732 152712 000100      BISB   #100,(R2)  ;;ENAB. DL11 INTR - N:C=1001
11805
11806 044736 000001      2$:      WAIT                    ;;TEST THE WAIT-GO TO 4$ ON INTR
11807

```

F03

MAINDEC-11-DQKDA-A K011-K BASIC LOGIC TESTS
DQKDA.A.P11 08-FEB-77 16:17 T646

MACY11 27(1006) 08-FEB-77 16:23 PAGE 215
WAIT INSTRUCTION TEST - [PSW] = 151

```

11808 044740 012737 000340 177776      MOV      #340,2#PSW      ;LOCK OUT INTR
11809 044746 005012                    CLR      (R2)           ;TURN OFF DL11 INTR ENAB
11810 044750 104006                    ERROR   6               ;WAIT FAILED TO EXECUTE PROPERLY
11811 044752 000424                    BR       8$             ;GO EXIT THIS TEST
11812
11813 044754 042712 000100      4$:    BIC      #100,(R2)    ;TURN OFF DL11 INTR ENAB
11814 044760 022716 044740      CMP      #2$+2,(SP)    ;DID WAIT GET FETCHED ??
11815 044764 001402                    BEQ     6$              ;BR IF YES
11816
11817 044766 104006                    ERROR   6               ;WAIT NOT FETCHED PROPERLY
11818 044770 000415                    BR       8$             ;GO EXIT THE TEST
11819
11820 044772 022766 000151 000002 6$:    CMP      #151,2(SP)    ;DID "WAIT" ALTER THE PSW ??
11821 045000 001411                    BEQ     8$              ;BR IF YES
11822
11823 045002 012704 000151      MOV      #151,R4        ;[R4] = S/B PSW
11824 045006 016603 000002      MOV      2(SP),R3      ;[R3] = WAS PSW
11825 045012 012702 177776      MOV      #PSW,R2       ;DEST = PSW
11826 045016 104001      7$:    ERROR   1           ;"WAIT" ALTERED THE PSW
11827 045020 000401                    BR       8$             ;GOT TO EXIT TEST
11828
11829 045022 104006      9$:    ERROR   6           ;DL11 FAILED TO SET READY ON TIME
11830
11831 045024 010506      8$:    MOV      R5,SP        ;RESET THE SP
11832 045026 005037 177776      CLR      2#PSW         ;CLEAR OUT THE PSW
11833 045032 005012                    CLR      (R2)           ;TURN OFF DL11 INTR.
11834 045034 012737 000066 000064      MOV      #66,2#64     ;RESTORE DL11 VECTOR WITH TRAPCATCHER
11835 045042 005037 000066      CLR      2#66
11836
11837
11838
11839
11840 045046
11841 045046 000004
11842 045050 012700 000647      SCOPE
11843 045054 013701 045140      MOV      #647,R0      ;CALL THE SCOPE LOOP UTILITY
11844 045060 010605                    MOV      2#2$,R1      ;LOAD R0 WITH TEST NUMBER
11845 045062 010737 001010      MOV      SP,R5        ;LOAD R1 WITH TEST INSTRUCTION WORD
11846 045066 012702 177564      MOV      PC,2#SLPERR  ;SAVE THE SP
11847 045072 012737 045156 000064 1$:    MOV      #XCSR,R2     ;SET ERROR LOOP ADDRESS
11848 045100 012737 000200 000066      MOV      #4$,2#64     ;R2 POINT TO DL11 XCSR
11849 045106 010506                    MOV      #200,2#66    ;GO TO 4$ ON DL11 INTR.
11850 045110 005012                    MOV      R5,SP        ;AT LEVEL 4
11851 045112 005003                    CLR      (R2)         ;RESET SP FOR ERROR LOOP
11852
11853 045114 105712      3$:    TSTB   (R2)         ;DL11 XMIT READY SET ??
11854 045116 100403                    BMI     5$             ;BR IF YES
11855 045120 005303                    DEC     R3             ;COUNT THE TIMER
11856 045122 001374                    BNE    3$             ;BR IF NO TIMEOUT
11857 045124 000437                    BR     9$              ;GO REPORT TIMEOUT
11858
11859 045126 005037 177776      5$:    CLR      2#PSW        ;SET PSW PRY BITS TO LEVEL 0
11860 045132 000257                    CCC
11861 045134 152712 000100      BISB   #100,(R2)     ;N:C=0000
11862
11863 045140 000001      2$:    WAIT              ;ENAB. DL11 INTR - N:C=1000
;TEST THE WAIT-GO TO 4$ ON INTR

```

```

*****
;TEST 647 WAIT INSTRUCTION TEST - [PSW] = 010
*****
TST647:

```


H03

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 217
 BR PRIORITY ARBITRATION TEST - LEVEL 1 USING LINE CLK

```

11920 045346 012737 000102 000100      MOV    #102,#100      ;RESTORE TRAP CATCHER IN THE VECTOR
11921 045354 005037 000102              CLR    #102
11922 045360 010506              MOV    R5,SP          ;RESET THE SP
11923 045362 005037 177776              CLR    #PSW          ;SET CPU PRIORITY BACK TO LEVEL 0
11924
11925      ;*****
11926      ;*TEST 651      BR PRIORITY ARBITRATION TEST - LEVEL 2 USING LINE CLK
11927      ;*****
11928 045366      †ST651:
11929 045366 000004              SCOPE                ;CALL THE SCOPE LOOP UTILITY
11930 045370 012700 000651      MOV    #651,R0        ;LOAD R0 WITH TEST NUMBER
11931 045374 013701 045442      MOV    #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
11932 045400 010605              MOV    SP,R5          ;SAVE THE SP
11933 045402 010737 001010      MOV    PC,#SLPERR     ;SET ERROR LOOP ADDRESS
11934 045406 012702 177546      1$:  MOV    #LKCSR,R2    ;R2 POINTS TO LINE CLK CSR
11935 045412 012737 045460 000100      MOV    #45,#100       ;IF INTR OCCURS - GO TO 4$
11936 045420 012737 000340 000102      MOV    #340,#102     ;WITH CPU PRIORITY AT LEVEL 7
11937 045426 010506              MOV    R5,SP          ;RESET SP FOR ERROR LOOPING
11938 045430 005004              CLR    R4             ;INITIALIZE R4 AS TIMER
11939 045432 012737 000100 177776      MOV    #100,#PSW     ;SET CPU PRIORITY TO LEVEL 2
11940 045440 000257              CCC                  ;SCOPE SYNC
11941
11942 045442 052712 000100      2$:  BIS    #100,(R2)    ;ENABLE LINE CLK INTERRUPTS
11943
11944 045446 005304              DEC    R4             ;COUNT THE TIMER - LCLK SHOULD PREVENT
11945 045450 001376              BNE    .-2           ;TIMER FROM GETTING BACK TO 000000
11946
11947 045452 042712 000100      3$:  BIC    #100,(R2)    ;TURN OFF THE INTERRUPT ENABLE
11948 045456 104006              ERROR 6              ;LINE CLK FAILED TO INTR AT LEVEL 2
11949
11950 045460 042712 000100      4$:  BIC    #100,(R2)    ;TURN OFF INTR. ENABLE
11951 045464 012737 000102 000100      MOV    #102,#100     ;RESTORE TRAP CATCHER IN THE VECTOR
11952 045472 005037 000102              CLR    #102
11953 045476 010506              MOV    R5,SP          ;RESET THE SP
11954 045500 005037 177776              CLR    #PSW          ;SET CPU PRIORITY BACK TO LEVEL 0
11955
11956      ;*****
11957      ;*TEST 652      BR PRIORITY ARBITRATION TEST - LEVEL 3 USING LINE CLK
11958      ;*****
11959 045504      †ST652:
11960 045504 000004              SCOPE                ;CALL THE SCOPE LOOP UTILITY
11961 045506 012700 000652      MOV    #652,R0        ;LOAD R0 WITH TEST NUMBER
11962 045512 013701 045560      MOV    #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
11963 045516 010605              MOV    SP,R5          ;SAVE THE SP
11964 045520 010737 001010      MOV    PC,#SLPERR     ;SET ERROR LOOP ADDRESS
11965 045524 012702 177546      1$:  MOV    #LKCSR,R2    ;R2 POINTS TO LINE CLK CSR
11966 045530 012737 045576 000100      MOV    #45,#100       ;IF INTR OCCURS - GO TO 4$
11967 045536 012737 000340 000102      MOV    #340,#102     ;WITH CPU PRIORITY AT LEVEL 7
11968 045544 010506              MOV    R5,SP          ;RESET SP FOR ERROR LOOPING
11969 045546 005004              CLR    R4             ;INITIALIZE R4 AS TIMER
11970 045550 012737 000140 177776      MOV    #140,#PSW     ;SET CPU PRIORITY TO LEVEL 3
11971 045556 000257              CCC                  ;SCOPE SYNC
11972
11973 045560 052712 000100      2$:  BIS    #100,(R2)    ;ENABLE LINE CLK INTERRUPTS
11974
11975 045564 005304              DEC    R4             ;COUNT THE TIMER - LCLK SHOULD PREVENT

```

```

11976 045566 001376          BNE      .-2          ;TIMER FROM GETTING BACK TO 000000
11977
11978 045570 042712 000100    3$:      BIC      #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
11979 045574 104006          ERROR    6          ;LINE CLK FAILED TO INTR AT LEVEL 3
11980
11981 045576 042712 000100    4$:      BIC      #100,(R2) ;TURN OFF INTR. ENABLE
11982 045602 012737 000102 000100    MOV      #102,#100 ;RESTORE TRAP CATCHER IN THE VECTOR
11983 045610 005037 000102          CLR      #102
11984 045614 010506          MOV      R5,SP      ;RESET THE SP
11985 045616 005037 177776          CLR      #PSW      ;SET CPU PRIORITY BACK TO LEVEL 0
11986

```

```

;*****
;TEST 653 BR PRIORITY ARBITRATION TEST - LEVEL 4 USING LINE CLK
;*****
†T653:

```

```

11990 045622          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11991 045622 000004          MOV      #653,R0    ;LOAD R0 WITH TEST NUMBER
11992 045624 012700 000653          MOV      #25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
11993 045630 013701 045676          MOV      SP,R5     ;SAVE THE SP
11994 045634 010605          MOV      PC,#SLPERR ;SET ERROR LOOP ADDRESS
11995 045636 010737 001010          MOV      #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
11996 045642 012702 177546          1$:      MOV      #45,#100 ;IF INTR OCCURS - GO TO 4$
11997 045646 012737 045714 000100    MOV      #340,#102 ;WITH CPU PRIORITY AT LEVEL 7
11998 045654 012737 000340 000102    MOV      R5,SP     ;RESET SP FOR ERROR LOOPING
12000 045664 005004          CLR      R4       ;INITIALIZE R4 AS TIMER
12001 045666 012737 000200 177776    MOV      #200,#PSW ;SET CPU PRIORITY TO LEVEL 4
12002 045674 000257          CCC          ;SCOPE SYNC
12003
12004 045676 052712 000100    2$:      BIS      #100,(R2) ;ENABLE LINE CLK INTERRUPTS
12005
12006 045702 005304          DEC      R4       ;COUNT THE TIMER - LCLK SHOULD PREVENT
12007 045704 001376          BNE      .-2     ;TIMER FROM GETTING BACK TO 000000
12008
12009 045706 042712 000100    3$:      BIC      #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
12010 045712 104006          ERROR    6          ;LINE CLK FAILED TO INTR AT LEVEL 4
12011
12012 045714 042712 000100    4$:      BIC      #100,(R2) ;TURN OFF INTR. ENABLE
12013 045720 012737 000102 000100    MOV      #102,#100 ;RESTORE TRAP CATCHER IN THE VECTOR
12014 045726 005037 000102          CLR      #102
12015 045732 010506          MOV      R5,SP     ;RESET THE SP
12016 045734 005037 177776          CLR      #PSW     ;SET CPU PRIORITY BACK TO LEVEL 0
12017

```

```

;*****
;TEST 654 BR PRIORITY ARBITRATION TEST - LEVEL 5 USING LINE CLK
;*****
†T654:

```

```

12021 045740          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12022 045740 000004          MOV      #654,R0    ;LOAD R0 WITH TEST NUMBER
12023 045742 012700 000654          MOV      #25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
12024 045746 013701 046014          MOV      SP,R5     ;SAVE THE SP
12025 045752 010605          MOV      PC,#SLPERR ;SET ERROR LOOP ADDRESS
12026 045754 010737 001010          MOV      #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
12027 045760 012702 177546          1$:      MOV      #45,#100 ;IF INTR OCCURS - GO TO 4$
12028 045764 012737 046032 000100    MOV      #340,#102 ;WITH CPU PRIORITY AT LEVEL 7
12029 045772 012737 000340 000102    MOV      R5,SP     ;RESET SP FOR ERROR LOOPING
12030 046000 010506          CLR      R4       ;INITIALIZE R4 AS TIMER
12031 046002 005004

```

```

12032 046004 012737 000240 177776      MOV      #240,2#PSW      ;SET CPU PRIORITY TO LEVEL 5
12033 046012 000257                      CCC                      ;SCOPE SYNC
12034
12035 046014 052712 000100      2$:     BIS      #100,(R2) ;ENABLE LINE CLK INTERRUPTS
12036
12037 046020 005304                      DEC      R4              ;COUNT THE TIMER - LCLK SHOULD PREVENT
12038 046022 001376                      BNE     .-2             ;TIMER FROM GETTING BACK TO 000000
12039
12040 046024 042712 000100      3$:     BIC      #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
12041 046030 104006                      ERROR   6               ;LINE CLK FAILED TO INTR AT LEVEL 5
12042
12043 046032 042712 000100      4$:     BIC      #100,(R2) ;TURN OFF INTR. ENABLE
12044 046036 012737 000102 000100  MOV      #102,2#100    ;RESTORE TRAP CATCHER IN THE VECTOR
12045 046044 005037 000102      CLR      2#102
12046 046050 010506                      MOV      R5,SP          ;RESET THE SP
12047 046052 005037 177776      CLR      2#PSW         ;SET CPU PRIORITY BACK TO LEVEL 0
12048
12049
12050                                     ;*****
12051                                     ;*TEST 655      BR PRIORITY ARBITRATION TEST - LEVEL 6 USING LINE CLK
12052                                     ;*****
12052 046056                                     †ST655:
12053 046056 000004                      SCOPE                   ;CALL THE SCOPE LOOP UTILITY
12054 046060 012700 000655      MOV      #655,R0        ;LOAD R0 WITH TEST NUMBER
12055 046064 013701 046144      MOV      2#25,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
12056                                     .SBTTL USER CONTROLLED BREAKPOINT -- BIT13
12057 046070 032737 020000 063160  BIT      #BIT13,2#BPTLOC ;BREAKPOINT HALT SET ??
12058 046076 001401                      BEQ     .+4             ;BR IF NOT
12059 046100 000000                      HALT                    ;BREAK-DEPRESS CONTINUE TO CONTINUE
12060 046102 010605                      MOV      SP,R5          ;SAVE THE SP
12061 046104 010737 001010      MOV      PC,2#SLPERR    ;SET ERROR LOOP ADDRESS
12062 046110 012702 177546      1$:     MOV      #LKCSR,R2  ;R2 POINTS TO LINE CLK CSR
12063 046114 012737 046156 000100  MOV      #45,2#100      ;IF INTR OCCURS - GO TO 45
12064 046122 012737 000340 000102  MOV      #340,2#102     ;WITH CPU PRIORITY AT LEVEL 7
12065 046130 010506                      MOV      R5,SP          ;RESET SP FOR ERROR LOOP
12066 046132 005004                      CLR      R4             ;INITIALIZE R4 AS TIMER
12067 046134 012737 000300 177776  MOV      #300,2#PSW     ;SET CPU PRIORITY TO LEVEL 6
12068 046142 000257                      CCC                      ;SCOPE SYNC
12069
12070 046144 052712 000100      2$:     BIS      #100,(R2) ;ENABLE INTERRUPTS
12071
12072 046150 005304                      DEC      R4              ;COUNT UNTIL (R4) = 000000 - THEN
12073 046152 001376                      BNE     .-2             ;CONTINUE - NO INTERRUPT SHOULD OCCUR
12074 046154 000403                      BR      6$              ;GO TO EXIT - ALL OK
12075
12076 046156 042712 000100      4$:     BIC      #100,(R2) ;TURN OFF THE INTR ENABLE
12077 046162 104006      3$:     ERROR   6               ;INTR OCCURRED WITH CPU AT LEVEL 6
12078
12079 046164 042712 000100      6$:     BIC      #100,(R2) ;TURN OFF INTR ENABLE
12080 046170 012737 000102 000100  MOV      #102,2#100    ;RESET THE TRAP CATCHER IN THE VECTOR
12081 046176 005037 000102      CLR      2#102
12082 046202 010506                      MOV      R5,SP          ;RESET SP JUST IN CASE
12083 046204 005037 177776      CLR      2#PSW         ;SET CPU PRIORITY BACK TO LEVEL 0
12084
12085                                     ;*****
12086                                     ;*TEST 656      BR PRIORITY ARBITRATION TEST - LEVEL 7 USING DL11
12087                                     ;*****
    
```

K03

MAINDEC-11-DOKDA-A KD11-K BASIC LOGIC TESTS
 DOKDAA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 220
 BR PRIORITY ARBITRATION TEST - LEVEL 7 USING DL11

```

12088 046210          TST656:
12089 046210 000004          SCOPE
12090 046212 012700 000656  MOV      #656,R0          ;CALL THE SCOPE LOOP UTILITY
12091 046216 013701 046264  MOV      @#25,R1         ;LOAD R0 WITH TEST NUMBER
12092 046222 010605          MOV      SP,R5          ;LOAD R1 WITH TEST INSTRUCTION WORD
12093 046224 010737 001010  MOV      PC,@#SLPERR    ;SAVE THE SP
12094 046230 012702 177564  MOV      @XCSR,R2       ;SET ERROR LOOP ADDRESS
12095 046234 012737 046276 000064 15:  MOV      #45,@#64      ;R2 POINTS TO DL11 XCSR
12096 046242 012737 000340 000066  MOV      #340,@#66     ;IF INTR OCCURS - GO TO 45
12097 046250 010506          MOV      R5,SP         ;WITH CPU PRIORITY AT LEVEL 7
12098 046252 005004          CLR      R4            ;RESET SP FOR ERROR LOOP
12099 046254 012737 000340 177776  MOV      #340,@#PSW    ;INITIALIZE R4 AS TIMER
12100 046262 000257          CCC                    ;SET CPU PRIORITY TO LEVEL 7
12101                                     ;SCOPE SYNC
12102 046264 052712 000100 25:  BIS      #100,(R2)      ;ENABLE INTERRUPTS
12103                                     ;
12104 046270 005304          DEC      R4            ;COUNT UNTIL [R4] = 000000 - THEN
12105 046272 001376          BNE     .-2            ;CONTINUE - NO INTERRUPT SHOULD OCCUR
12106 046274 000403          BR      65            ;GO TO EXIT - ALL OK
12107                                     ;
12108 046276 042712 000100 45:  BIC      #100,(R2)      ;TURN OFF THE INTR ENABLE
12109 046302 104006          35:  ERROR  6            ;INTR OCCURRED WITH CPU AT LEVEL 7
12110                                     ;
12111 046304 042712 000100 65:  BIC      #100,(R2)      ;TURN OFF INTR ENABLE
12112 046310 012737 000066 000064  MOV      #66,@#64      ;RESET THE TRAP CATCHER IN THE VECTOR
12113 046316 005037 000066          CLR      @#66
12114 046322 010506          MOV      R5,SP         ;RESET SP JUST IN CASE
12115 046324 005037 177776          CLR      @#PSW        ;SET CPU PRIORITY BACK TO LEVEL 0
12116                                     ;
12117                                     ;*****
12118                                     ;*TEST 657 "CLR @#PSW" ALLOWS IMMEDIATE BR-BG-INTR SEQUENCE
12119                                     ;THIS TEST VERIFIES THAT IF A "BR" REQUEST IS PENDING WHEN A "CLR @#PSW"
12120                                     ;IS EXECUTED TO LOWER THE CPU PRIORITY, THE REQUEST IS GRANTED BEFORE
12121                                     ;EXECUTION OF THE INSTRUCTION FOLLOWING THE "CLR"
12122                                     ;*****
12123                                     ;TST657:
12124 046330          SCOPE
12125 046330 000004          MOV      #657,R0          ;CALL THE SCOPE LOOP UTILITY
12126 046332 012700 000657  MOV      @#25,R1         ;LOAD R0 WITH TEST NUMBER
12127 046336 013701 046430  MOV      @LKCSR,R2      ;LOAD R1 WITH TEST INSTRUCTION WORD
12128 046342 012702 177546  MOV      @LKCSR,R2      ;R2 POINTS TO LINE CLK CSR
12129 046346 010605          MOV      SP,R5          ;SAVE THE SP
12130 046350 010737 001010  MOV      PC,@#SLPERR    ;SET ERROR LOOP ADDRESS
12131 046354 012737 046436 000100 15:  MOV      #45,@#100     ;SET UP LCLK VECTOR TO GO TO 45
12132 046362 012737 000300 000102  MOV      #300,@#102    ;
12133 046370 010506          MOV      R5,SP         ;RESET THE SP FOR ERROR LOOPING
12134 046372 005004          CLR      R4            ;INITIALIZE TIMER FO KW
12135 046374 005003          CLR      R3            ;CLEAR SOFTWARE FLAG
12136 046376 012737 000340 177776  MOV      #340,@#PSW    ;LOCK OUT ALL INTRs
12137 046404 052712 000100  BIS      #100,(R2)      ;ENABLE LCLK INTRs
12138 046414 105712          115: BIC      #200,(R2)     ;CLEAR LINE CLOCK READY
12139 046416 100403          TSTB   (R2)           ;LCLK READY TO INTR ??
12140 046420 005304          BMI     125           ;BR IF YES
12141 046422 001374          DEC      R4            ;COUNT THE TIMER
12142 046424 000411          BNE     115           ;BR IF NO TIMEOUT
12143 046426 000257          BR      65            ;GO REPORT TIMEOUT
12143                                     ;SCOPE SYNC

```

```

12144
12145 046430 005037 177776      2S:  CLR      @PSW      ;ALLOW INTRS - LCLK SHOULD INTERRUPT
12146                                     ;BEFORE FETCHING NEXT INSTRUCTION
12147 046434 005103                                     ;SHOULD NOT BE FETCHED
12148 046436 005012      4S:  CLR      (R2)      ;DISABLE THE LCLK INTR
12149 046440 005703                                     ;DID SOFTWARE FLAG GET SET ??
12150 046442 001404                                     ;BR IF NOT - IT WORKED OK
12151 046444 104006      3S:  ERROR    6        ;LCLK FAILED TO INTR ONTIME
12152 046446 000402      BR      8S        ;GO EXIT
12153
12154 046450 005012      6S:  CLR      (R2)      ;DISABLE LCLK INTR
12155 046452 104006      5S:  ERROR    6        ;LINE CLK TIMED OUT
12156
12157 046454 010506      8S:  MOV      R5,SP      ;RESET THE SP
12158 046456 012737 000102 000100  MOV      @102,@#100  ;RESTORE THE LINE CLK TRAPCATCHER
12159 046464 005037 000102      CLR      @#102
12160
12161                                     ;*****
12162                                     ;*TEST 660 "BR6 VS BR4" PRIORITY ARBITRATION TEST
12163                                     ;THIS TEST VERIFIES THAT IF BOTH A "BR4" AND A "BR6" REQUEST ARE
12164                                     ;PENDING WHEN THE CPU PRIORITY IS LOWERED TO ALLOW INTRS. THAT "BR6"
12165                                     ;REQUEST IS GRANTED FIRST EVEN THOUGH THE "BR4" REQUEST MAY HAVE
12166                                     ;OCCURRED FIRST
12167                                     ;*****
12168 046470                                     ;TST660:
12169 046470 000004      SCOPE
12170 046472 012700 000660  MOV      @#660,R0      ;CALL THE SCOPE LOOP UTILITY
12171 046476 013701 046636  MOV      @#25,R1      ;LOAD R0 WITH TEST NUMBER
12172 046502 010605      MOV      SP,R5        ;LOAD R1 WITH TEST INSTRUCTION WORD
12173 046504 010737 001010  MOV      PC,@#SLPERR  ;SAVE THE SP
12174 046510 012702 177546  MOV      @LKCSR,R2    ;SET ERROR LOOP ADDRESS
12175 046514 012703 177564  MOV      @XCSR,R3     ;R2 POINTS TO LINE CLK CSR
12176 046520 012737 046646 000100  MOV      @#45,@#100   ;R3 POINTS TO DL11 XCSR
12177 046526 012737 000300 000102  MOV      @#300,@#102  ;SET UP THE LCLK VECTOR - GO TO 4S
12178 046534 012737 046700 000064  MOV      @#85,@#64    ;SET UP THE DL11 VECTOR - GO TO 8S
12179 046542 012737 000200 000066  MOV      @#200,@#66
12180 046550 010506      MOV      R5,SP        ;RESET SP FOR ERROR LOOPING
12181 046552 012737 000340 177776  MOV      @#340,@PSW   ;LOCK OUT ALL INTRS
12182 046560 005037 063236  CLR      @#MBUFO      ;INIT TIMER
12183 046564 005037 063242  CLR      @#MBUF1     ;CLEAR DL11 INTR FLAG
12184 046570 005004      CLR      R4          ;INIT TIMER
12185 046572 052713 000100  BIS      @#100,(R3)   ;ENABLE DL11 XMIT INTR
12186 046576 105713      11S: TSTB      (R3)    ;XMIT READY SET ??
12187 046600 100403      BMI     12S         ;BR IF YES
12188 046602 005304      DEC     R4          ;COUNT THE TIMER
12189 046604 001374      BNE    11S         ;BR IF NO TIMEOUT
12190 046606 000443      BR     5S          ;GO REPORT TIMEOUT FOR DL11
12191
12192 046610 005004      12S: CLR      R4          ;INIT THE TIMER AGAIN
12193 046612 052712 000100  BIS      @#100,(R2)   ;ENABLE LCLK INTRS
12194 046616 042712 000200  BIC     @#200,(R2)   ;CLEAR THE LINE CLOCK READY BIT
12195 046622 105712      13S: TSTB      (R2)    ;LCLK READY TO INTR
12196 046624 100403      BMI     14S         ;BR IF YES
12197 046626 005304      DEC     R4          ;COUNT THE TIMER
12198 046630 001374      BNE    13S         ;BR IF NO TIMEOUT
12199 046632 000436      BR     7S          ;GO REPORT LINE CLK TIMEOUT

```



```

12256 047004 010737 001010      MOV      PC, @SLPERR      ;SET ERROR LOOP ADDRESS
12257 047010 012737 047026 000014 1$:      MOV      @4$, @#14      ;GO TO 4$ ON "BPT" TRAP
12258 047016 010506      MOV      RS, SP          ;RESET THE SP FOR ERROR LOOPING
12259 047020 000257      CCC                      ;SCOPE SYNC
12260
12261 047022 000003      2$:      BPT                      ;TEST THE "BPT" - GO TO 4$
12262
12263 047024 104005      3$:      ERROR      5          ;BPT FAILED TO TRAP
12264
12265 047026 010506      4$:      MOV      RS, SP          ;RESET THE SP
12266 047030 012737 000016 000014      MOV      @16, @#14      ;RESTORE THE VECTOR
12267
12268      ;*****
12269      ;#TEST 662      RED ZONE OVERFLOW TEST - MOV R,-(SP)
12270      ;*****
12271      †ST662:
12272 047036 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
12273 047040 012700 000662      MOV      @662, R0      ;LOAD R0 WITH TEST NUMBER
12274 047044 013701 047104      MOV      @#2$, R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
12275 047050 010605      MOV      SP, R5          ;SAVE SP
12276 047052 013704 000004      MOV      @#4, R4          ;SAVE T.O. VECTOR
12277 047056 013703 000336      MOV      @#336, R3      ;SAVE VECTOR AT 336
12278 047062 012737 047122 000004      MOV      @4$, @#4      ;GO TO 4$ ON OVFLW
12279 047070 012737 125252 000336      MOV      @125252, @#336 ;INIT. [336]
12280 047076 012706 000340      MOV      @340, SP      ;SET SP TO CAUSE RED ZONE TRAP
12281 047102 000257      CCC                      ;SCOPE SYNC
12282
12283 047104 010046      2$:      MOV      R0, -(SP)      ;FORCE RED ZONE TRAP - GO TO 4$
12284
12285 047106 010437 000004      MOV      R4, @#4          ;RESTORE T.O. VECTOR
12286 047112 010637 001074      MOV      SP, @#SREG5     ;SAVE BAD SP FOR PRINTING
12287 047116 010506      MOV      RS, SP          ;RESET SP FOR ERROR CALL
12288 047120 104005      3$:      ERROR      5          ;MOV FAILED TO CAUSE TRAP
12289
12290 047122 010437 000004      4$:      MOV      R4, @#4          ;RESTORE T.O. VECTOR
12291 047126 022706 000000      CMP      @0, SP          ;[SP]=0?
12292 047132 001404      BEQ      @6$, SP          ;BE IF YES
12293
12294 047134 010637 001074      MOV      SP, @#SREG5     ;SAVE BAD SP FOR PRINTING
12295 047140 010506      MOV      RS, SP          ;RESET SP FOR ERROR CALL
12296 047142 104005      5$:      ERROR      5          ;SP NOT BEING JAMMED TO 4
12297
12298 047144 022737 125252 000336 6$:      CMP      @125252, @#336 ;DID PUSH OCCUR IN YELLOW ZONE?
12299 047152 001404      BEQ      @8$, SP          ;BR IF NOT
12300
12301 047154 010637 001074      MOV      SP, @#SREG5     ;SAVE BAD SP FOR PRINTING
12302 047160 010506      MOV      RS, SP          ;RESET SP FOR ERROR CALL
12303 047162 104005      7$:      ERROR      5          ;MOV PUSHED INTO YELLOW ZONE
12304
12305 047164 010337 000336      8$:      MOV      R3, @#336      ;RESTORE VECTOR 336
12306 047170 010506      MOV      RS, SP          ;RESET SP
12307
12308      ;*****
12309      ;#TEST 663      YELLOW ZONE OVERFLOW TEST - MOV R,-(SP)
12310      ;*****
12311 047172      †ST663:

```



```

12312 047172 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12313 047174 012700 000663  MOV      #663,R0      ;:LOAD R0 WITH TEST NUMBER
12314 047200 013701 047232  MOV      @#25,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
12315 047204 010605          MOV      SP,R5        ;:SAVE SP
12316 047206 012702 000376  MOV      #376,R2      ;:R2 POINTS TO STACK
12317 047212 013704 000004  MOV      @#4,R4        ;:SAVE T.O. VECTOR
12318 047216 012737 047250 000004  MOV      #45,@#4 ;ON OVFLW - GO TO 4$
12319 047224 012706 000400  MOV      #400,SP      ;:SET SP TO CAUSE OVFLW
12320 047230 000257          CCC                    ;:SCOPE SYNC
12321
12322 047232 010046          2$:  MOV      RO,-(SP)  ;:FORCE STACK OVFLW - GO TO 4$
12323
12324 047234 010437 000004  MOV      R4,@#4        ;:RESTORE T.O. VECTOR
12325 047240 010637 001074  MOV      SP,@#SREG5    ;:SAVE BAD SP FOR PRINTING
12326 047244 010506          MOV      R5,SP        ;:RESET SP FOR ERROR CALL
12327 047246 104005          3$:  ERROR    5        ;:STACK OVFLW FAILED TO TRAP
12328
12329 047250 010437 000004  4$:  MOV      R4,@#4        ;:RESTORE T.O. VECTOR
12330 047254 020012          CMP      RO,(R2)      ;:DID (R0) GET PUSHED?
12331 047256 001404          BEQ      6$           ;:BR IF YES
12332
12333 047260 010637 001074  MOV      SP,@#SREG5    ;:SAVE BAD SP FOR PRINTING
12334 047264 010506          MOV      R5,SP        ;:RESET SP FOR ERROR CALL
12335 047266 104005          5$:  ERROR    5        ;:MOV FAILED TO PUSH IN YELLOW ZONE
12336
12337 047270 005706          6$:  TST      SP          ;:[SP]=0?
12338 047272 001004          BNE      8$           ;:BR IF NOT
12339
12340 047274 010637 001074  MOV      SP,@#SREG5    ;:SAVE BAD SP FOR PRINTING
12341 047300 010506          MOV      R5,SP        ;:RESET SP FOR ERROR CALL
12342 047302 104005          7$:  ERROR    5        ;:RED ZONE INSTEAD OF YELLOW ZONE
12343
12344 047304 010506          8$:  MOV      R5,SP        ;:RESET SP
12345
12346          ;:*****
12347          ;:TEST 664      YELLOW ZONE OVERFLOW TEST - (CMP RO,-(SP))
12348          ;:*****
12349 047306          TST664:
12350 047306 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12351 047310 012700 000664  MOV      #664,R0      ;:LOAD R0 WITH TEST NUMBER
12352 047314 013701 047342  MOV      @#25,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
12353 047320 010605          MOV      SP,R5        ;:SAVE THE SP
12354 047322 013704 000004  MOV      @#4,R4        ;:SAVE TRAP VECTOR
12355 047326 012737 047346 000004  MOV      #45,@#4 ;GO TO 4$ IF TRAP SPRUNG
12356 047334 012706 000400  MOV      #400,SP      ;:SET SP TO PUSH INTO "YELLOW ZONE"
12357 047340 000257          CCC                    ;:SCOPE SYNC
12358
12359 047342 020046          2$:  CMP      RO,-(SP)  ;:TEST THE CMP - NO TRAP SHOULD OCCUR
12360
12361 047344 000406          BR      6$           ;:GO TO EXIT TEST
12362
12363 047346 010437 000004  4$:  MOV      R4,@#4        ;:RESTORE TRAP VECTOR
12364 047352 010637 001074  MOV      SP,@#SREG5    ;:SAVE BAD SP FOR PRINTING
12365 047356 010506          MOV      R5,SP        ;:RESET THE SP
12366 047360 104005          3$:  ERROR    5        ;:CMP CAUSED OVERFLOW TRAP
12367
    
```

```

12368 047362 010437 000004      6$:  MOV    R4,R#4      ;RESTORE THE VECTOR
12369 047366 010505              MOV    R5,SP      ;RESET THE SP
12370
12371
12372
12373
12374 047370
12375 047370 000004
12376 047372 012700 000665      SCOPE              ;CALL THE SCOPE LOOP UTILITY
12377 047376 013701 047424      MOV    #665,R0    ;LOAD R0 WITH TEST NUMBER
12378 047402 010605              MOV    @#25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
12379 047404 013704 000004      MOV    SP,R5      ;SAVE THE SP
12380 047410 012737 047430 000004  MOV    @#4,R4     ;SAVE TRAP VECTOR
12381 047416 012706 000400      MOV    #45,@#4   ;GO TO 4$ IF TRAP SPRUNG
12382 047422 000257              CCC    #400,SP    ;SET SP TO PUSH INTO "YELLOW ZONE"
12383
12384 047424 030046      2$:  BIT    RO,-(SP)  ;TEST THE BIT - NO TRAP SHOULD OCCUR
12385
12386 047426 000406              BR     6$         ;GO TO EXIT TEST
12387
12388 047430 010437 000004      4$:  MOV    R4,R#4      ;RESTORE TRAP VECTOR
12389 047434 010637 001074      MOV    SP,@#5REGS ;SAVE BAD SP FOR PRINTING
12390 047440 010506              MOV    R5,SP      ;RESET THE SP
12391 047442 104005      3$:  ERROR  5         ;BIT CAUSED OVERFLOW TRAP
12392
12393 047444 010437 000004      6$:  MOV    R4,R#4      ;RESTORE THE VECTOR
12394 047450 010506              MOV    R5,SP      ;RESET THE SP
12395
12396
12397
12398
12399 047452
12400 047452 000004
12401 047454 012700 000666      SCOPE              ;CALL THE SCOPE LOOP UTILITY
12402 047460 013701 047506      MOV    #666,R0    ;LOAD R0 WITH TEST NUMBER
12403 047464 010605              MOV    @#25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
12404 047466 013704 000004      MOV    SP,R5      ;SAVE THE SP
12405 047472 012737 047512 000004  MOV    @#4,R4     ;SAVE TRAP VECTOR
12406 047500 012706 000400      MOV    #45,@#4   ;GO TO 4$ IF TRAP SPRUNG
12407 047504 000257              CCC    #400,SP    ;SET SP TO PUSH INTO "YELLOW ZONE"
12408
12409 047506 005746      2$:  TST    -(SP)     ;TEST THE TST - NO TRAP SHOULD OCCUR
12410
12411 047510 000406              BR     6$         ;GO TO EXIT TEST
12412
12413 047512 010437 000004      4$:  MOV    R4,R#4      ;RESTORE TRAP VECTOR
12414 047516 010637 001074      MOV    SP,@#5REGS ;SAVE BAD SP FOR PRINTING
12415 047522 010506              MOV    R5,SP      ;RESET THE SP
12416 047524 104006      3$:  ERROR  6         ;TST CAUSED OVERFLOW TRAP
12417
12418 047526 010437 000004      6$:  MOV    R4,R#4      ;RESTORE THE VECTOR
12419 047532 010506              MOV    R5,SP      ;RESET THE SP
12420
12421
12422
12423

```

```

12424 047534                                TST667:
12425 047534 000004                          SCOPE                                ;CALL THE SCOPE LOOP UTILITY
12426 047536 012700 000667                  MOV #667,R0                          ;LOAD R0 WITH TEST NUMBER
12427 047542 013701 047576                  MOV @#25,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
12428 047546 010605                          MOV SP,R5                             ;SAVE SP
12429 047550 010737 001010                  MOV PC,@#SLPERR                       ;SET ERROR LOOP ADDRESS
12430 047554 013704 000004 1$:             MOV @#4,R4                             ;SAVE T.O. VECTOR
12431 047560 012737 047606 000004          MOV @#4,@#4 ;ON ODD ADDR ERROR - GO TO 4$
12432 047566 010506                          MOV R5,SP                              ;RESET SP FOR ERROR LOOP
12433 047570 012702 000001                  MOV @#1,R2                             ;R2 GETS ODD ADDRESS
12434 047574 000257                          CCC                                     ;SCOPE SYNC
12435
12436 047576 160012 2$:                     SUB R0,(R2)                            ;FORCE ODD ADDR ERROR - GO TO 4$
12437
12438 047600 010437 000004                  MOV R4,@#4                             ;RESTORE T.O. VECTOR
12439 047604 104006 3$:                     ERROR 6                                ;ODD ADDR FAILED TO TRAP
12440
12441 047606 010437 000004 4$:             MOV R4,@#4                             ;RESTORE T.O. VECTOR
12442 047612 010506                          MOV R5,SP                              ;RESET SP
12443 047614 005037 000000                  CLR @#0                                ;CLR LOC. 0 JUST IN CASE
12444
12445 ;*****
12446 ;*TEST 670 TEST FOR ODD ADDR. ERROR TRAP FOR DEST. DEFERRED MODES
12447 ;*****
12448 047620                                TST670:
12449 047620 000004                          SCOPE                                ;CALL THE SCOPE LOOP UTILITY
12450 047622 012700 000670                  MOV #670,R0                          ;LOAD R0 WITH TEST NUMBER
12451 047626 013701 047650                  MOV @#25,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
12452 047632 012702 063243                  MOV @MBUF1+1,R2                       ;DEST ADDR=MBUF1+1 (ODD)
12453 047636 012737 047720 000004          MOV @#4,@#4 ;GO TO 4$ ON ODA TRAP
12454
12455 047644 010205                          MOV R2,R5                             ;[R5] = DEST. ADDR
12456 047646 000257                          CCC                                     ;SCOPE SYNC
12457
12458 047650 105435 2$:                     NEGB @ (R5)+                          ;TEST DM=3 TRAP
12459
12460 047652 104006 3$:                     ERROR 6                                ;ODA TRAP NOT SPRUNG
12461
12462 047654 012705 063245                  MOV @MBUF1+3,R5                       ;[R5] = DEST. ADDR
12463 047660 013701 047666                  MOV @#205,R1                          ;[R1] = TEST INSTR
12464 047664 000257                          CCC                                     ;SCOPE SYNC
12465
12466 047666 105455 20$:                    NEGB @-(R5)                            ;TEST DM=5 TRAP
12467
12468 047670 104006 5$:                     ERROR 6                                ;ODA TRAP NOT SPRUNG
12469
12470 047672 010205                          MOV R2,R5                             ;[R5] = DEST ADDR
12471 047674 013701 047702                  MOV @#215,R1                          ;[R1] = TEST INSTR
12472 047700 000257                          CCC                                     ;SCOPE SYNC
12473
12474 047702 105475 000000 21$:            NEGB @0(R5)                            ;TEST DM=7 TRAP
12475
12476 047706 104006 7$:                     ERROR 6                                ;ODA TRAP NOT SPRUNG
12477
12478 047710 012737 061144 000004          MOV @BERR,@#4                          ;RESET T.O. VECTOR
12479 047716 000403                          BR TST671                              ;GO TO SCOPE EXIT

```

E04

```

12480
12481 047720 062716 000002      4$:  ADD      #2,(SP)          ;MOV RETURN PC AROUND ERROR CALL
12482 047724 000002              RTI          ;RETURN TO NEXT SUB-TEST
12483
12484
12485      ;*****
12486      ;*TEST 671      TEST FOR ODD ADDR ERROR TRAP FOR SOURCE DEFERRED MODES
12487      ;*****
12488      TST671:
12489      SCOPE          ;CALL THE SCOPE LOOP UTILITY
12490      MOV      #671,R0      ;:LOAD R0 WITH TEST NUMBER
12491      MOV      @#25,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
12492      MOV      #MBUF1+1,R2  ;:[R2] = SOURCE ADDR. (ODD)
12493      MOV      #4$,@#4 ;GO TO 4$ ON TRAP
12494      MOV      R2,R5          ;:[R5] = SOURCE ADDR.
12495      CCC              ;SCOPE SYNC
12496
12497 047756 113504      2$:  MOVB      @ (R5)+,R4      ;TEST SM=3
12498
12499 047760 104006      3$:  ERROR      6              ;ODA TRAP NOT SPRUNG
12500
12501 047762 012705 063245      MOV      #MBUF1+3,R5      ;:[R5] = SOURCE ADDR
12502 047766 013701 047774      MOV      @#20$,R1        ;:[R1] = TEST INSTR
12503 047772 000257      CCC              ;SCOPE SYNC
12504
12505 047774 115504      20$: MOVB      @-(R5),R4      ;TEST SM=5
12506
12507 047776 104006      5$:  ERROR      6              ;ODA TRAP NOT SPRUNG
12508 050000 010205      MOV      R2,R5          ;:[R5] = SOURCE ADDR
12509 050002 013701 050010      MOV      @#21$,R1        ;:[R1] = TEST INSTR
12510 050006 000257      CCC              ;SCOPE SYNC
12511
12512 050010 117504 000000      21$: MOVB      @0(R5),R4    ;TEST SM=7
12513
12514 050014 104006      7$:  ERROR      6              ;ODA TRAP NOT SPRUNG
12515
12516 050016 012737 061144 000004      MOV      #BERR,@#4      ;RESET T.O. VECTOR
12517 050024 000403      BR      TST672          ;:GO TO SCOPE EXIT
12518
12519 050026 062716 000002      4$:  ADD      #2,(SP)          ;MOVE RETURN PC AROUND ERROR CALL
12520 050032 000002              RTI          ;RETURN TO NEXT SUB-TEST
12521
12522      ;*****
12523      ;*TEST 672      TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES
12524      ;*****
12525      TST672:
12526      SCOPE          ;CALL THE SCOPE LOOP UTILITY
12527      MOV      #672,R0      ;:LOAD R0 WITH TEST NUMBER
12528      MOV      @#25,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
12529      MOV      #6$,R2        ;DEST ADDR = 6$+3 (ODD)
12530      MOV      #4$,@#4 ;GO TO 4$ ON ODA TRAP
12531
12532      MOV      R2,R5          ;:[R5] = DEST ADDR
12533      CCC              ;SCOPE SYNC
12534
12535 050064 000135      2$:  JMP      @ (R5)+          ;TEST JMP DM=3
    
```

F04

MAINDEC-11-DQKDA-A K011-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T672

MACY11 27(1006) 08-FEB-77 16:23 PAGE 228
 TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES

```

12536
12537 050066 104006      3$:  ERROR 6          ;ODA TRAP NOT SPRUNG IN ROM LOC 153
12538
12539 050070 012705 050137      MOV  #6$+3,R5        ;[R5] = DEST ADDR
12540 050074 013701 050102      MOV  @#20$,R1        ;[R1] = TEST INSTR
12541 050100 000257      CCC                    ;SCOPE SYNC
12542
12543 050102 000155      20$: JMP  @-(R5)      ;TEST JMP DM=5
12544
12545 050104 104006      5$:  ERROR 6          ;ODA TRAP NOT SPRUNG IN ROM LOC 155
12546
12547 050106 010205      MOV  R2,R5          ;[R5] = DEST ADDR
12548 050110 013701 050116      MOV  @#21$,R1        ;[R1] = TEST INSTR
12549 050114 000257      CCC                    ;SCOPE SYNC
12550
12551 050116 000175 000000      21$: JMP  @0(R5)     ;TEST JMP DM=7
12552
12553 050122 104006      7$:  ERROR 6          ;ODA TRAP NOT SPRUNG
12554
12555 050124 012737 061144 000004      MOV  #BERR,@#4      ;RESET BUS T.O. VECTOR
12556 050132 000420      BR    TST673        ;;GO TO SCOPE EXIT
12557
12558 050134 000000      6$:  HALT              ;CATASTOPHIC ERROR - [PC] QUESTIONABLE.
12559 050136 000000      HALT              ;RESTART PROGRAM - DO NOT CONTINUE.
12560 050140 000000      HALT
12561
12562 050142 032716 000001      4$:  BIT  #1,(SP)     ;TRAP DUE TO ODD PC?
12563 050146 001003      BNE  8$            ;BR IF YES
12564 050150 062716 000002      ADD  #2,(SP)       ;MOV RETURN PC AROUND ERROR CALL
12565 050154 000002      RTI                ;RETURN TO NEXT SUB TEST
12566
12567 050156 011603      8$:  MOV  (SP),R3     ;GET ODD PC OFF STACK INTO R3
12568 050160 062706 000004      ADD  #4,SP         ;FIX SP
12569
12570 050164 104007      9$:  ERROR 7          ;PC TRAPPED WITH ODD ADDRESS
12571
12572 050166 012737 061144 000004      MOV  #BERR,@#4      ;RESET T.O. VECTOR
12573
12574
12575
12576
12577 050174
12578 050174 000004      ;*****
12579 050176 012700 000673      ;*TEST 673 TEST FOR STACK OFLW FOR DEST MODES 1,2,4, AND 6.
12580 050202 013701 050226      ;*****
12581 050206 012737 050340 000004      ;TST673:
12582 050214 010605      SCOPE                ;CALL THE SCOPE LOOP UTILITY
12583 050216 012702 000376      MOV  #673,R0        ;LOAD R0 WITH TEST NUMBER
12584
12585 050222 010206      MOV  @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
12586 050224 000257      MOV  #4$,@#4 ;GO TO 4$ ON OVFLW TRAP
12587
12588 050226 005016      MOV  SP,R5          ;SAVE SP
12589
12590 050230 010637 001074      MOV  #376,R2        ;USE R2 TO SET UP SP TO CAUSE TRAP
12591 050234 010506      MOV  R2,SP         ;SET UP SP TO CAUSE OVERFLOW
12592
12593
12594
12595
12596
12597
12598
12599
12600
12601
12602
12603
12604
12605
12606
12607
12608
12609
12610
12611
12612
12613
12614
12615
12616
12617
12618
12619
12620
12621
12622
12623
12624
12625
12626
12627
12628
12629
12630
12631
12632
12633
12634
12635
12636
12637
12638
12639
12640
12641
12642
12643
12644
12645
12646
12647
12648
12649
12650
12651
12652
12653
12654
12655
12656
12657
12658
12659
12660
12661
12662
12663
12664
12665
12666
12667
12668
12669
12670
12671
12672
12673
12674
12675
12676
12677
12678
12679
12680
12681
12682
12683
12684
12685
12686
12687
12688
12689
12690
12691
12692
12693
12694
12695
12696
12697
12698
12699
12700
12701
12702
12703
12704
12705
12706
12707
12708
12709
12710
12711
12712
12713
12714
12715
12716
12717
12718
12719
12720
12721
12722
12723
12724
12725
12726
12727
12728
12729
12730
12731
12732
12733
12734
12735
12736
12737
12738
12739
12740
12741
12742
12743
12744
12745
12746
12747
12748
12749
12750
12751
12752
12753
12754
12755
12756
12757
12758
12759
12760
12761
12762
12763
12764
12765
12766
12767
12768
12769
12770
12771
12772
12773
12774
12775
12776
12777
12778
12779
12780
12781
12782
12783
12784
12785
12786
12787
12788
12789
12790
12791
12792
12793
12794
12795
12796
12797
12798
12799
12800
12801
12802
12803
12804
12805
12806
12807
12808
12809
12810
12811
12812
12813
12814
12815
12816
12817
12818
12819
12820
12821
12822
12823
12824
12825
12826
12827
12828
12829
12830
12831
12832
12833
12834
12835
12836
12837
12838
12839
12840
12841
12842
12843
12844
12845
12846
12847
12848
12849
12850
12851
12852
12853
12854
12855
12856
12857
12858
12859
12860
12861
12862
12863
12864
12865
12866
12867
12868
12869
12870
12871
12872
12873
12874
12875
12876
12877
12878
12879
12880
12881
12882
12883
12884
12885
12886
12887
12888
12889
12890
12891
12892
12893
12894
12895
12896
12897
12898
12899
12900
12901
12902
12903
12904
12905
12906
12907
12908
12909
12910
12911
12912
12913
12914
12915
12916
12917
12918
12919
12920
12921
12922
12923
12924
12925
12926
12927
12928
12929
12930
12931
12932
12933
12934
12935
12936
12937
12938
12939
12940
12941
12942
12943
12944
12945
12946
12947
12948
12949
12950
12951
12952
12953
12954
12955
12956
12957
12958
12959
12960
12961
12962
12963
12964
12965
12966
12967
12968
12969
12970
12971
12972
12973
12974
12975
12976
12977
12978
12979
12980
12981
12982
12983
12984
12985
12986
12987
12988
12989
12990
12991
12992
12993
12994
12995
12996
12997
12998
12999
13000
    
```

```

12592 050236 104006          3$:  ERROR  6          ;DM1 FAILED TO CAUSE OVERFLOW TRAP
12593
12594 050240 013701 050250      MOV  2#20$,R1      ;[R1] = TEST INSTR.
12595 050244 010206          MOV  R2,SP        ;SET UP SP TO CAUSE OVERFLOW
12596 050246 000257          CCC              ;SCOPE SYNC
12597
12598 050250 005026          20$: CLR  (SP)+    ;TEST DM2 - SHOULD SPRING TRAP
12599
12600 050252 010637 001074      MOV  SP,2#$REG5   ;SAVE BAD SP FOR PRINTING
12601 050256 010506          MOV  R5,SP        ;RESET SP
12602 050260 104006          5$:  ERROR  6          ;DM2 FAILED TO CAUSE OVERFLOW TRAP
12603
12604 050262 013701 050272      MOV  2#21$,R1     ;[R1] = TEST INSTR.
12605 050266 010206          MOV  R2,SP        ;SET UP SP TO CAUSE OVERFLOW
12606 050270 000257          CCC              ;SCOPE SYNC
12607
12608 050272 005046          21$: CLR  -(SP)    ;TEST DM4 - SHOULD SPRING TRAP
12609
12610 050274 010637 001074      MOV  SP,2#$REG5   ;SAVE BAD SP FOR PRINTING
12611 050300 010506          MOV  R5,SP        ;RESET SP
12612 050302 104006          7$:  ERROR  6          ;DM4 FAILED TO CAUSE OVERFLOW TRAP
12613
12614 050304 013701 050314      MOV  2#22$,R1     ;[R1] = TEST INSTR.
12615 050310 010206          MOV  R2,SP        ;SET SP TO CAUSE ERROR
12616 050312 000257          CCC              ;SCOPE SYNC
12617
12618 050314 005066 000000      22$: CLR  0(SP)    ;TEST DM6 - SHOULD SPRING TRAP
12619
12620 050320 010637 001074      MOV  SP,2#$REG5   ;SAVE BAD SP FOR PRINTING
12621 050324 010506          MOV  R5,SP        ;RESET SP
12622 050326 104006          9$:  ERROR  6          ;DM6 FAILED TO CAUSE OVERFLOW TRAP
12623
12624 050330 012737 061144 000004  MOV  #BERR,2#4    ;RESET BUS T.O. VECTOR
12625 050336 000407          BR   TST674       ;GO TO SCOPE EXIT
12626
12627 050340 011604          4$:  MOV  (SP),R4   ;GET RETURN PC OFF STACK
12628 050342 062704 000010      ADD  #10,R4       ;MOVE RETURN PC AROUND ERROR CALL
12629 050346 010506          MOV  R5,SP        ;RESET SP
12630 050350 005046          CLR  -(SP)        ;PUSH NEW PS ON STACK
12631 050352 010446          MOV  R4,-(SP)     ;PUSH RETURN PC ON STACK
12632 050354 000002          RTI              ;RETURN TO NEXT SUB-TEST
12633
12634
12635
12636
12637 050356
12638 050356 000004          TST674:          ;CALL THE SCOPE LOOP UTILITY
12639 050360 012700 000674      MOV  #674,R0      ;LOAD R0 WITH TEST NUMBER
12640 050364 013701 050410      MOV  2#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
12641 050370 012737 050522 000004  MOV  #4$,2#4 ;GO TO 4$ ON STACK OVFLW TRAP
12642 050376 010605          MOV  SP,R5        ;SAVE SP
12643 050400 012702 000376      MOV  #376,R2     ;USE R2 TO SET UP SP TO CAUSE TRAP
12644
12645 050404 010206          MOV  R2,SP        ;SET UP SP TO CAUSE OVERFLOW
12646 050406 000257          CCC              ;SCOPE SYNC
12647

```

```

;*****
;TEST 674 TEST FOR STACK OVFLW FOR MOV DEST MODES 1,2,4, AND 6.
;*****

```

H04

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T674

MACY11 27(1006) 08-FEB-77 16:23 PAGE 230
 TEST FOR STACK OVFLW FOR MOV DEST MODES 1,2,4, AND 6.

```

12648 050410 010016      25:  MOV      RD,(SP)          ;TEST MOV DM1 - SHOULD SPRING TRAP
12649
12650 050412 010637 001074      MOV      SP,@#SREG5      ;SAVE BAD SP FOR PRINTING
12651 050416 010506          MOV      R5,SP          ;RESET SP
12652 050420 104006      35:  ERROR      6          ;MOV DM1 FAILED TO SPRING TRAP
12653
12654 050422 013701 050432      MOV      @#20$,R1        ;[R1] = TEST INSTR.
12655 050426 010206          MOV      R2,SP          ;SET UP SP TO CAUSE OVERFLOW
12656 050430 000257          CCC                    ;SCOPE SYNC
12657
12658 050432 010026      205:  MOV      RD,(SP)+        ;TEST MOV DM2 - SHOULD SPRING TRAP
12659
12660 050434 010637 001074      MOV      SP,@#SREG5      ;SAVE BAD SP FOR PRINTING
12661 050440 010506          MOV      R5,SP          ;RESET SP
12662 050442 104006      55:  ERROR      6          ;MOV DM2 FAILED TO SPRING TRAP
12663
12664 050444 013701 050454      MOV      @#21$,R1        ;[R1] = TEST INSTR.
12665 050450 010206          MOV      R2,SP          ;SET UP SP TO CAUSE OVERFLOW
12666 050452 000257          CCC                    ;SCOPE SYNC
12667
12668 050454 010046      215:  MOV      RD,-(SP)        ;TEST MOV DM4 - SHOULD SPRING TRAP
12669
12670 050456 010637 001074      MOV      SP,@#SREG5      ;SAVE BAD SP FOR PRINTING
12671 050462 010506          MOV      R5,SP          ;RESET SP
12672 050464 104006      75:  ERROR      6          ;MOV DM4 FAILED TO SPRING TRAP
12673
12674 050466 013701 050476      MOV      @#22$,R1        ;[R1] = TEST INSTR.
12675 050472 010206          MOV      R2,SP          ;SET UP SP TO CAUSE OVERFLOW
12676 050474 000257          CCC                    ;SCOPE SYNC
12677
12678 050476 010066 000000      225:  MOV      RD,0(SP)        ;TEST MOV DM6 - SHOULD SPRING TRAP
12679
12680 050502 010637 001074      MOV      SP,@#SREG5      ;SAVE BAD SP FOR PRINTING
12681 050506 010506          MOV      R5,SP          ;RESET SP
12682 050510 104006      95:  ERROR      6          ;MOV DM6 FAILED TO CAUSE OVFLW TRAP
12683
12684 050512 012737 061144 000004      MOV      #BERR,@#4      ;RESET T.O. VECTOR
12685 050520 000407          BR       TST675          ;GO TO SCOPE EXIT
12686
12687 050522 011604      45:  MOV      (SP),R4         ;GET RETURN PC
12688 050524 062704 000010      ADD      #10,R4          ;MOVE RETURN PC AROUND ERROR CALL
12689 050530 010506          MOV      R5,SP          ;RESET SP
12690 050532 005046          CLR      -(SP)          ;PUSH NEW PSW
12691 050534 010446          MOV      R4,-(SP)        ;PUSH RETURN PC
12692 050536 000002          RTI                    ;RETURN TO NEXT SUB-TEST
12693
12694
12695
12696
12697 050540      ;*****
12698 050540 000004      ;*TEST 675 TEST THAT JSR CAN CAUSE OVERFLOW TRAP
12699 050542 012700 000675      ;*****
12700 050546 013701 050570      ;*****
12701 050552 012737 050612 000004      ;*****
12702 050560 010605          ;*****
12703 050562 012706 000400          ;*****
12704
12705
12706
12707
12708
12709
12710
12711
12712
12713
12714
12715
12716
12717
12718
12719
12720
12721
12722
12723
12724
12725
12726
12727
12728
12729
12730
12731
12732
12733
12734
12735
12736
12737
12738
12739
12740
12741
12742
12743
12744
12745
12746
12747
12748
12749
12750
12751
12752
12753
12754
12755
12756
12757
12758
12759
12760
12761
12762
12763
12764
12765
12766
12767
12768
12769
12770
12771
12772
12773
12774
12775
12776
12777
12778
12779
12780
12781
12782
12783
12784
12785
12786
12787
12788
12789
12790
12791
12792
12793
12794
12795
12796
12797
12798
12799
12800
12801
12802
12803
12804
12805
12806
12807
12808
12809
12810
12811
12812
12813
12814
12815
12816
12817
12818
12819
12820
12821
12822
12823
12824
12825
12826
12827
12828
12829
12830
12831
12832
12833
12834
12835
12836
12837
12838
12839
12840
12841
12842
12843
12844
12845
12846
12847
12848
12849
12850
12851
12852
12853
12854
12855
12856
12857
12858
12859
12860
12861
12862
12863
12864
12865
12866
12867
12868
12869
12870
12871
12872
12873
12874
12875
12876
12877
12878
12879
12880
12881
12882
12883
12884
12885
12886
12887
12888
12889
12890
12891
12892
12893
12894
12895
12896
12897
12898
12899
12900
12901
12902
12903
12904
12905
12906
12907
12908
12909
12910
12911
12912
12913
12914
12915
12916
12917
12918
12919
12920
12921
12922
12923
12924
12925
12926
12927
12928
12929
12930
12931
12932
12933
12934
12935
12936
12937
12938
12939
12940
12941
12942
12943
12944
12945
12946
12947
12948
12949
12950
12951
12952
12953
12954
12955
12956
12957
12958
12959
12960
12961
12962
12963
12964
12965
12966
12967
12968
12969
12970
12971
12972
12973
12974
12975
12976
12977
12978
12979
12980
12981
12982
12983
12984
12985
12986
12987
12988
12989
12990
12991
12992
12993
12994
12995
12996
12997
12998
12999
13000

```

```

12704 050566 000257          CCC          ;SCOPE SYNC
12705
12706 050570 004737 050616    2$:  JSR      PC,2#6$ ;TEST JSR - SHOULD SPRING TRAP
12707
12708 050574 010637 001074    MOV      SP,2#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
12709 050600 010506          MOV      R5,SP      ;RESET SP
12710 050602 104005          3$:  ERROR   5      ;JSR PUSH DID NOT SPRING OVFL TRAP
12711
12712 050604 000410          BR       8$        ;GO TO SCOPE EXIT
12713
12714 050606 010637 001074    MOV      SP,2#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
12715 050612 010506          4$:  MOV      R5,SP      ;RESET SP
12716 050614 000404          BR       8$        ;GO EXIT TEST - ALL OK
12717
12718 050616 010637 001074    6$:  MOV      SP,2#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
12719 050622 010506          MOV      R5,SP      ;RESET SP
12720 050624 104005          5$:  ERROR   5      ;JSR PUSH FAILED TO SPRING OVFLW TRAP
12721
12722 050626 012737 061144 000004 8$:  MOV      #BERR,2#4 ;RESET BUS T.O. VECTOR
12723
12724          ;*****
12725          ;*TEST 676      TEST THAT 1ST PUSH IN TRAP MICROROUTINE CAUSES OVFLW TRAP
12726          ;*****
12727          †ST676:
12728 050634 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12729 050636 012700 000676    MOV      #676,R0   ;LOAD R0 WITH TEST NUMBER
12730 050642 013701 050676    MOV      2#2$,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD
12731 050646 013704 000014    MOV      2#14,R4  ;SAVE BREAK POINT TRAP VECTOR
12732 050652 010605          MOV      SP,R5    ;SAVE SP
12733 050654 012737 050712 000004 MOV      #4$,2#4 ;GO TO 4$ ON OVFLW TRAP
12734 050662 012737 050716 000014 MOV      #6$,2#14 ;GO TO 6$ IF BPT SERVICED
12735 050670 012706 000400    MOV      #400,SP  ;SET UP SP TO CAUSE OVFLW ON 1ST PUSH
12736 050674 000257          CCC          ;SCOPE SYNC
12737
12738 050676 000003          2$:  BPT          ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
12739
12740 050700 010637 001074    MOV      SP,2#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
12741 050704 010506          MOV      R5,SP      ;RESET SP
12742 050706 104005          3$:  ERROR   5      ;BPT FAILED TO TRAP
12743
12744 050710 000406          BR       8$        ;GO TO SCOPE EXIT
12745
12746 050712 010506          4$:  MOV      R5,SP      ;RESET SP
12747 050714 000404          BR       8$        ;GO EXIT - ALL OK
12748
12749 050716 010637 001074    6$:  MOV      SP,2#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
12750 050722 010506          MOV      R5,SP      ;RESET SP
12751 050724 104005          5$:  ERROR   5      ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
12752
12753 050726 012737 061144 000004 8$:  MOV      #BERR,2#4 ;RESET VECTORS
12754 050734 010437 000014    MOV      R4,2#14
12755
12756          ;*****
12757          ;*TEST 677      TEST THAT 2ND PUSH IN TRAP MICROUTINE CAUSES OVFLW TRAP
12758          ;*****
12759 050740          †ST677:

```



```

12760 050740 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12761 050742 012700 000677  MOV          #677,R0      ;LOAD R0 WITH TEST NUMBER
12762 050746 013701 051002  MOV          @#25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
12763 050752 013704 000014  MOV          @#14,R4     ;SAVE BPT VECTOR
12764 050756 010605          MOV          SP,R5       ;SAVE SP
12765 050760 012737 051016 000004  MOV          #45,@#4 ;GO TO 4$ ON STACK OVFLOW
12766 050766 012737 051022 000014  MOV          #65,@#14 ;GO TO 6$ IF BPT SERVICED
12767 050774 012706 000402  MOV          #402,SP    ;SET SP TO CAUSE TRAP ON 2ND PUSH
12768 051000 000257          CCC                   ;SCOPE SYNC
12769
12770 051002 000003          2$: BPT              ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
12771
12772 051004 010637 001074  MOV          SP,@#SREG5  ;SAVE BAD SP FOR PRINTING
12773 051010 010506          MOV          R5,SP      ;RESET SP
12774 051012 104005          3$: ERROR          5    ;BPT FAILED TO TRAP
12775
12776 051014 000406          BR           8$        ;GO TO SCOPE EXIT
12777
12778 051016 010506          4$: MOV          R5,SP  ;RESET SP
12779 051020 000404          BR           8$        ;GO EXIT - ALL OK
12780
12781 051022 010637 001074  6$: MOV          SP,@#SREG5 ;SAVE BAD SP FOR PRINTING
12782 051026 010506          MOV          R5,SP      ;RESET SP
12783 051030 104005          5$: ERROR          5    ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
12784
12785 051032 012737 061144 000004  8$: MOV          #BERR,@#4 ;RESET VECTORS
12786 051040 010437 000014  MOV          R4,@#14
12787
12788 ;*****
12789 ;*TEST 700 ILLEGAL INSTRUCTION TEST - JSR RN,%R
12790 ;*****
12791 ;*ST700:
12792 051044 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12793 051046 012700 000700  MOV          #700,R0     ;LOAD R0 WITH TEST NUMBER
12794 051052 013701 051106  MOV          @#25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
12795 051056 010605          MOV          SP,R5       ;SAVE SP
12796 051060 010737 001010  MOV          PC,@#SLPERR ;SET ERROR LOOP ADDRESS
12797 051064 013704 000004  1$: MOV          @#4,R4   ;SAVE T.O. VECTOR
12798 051070 012737 051116 000004  MOV          #45,@#4 ;ILLEGAL INSTR. TRAP GOES TO 4$
12799 051076 010506          MOV          R5,SP      ;RESET SP FOR ERROR LOOP
12800 051100 012702 051114  MOV          #35,R2     ;IN CASE JSR JUMPS TO [R2]
12801 051104 000257          CCC                   ;SCOPE SYNC
12802
12803 051106 004302          2$: JSR          R3,R2  ;JSR MODE 0 FORCES TRAP - GO TO 4$
12804
12805 051110 010437 000004  3$: MOV          R4,@#4  ;RESTORE T.O. VECTOR
12806 051114 104005          ERROR          5    ;JSR FAILED TO SPRING TRAP
12807
12808 051116 010437 000004  4$: MOV          R4,@#4  ;RESTORE VECTOR
12809 051122 010506          MOV          R5,SP      ;RESET SP
12810
12811 ;*****
12812 ;*TEST 701 ILLEGAL INSTRUCTION TEST - JMP %R
12813 ;*****
12814 ;*ST701:
12815 051124 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY

```

K04

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T701

MACY11 27(1006) 08-FEB-77 16:23 PAGE 233
 ILLEGAL INSTRUCTION TEST - JMP %R

```

12816 051126 012700 000701      MOV      #701,R0      ;:LOAD R0 WITH TEST NUMBER
12817 051132 013701 051166      MOV      @#2$,R1     ;:LOAD R1 WITH TEST INSTRUCTION WORD
12818 051136 010605              MOV      SP,R5       ;:SAVE SP
12819 051140 010737 001010      MOV      PC,@#SLPERR ;:SET ERROR LOOP ADDRESS
12820 051144 013704 000004      MOV      @#4,R4      ;:SAVE VECTOR POINTER AT LOC. 4
12821 051150 012737 051176 000004 1$:  MOV      @#4,@#4 ;ON TRAP - GO TO 4$
12822 051156 010506              MOV      R5,SP       ;:RESET SP FOR ERROR LOOP
12823 051160 012702 051174      MOV      @#3$,R2     ;:IN CASE IT JUMPS TO ADDR IN RN
12824 051164 000257              CCC                 ;:SCOPE SYNC
12825
12826 051166 000102              2$:  JMP      R2         ;:JMP MODE 0 FORCES TRAP - GO TO 4$
12827
12828 051170 010437 000004      MOV      R4,@#4     ;:RESTORE VECTOR POINTER AT LOC. 4
12829 051174 104005              3$:  ERROR      5      ;:ILLEGAL INSTR TRAP FAILED
12830
12831 051176 010437 000004      4$:  MOV      R4,@#4     ;:RESTORE VECTOR POINTER AT LOC. 4
12832 051202 010506      MOV      R5,SP       ;:RESET SP
12833
12834      ;:*****
12835      ;:*TEST 702      BUS TIMEOUT TRAP TEST - TST (R)
12836      ;:*****
12837 051204      TST702:
12838 051204 000004      SCOPE
12839 051206 012700 000702      MOV      #702,R0     ;:CALL THE SCOPE LOOP UTILITY
12840 051212 013701 051246      MOV      @#2$,R1     ;:LOAD R0 WITH TEST NUMBER
12841 051216 010605              MOV      SP,R5       ;:LOAD R1 WITH TEST INSTRUCTION WORD
12842 051220 010737 001010      MOV      PC,@#SLPERR ;:SAVE SP
12843 051224 013704 000004      MOV      @#4,R4      ;:SET ERROR LOOP ADDRESS
12844 051230 012737 051256 000004 1$:  MOV      @#4,R4      ;:SAVE ORIGINAL T.O. VECTOR POINTER
12845 051236 012702 160000      MOV      @#4,@#4 ;ON T.O. TRAP - GO TO 4$
12846 051242 010506      MOV      @#160000,R2 ;:ADDRESS CAUSES T.O.
12847 051244 000257      MOV      R5,SP       ;:RESET SP FOR ERROR LOOP
12848              CCC                 ;:SCOPE SYNC
12849 051246 005712              2$:  TST      (R2)      ;:FORCE T.O. TRAP - GO TO 4$
12850
12851 051250 010437 000004      MOV      R4,@#4     ;:RESTORE T.O. VECTOR
12852 051254 104005              3$:  ERROR      5      ;:TIMEOUT TRAP FAILED
12853 051256 010437 000004      4$:  MOV      R4,@#4     ;:RESTORE T.O. VECTOR
12854 051262 010506      MOV      R5,SP       ;:RESET SP
12855
12856      ;:*****
12857      ;:*TEST 703      "T" BIT TRAP TEST
12858      ;:*****
12859 051264      TST703:
12860 051264 000004      SCOPE
12861 051266 012700 000703      MOV      #703,R0     ;:CALL THE SCOPE LOOP UTILITY
12862 051272 013701 051330      MOV      @#2$,R1     ;:LOAD R0 WITH TEST NUMBER
12863 051276 010605              MOV      SP,R5       ;:LOAD R1 WITH TEST INSTRUCTION WORD
12864 051300 010737 001010      MOV      PC,@#SLPERR ;:SAVE SP
12865 051304 010506      MOV      R5,SP       ;:SET ERROR LOOP ADDRESS
12866 051306 012737 051336 000014 1$:  MOV      @#4,@#14    ;:RESET SP FOR ERROR LOOP
12867 051314 012746 000020      MOV      @#20,-(SP)  ;:GO TO 4$ WHEN "T" TRAP SPRUNG
12868 051320 012746 051330      MOV      @#25,-(SP)  ;:SET "T" BIT ON STACK
12869 051324 000257      CCC                 ;:SET UP NEW PC ON STACK
12870 051326 000006      RTT                 ;:SCOPE SYNC
12871              ;:TURN ON "T" BIT - GO TO 2$

```

```

12872 051330 005700      2S:   TST      R0          ;SPRING "T" BIT TRAP - GO TO 4S
12873
12874 051332 104005      3S:   ERROR    5          ;NO "T" BIT TRAP OCCURRED
12875
12876 051334 000405      BR      6S          ;GO EXIT
12877
12878 051336 032766 000020 000002 4S:   BIT      #20,2(SP)    ;"T" BIT SET IN OLD PSW?
12879 051344 001001      BNE     6S          ;BR IF YES
12880
12881 051346 104001      5S:   ERROR    1          ;#T# BIT NOT SAVED ON STACK
12882
12883 051350 012737 000016 000014 6S:   MOV      #16,#14     ;RESTORE "T" BIT TRAP CATCHER
12884 051356 005037 000016      CLR     #16
12885 051362 010506      MOV     RS,SP       ;RESET SP
12886
12887
12888
12889
12890
12891
12892 051364
12893 051364 000004
12894 051366 012700 000704
12895 051372 013701 051416
12896 051376 010605
12897 051400 013704 000004
12898 051404 012737 051440 000004
12899 051412 005006
12900 051414 000257
12901
12902 051416 012746 007777      2S:   MOV      #7777,-(SP) ;ATTEMPT PUSH INTO PSW - SHOULD CAUSE
12903
12904
12905 051422 010437 000004      MOV     R4,#4       ;RESTORE BUS ERROR VECTOR
12906 051426 005004      CLR     R4          ;[R4] = S / B SP
12907 051430 010603      MOV     SP,R3       ;[R3] = WAS SP
12908 051432 010506      MOV     RS,SP       ;RESET THE SP
12909 051434 104003      3S:   ERROR    3          ;TRAP NOT SPRUNG
12910 051436 000414      BR      TST705     ;GO TO SCOPE EXIT - SCHOOL'S OUT
12911
12912 051440 022706 000000      4S:   CMP      #0,SP      ;WAS IT A RED ZONE TRAP ?
12913 051444 001406      BEQ     6S          ;BR IF YES
12914
12915 051446 010437 000004      MOV     R4,#4       ;RESTORE BUS ERROR VECTOR
12916 051452 005004      CLR     R4          ;[R4] = S / B SP
12917 051454 010603      MOV     SP,R3       ;[R3] = WAS SP
12918 051456 010506      MOV     RS,SP       ;RESET THE SP
12919 051460 104003      5S:   ERROR    3          ;TRAP SPRUNG BUT NOT RED ZONE
12920
12921 051462 010506      6S:   MOV     RS,SP       ;FIX UP THE SP
12922 051464 010437 000004      MOV     R4,#4       ;RESTORE BERR VECTOR
12923
12924
12925
12926
12927 051470

```

;TEST 704 TEST PUSH INTO PSW WITH [SP] = 000000
;THESE NEXT TWO TESTS VERIFY THAT A "RED ZONE" TRAP OCCURS IF A
;PUSH IS ATTEMPTED WITH THE [SP] INITIALLY EQUAL TO 000000,177572,

†T704:

SCOPE
MOV #704,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #2S,R1 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #4,R4 ;SAVE THE SP
MOV #4S,#4 ;SAVE THE BUS ERROR VECTOR
CLR SP ;"RED ZONE" TRAP GOES TO 4S
CCC ;MAKE SP = 000000
;SCOPE SYNC

;TEST 705 TEST PUSH INTO SR WITH [SP] = 177572

†T705:

M04

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17 T705

MACY11 27(1006) 08-FEB-77 16:23 PAGE 235
 TEST PUSH INTO SR WITH [SP] = 177572

```

12928 051470 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12929 051472 012700 000705  MOV      #705,R0    ;LOAD R0 WITH TEST NUMBER
12930 051476 013701 051524  MOV      @#25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
12931 051502 010605          MOV      SP,R5      ;SAVE THE SP
12932 051504 013704 000004  MOV      @#4,R4      ;SAVE THE BUS ERROR VECTOR
12933 051510 012737 051546 000004  MOV      #45,@#4 ;"RED ZONE" TRAP GOES TO 45
12934 051516 012706 177572  MOV      #177572,SP ;MAKE SP=177572
12935 051522 000257          CCC              ;SCOPE SYNC
12936
12937 051524 012746 177777 25:     MOV      #-1,-(SP) ;ATTEMPT PUSH INTO SR - SHOULD CAUSE
12938                                     ;"RED ZONE" TRAP TO BE SPRUNG
12939
12940 051530 010437 000004          MOV      R4,@#4    ;RESTORE BUS ERROR VECTOR
12941 051534 005004          CLR      R4        ;[R4] = S / B SP
12942 051536 010603          MOV      SP,R3     ;[R3] = WAS SP
12943 051540 010506          MOV      R5,SP     ;RESET THE SP
12944 051542 104003 35:     ERROR    3        ;TRAP NOT SPRUNG
12945 051544 000414          BR       TST706    ;GO TO SCOPE EXIT - SCHOOL'S OUT
12946
12947 051546 022706 000000 45:     CMP      #0,SP     ;WAS IT A RED ZONE TRAP ?
12948 051552 001406          BEQ     65         ;BR IF YES
12949
12950 051554 010437 000004          MOV      R4,@#4    ;RESTORE BUS ERROR VECTOR
12951 051560 005004          CLR      R4        ;[R4] = S / B SP
12952 051562 010603          MOV      SP,R3     ;[R3] = WAS SP
12953 051564 010506          MOV      R5,SP     ;RESET THE SP
12954 051566 104003 55:     ERROR    3        ;TRAP SPRUNG BUT NOT RED ZONE
12955
12956 051570 010506 65:     MOV      R5,SP     ;FIX UP THE SP
12957 051572 010437 000004  MOV      R4,@#4    ;RESTORE BUS ERROR VECTOR
12958
12959 ;*****
12960 ;*TEST 706 TEST PUSH INTO SLR WITH [SP] = 177776
12961 ;*****
12962 051576          TST706:
12963 051576 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12964 051600 012700 000706  MOV      #706,R0    ;LOAD R0 WITH TEST NUMBER
12965 051604 013701 051632  MOV      @#25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
12966 051610 010605          MOV      SP,R5      ;SAVE THE SP
12967 051612 013704 000004  MOV      @#4,R4      ;SAVE THE BUS ERROR VECTOR
12968 051616 012737 051654 000004  MOV      #45,@#4 ;"RED ZONE" TRAP GOES TO 45
12969 051624 012706 177776  MOV      #177776,SP ;MAKE SP=177776
12970 051630 000257          CCC              ;SCOPE SYNC
12971
12972 051632 012746 000200 25:     MOV      #200,-(SP) ;ATTEMPT PUSH INTO SLR - SHOULD CAUSE
12973                                     ;"RED ZONE" TRAP TO BE SPRUNG
12974
12975 051636 010437 000004          MOV      R4,@#4    ;RESTORE BUS ERROR VECTOR
12976 051642 005004          CLR      R4        ;[R4] = S / B SP
12977 051644 010603          MOV      SP,R3     ;[R3] = WAS SP
12978 051646 010506          MOV      R5,SP     ;RESET THE SP
12979 051650 104003 35:     ERROR    3        ;TRAP NOT SPRUNG
12980 051652 000414          BR       TST707    ;GO TO SCOPE EXIT - SCHOOL'S OUT
12981
12982 051654 022706 000000 45:     CMP      #0,SP     ;WAS IT A RED ZONE TRAP ?
12983 051660 001406          BEQ     65         ;BR IF YES
    
```

N04

MAINDEC-11-DGKDA-A KD11-K BASIC LOGIC TESTS
 DGKDA.P11 08-FEB-77 16:17 T706

MACY11 27(1006) 08-FEB-77 16:23 PAGE 236
 TEST PUSH INTO SLR WITH [SP] = 177776

```

12984
12985 051662 010437 000004      MOV      R4,0#4      ;RESTORE BUS ERROR VECTOR
12986 051666 005004      CLR      R4          ;[R4] = S / B SP
12987 051670 010603      MOV      SP,R3      ;[R3] = WAS SP
12988 051672 010506      MOV      R5,SP      ;RESET THE SP
12989 051674 104003      5$:      ERROR      3      ;TRAP SPRUNG BUT NOT RED ZONE
12990
12991 051676 010506      6$:      MOV      R5,SP      ;FIX UP THE SP
12992 051700 010437 000004      MOV      R4,0#4      ;RESTORE BUS ERROR VECTOR
12993
12994      ;:*****
12995      ;:TEST 707      RSVD INSTRUCTION TEST - 000007 THRU 000077
12996      ;:*****
12997 051704      TST707:
12998 051704 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
12999 051706 012700 000707      MOV      #707,R0    ;LOAD R0 WITH TEST NUMBER
13000 051712 010605      5$:      MOV      SP,R5      ;SAVE THE SP
13001 051714 012737 051752 000010      MOV      #4$,0#10  ;SET UP RSVD INSTR. TRAP VECTOR
13002 051722 005037 000012      CLR      0#12
13003 051726 012701 000007      MOV      #7,R1      ;SET UP FIRST ONE IN GROUP
13004 051732 010737 001010      MOV      PC,0#SLPERR ;ONLY LOOP ON BAD OP CODE
13005 051736 010506      1$:      MOV      R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
13006 051740 010137 051746      MOV      R1,0#2$ ;LOAD NEW INSTR
13007 051744 000257      CCC      ;SCOPE SYNC
13008
13009 051746 000007      2$:      000007      ;TEST THE RSVD INSTR - THIS LOCATION
13010      ;GETS CHANGED EACH PASS THROUGH
13011
13012 051750 104005      3$:      ERROR      5      ;RSVD INSTR. IN R1 FAILED TO TRAP
13013
13014 051752 005201      4$:      INC      R1          ;GENERATE NEW RSVD INSTR
13015 051754 022701 000100      CMP      #100,R1    ;AT END OF THIS GROUP ??
13016 051760 001366      BNE     1$          ;BR IF NOT
13017
13018 051762 010506      MOV      R5,SP      ;MAKE SURE TO RESET THE SP
13019 051764 012737 051712 001010      MOV      #5$,0#SLPERR ;LOOP FROM BEGINNING ON ERROR
13020      ;:*****
13021      ;:TEST 710      RSVD INSTRUCTION TEST - 000210 THRU 000237
13022      ;:*****
13023 051772      TST710:
13024 051772 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
13025 051774 012700 000710      MOV      #710,R0    ;LOAD R0 WITH TEST NUMBER
13026 052000 010605      5$:      MOV      SP,R5      ;SAVE THE SP
13027 052002 012737 052040 000010      MOV      #4$,0#10  ;SET UP RSVD INSTR. TRAP VECTOR
13028 052010 005037 000012      CLR      0#12
13029 052014 012701 000210      MOV      #210,R1    ;SET UP FIRST ONE IN GROUP
13030 052020 010737 001010      MOV      PC,0#SLPERR ;SET ERROR LOOP ADDRESS
13031 052024 010506      1$:      MOV      R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
13032 052026 010137 052034      MOV      R1,0#2$ ;LOAD NEW INSTR
13033 052032 000257      CCC      ;SCOPE SYNC
13034
13035 052034 000210      2$:      000210      ;TEST THE RSVD INSTR - THIS LOCATION
13036      ;GETS CHANGED EACH PASS THROUGH
13037
13038 052036 104005      3$:      ERROR      5      ;RSVD INSTR. IN R1 FAILED TO TRAP
13039

```

```

13040 052040 005201          4$: INC R1 ;GENERATE NEW RSVD INSTR
13041 052042 022701 000240  CMP #240,R1 ;AT END OF THIS GROUP ??
13042 052046 001366          BNE 1$ ;BR IF NOT
13043
13044 052050 010506          MOV R5,SP ;MAKE SURE TO RESET THE SP
13045 052052 012737 052000 001010  MOV #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
13046
13047 ;:*****
13048 ;:#TEST 711 RSVD INSTRUCTION TEST - 007000 THRU 007777
13049 ;:*****
13050 052060          TST711:
13051 052060 000004          SCOPE ;CALL THE SCOPE LOOP UTILITY
13052 052062 012700 000711  MOV #711,R0 ;LOAD R0 WITH TEST NUMBER
13053 052066 010605          5$: MOV SP,R5 ;SAVE THE SP
13054 052070 012737 052126 000010  MOV #4$,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
13055 052076 005037 000012  CLR @#12
13056 052102 012701 007000  MOV #7000,R1 ;SET UP FIRST ONE IN GROUP
13057 052106 010737 001010  MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
13058 052112 010506          1$: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
13059 052114 010137 052122  MOV R1,@#2$ ;LOAD NEW INSTR
13060 052120 000257          CCC ;SCOPE SYNC
13061
13062 052122 007000          2$: 007000 ;TEST THE RSVD INSTR - THIS LOCATION
13063 ;GETS CHANGED EACH PASS THROUGH
13064
13065 052124 104005          3$: ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP
13066
13067 052126 005201          4$: INC R1 ;GENERATE NEW RSVD INSTR
13068 052130 022701 010000  CMP #10000,R1 ;AT END OF THIS GROUP ??
13069 052134 001366          BNE 1$ ;BR IF NOT
13070
13071 052136 010506          MOV R5,SP ;MAKE SURE TO RESET THE SP
13072 052140 012737 052066 001010  MOV #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
13073
13074 ;:*****
13075 ;:#TEST 712 RSVD INSTRUCTION TEST - 075000 THRU 076777
13076 ;:*****
13077 052146          TST712:
13078 052146 000004          SCOPE ;CALL THE SCOPE LOOP UTILITY
13079 052150 012700 000712  MOV #712,R0 ;LOAD R0 WITH TEST NUMBER
13080 052154 010605          5$: MOV SP,R5 ;SAVE THE SP
13081 052156 012737 052216 000010  MOV #4$,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
13082 052164 005037 000012  CLR @#12
13083 052170 012701 075000  MOV #75000,R1 ;SET UP FIRST ONE IN GROUP
13084 052174 010737 001010  MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
13085 052200 010506          1$: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
13086 052202 010137 052210  MOV R1,@#2$ ;LOAD NEW INSTR
13087 052206 000257          CCC ;SCOPE SYNC
13088
13089 052210 075000          2$: 75000 ;TEST THE RSVD INSTR - THIS LOCATION
13090 ;GETS CHANGED EACH PASS THROUGH
13091
13092 052212 000240          3$: NOP ;IN CASE NON TRAPPING INSTR IS TWO WORDS
13093 052214 104005          ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP
13094
13095 052216 005201          4$: INC R1 ;GENERATE NEW RSVD INSTR

```

```

13096 052220 022701 076600      CMP      #MED,R1      ;MED INSTRUCTION?
13097 052224 001774              BEQ      4$           ;BR IF YES--SKIP IT.
13098 052226 022701 077000      CMP      #077000,R1  ;AT END OF THIS GROUP ??
13099 052232 001362              BNE      1$           ;BR IF NOT
13100
13101 052234 010506              MOV      R5,SP       ;MAKE SURE TO RESET THE SP
13102 052236 012737 052154 001010  MOV      #5$,2#SLPERR ;LOOP FROM BEGINNING ON ERROR
13103
13104
13105
13106
13107 052244
13108 052244 000004
13109 052246 012700 000713      SCOPE
13110 052252 010605              MOV      #713,RO    ;CALL THE SCOPE LOOP UTILITY
13111 052254 012737 052312 000010 5$:      MOV      SP,R5      ;LOAD RO WITH TEST NUMBER
13112 052262 005037 000012      MOV      #4$,2#10  ;SAVE THE SP
13113 052266 012701 106400      MOV      #4$,2#10  ;SET UP RSVD INSTR. TRAP VECTOR
13114 052272 010737 001010      CLR      2#12
13115 052276 010506              MOV      #106400,R1 ;SET UP FIRST ONE IN GROUP
13116 052300 010137 052306      MOV      PC,2#SLPERR ;SET ERROR LOOP ADDRESS
13117 052304 000257              MOV      R5,SP       ;RESET SP FOR ERROR LOOP AND NEW INSTR
13118
13119 052306 106400              MOV      R1,2#2$ ;LOAD NEW INSTR
13120
13121
13122 052310 104005              CCC          ;SCOPE SYNC
13123
13124 052312 005201              2$:      106400      ;TEST THE RSVD INSTR - THIS LOCATION
13125 052314 022701 106500      ;GETS CHANGED EACH PASS THROUGH
13126 052320 001002
13127 052322 012701 106700      3$:      ERROR 5      ;RSVD INSTR. IN R1 FAILED TO TRAP
13128 052326 022701 110000      4$:      INC      R1      ;GENERATE NEW RSVD INSTR
13129 052332 001361              CMP      #106500,R1 ;MFPD INSTRUCTION ??
13130
13131 052334 010506              BNE      10$        ;BR IF NOT
13132 052336 012737 052252 001010  MOV      #106700,R1 ;SKIP MFPD AND MTPD INSTRUCTIONS
13133 052344 012737 061046 000010 10$:     CMP      #110000,R1 ;AT END OF THIS GROUP ??
13134 052352 012737 000340 000012  BNE      1$         ;BR IF NOT
13135 052360 000004              MOV      R5,SP       ;MAKE SURE TO RESET THE SP
13136
13137
13138
13139
13140
13141 052362 012737 061014 000014  MOV      #5$,2#SLPERR ;LOOP FROM BEGINNING ON ERROR
13142 052370 012737 000340 000016  MOV      #RSERR,2#10 ;RESTORE RSVD INSTR VECTOR
13143
13144
13145
13146
13147
13148
13149
13150 052376
13151 052376 012700 000714      MOV      #340,2#12  ;CALL THE SCOPE LOOP UTILITY

```

```

; THIS NEXT GROUP OF SEQUENTIAL TESTS VERIFIES THAT A "T" BIT
; TRAP CAN BE SERVICED IN EACH MICROWORD THAT DOES A "BUT SERVICE"
; EACH ROUTINE ENTERS THE TRAP MICROUTINE WHEN THE TRAP IS SPRUNG

```

```

13141 052362 012737 061014 000014 TSET:  MOV      #TBSER,2#14 ;SET UP THE "T" BIT TRAP VECTOR
13142 052370 012737 000340 000016  MOV      #340,2#16 ;PRIORITY ?

```

```

; *****
; TEST 714 BUT SERVICE -- ONE WORD INSTRUCTIONS--ALL MODES -- FROM TABLE
; "INSTAB" (INSTRUCTION TABLE) CONTAINS ALL ONE WORD INSTRUCTIONS
; THAT TEST A "BUT SERVICE" IN A UNIQUE ROM LOCATION. THE TABLE MUST
; BE TERMINATED WITH A 0 ENTRY.
; *****

```

```

13149
13150 052376
13151 052376 012700 000714      TST714: MOV      #714,RO    ;;LOAD RO WITH TEST NUMBER

```

D05

MAINDEC-11-DOKDA-A KDI1-K BASIC LOGIC TESTS
 DOKDAA.P11 08-FEB-77 16:17 T714

MACY11 27(1006) 08-FEB-77 16:23 PAGE 239
 BUT SERVICE -- ONE WORD INSTRUCTIONS-ALL MODES -- FROM TABLE

13152	052402	010605			6S:	MOV	SP,R5		;SAVE THE SP
13153	052404	012704	063562			MOV	#INSTAB,R4		;PUT POINTER TO TABLE IN R4
13154	052410	012401			4S:	MOV	(R4)+,R1		;LOAD R1 WITH TEST INSTRUCTION WORD
13155	052412	001422				BEQ	5S		;EXIT TEST IF END OF TABLE
13156	052414	010737	001010			MOV	PC,#SLPERR		;LOOP ON FAILING INSTRUCTION ONLY
13157	052420	010137	052452		1S:	MOV	R1,#2S		;STORE TEST INSTRUCTION TO BE EXECUTED
13158	052424	012702	063236			MOV	#MBUF0,R2		;IN CASE DMI DEST--(R2)
13159	052430	012703	063242			MOV	#MBUF1,R3		;IN CASE SMI--(R3)
13160	052434	010506				MOV	R5,SP		;RESTORE SP FOR ERROR LOOPING
13161	052436	012746	000020			MOV	#20,-(SP)		;SET "T" BIT IN THE NEW PSW
13162	052442	012746	052452			MOV	#2S,-(SP)		;MAKE NEW PC = 2S
13163	052446	000257				CCC			;SCOPE SYNC
13164	052450	000006				RTT			;SET "T" BIT - GO TO 2S
13165									
13166	052452	000240			2S:	NOP			;INSTRUCTION FROM TABLE IS STORED HERE AND ;SHOULD SPRING TRAP
13167									
13168									
13169	052454	104005			3S:	ERROR	5		;BUT SERVICE FAILED
13170									
13171	052456	000754				BR	4S		;GET NEXT INSTRUCTION FOR BUT SERVICE TEST
13172	052460	012737	052402	001010	5S:	MOV	#6S,#SLPERR		;LOOP FROM BEGINNING ON ERROR
13173									
13174									
13175									
13176									
13177	052466								
13178	052466	000004				SCOPE			;CALL THE SCOPE LOOP UTILITY
13179	052470	012700	000715			MOV	#715,R0		;LOAD R0 WITH TEST NUMBER
13180	052474	013701	052512			MOV	#2S,R1		;LOAD R1 WITH TEST INSTRUCTION WORD
13181	052500	012746	000020			MOV	#20,-(SP)		;SET "T" BIT IN THE NEW PSW
13182	052504	012746	052514			MOV	#3S,-(SP)		;MAKE NEW PC = 3S
13183	052510	000257				CCC			;SCOPE SYNC
13184									
13185	052512	000002			2S:	RTI			;INSTRUCTION SHOULD SPRING TRAP
13186									
13187	052514	104005			3S:	ERROR	5		;BUT SERVICE IN XXX FAILED
13188									
13189									
13190									
13191									
13192	052516								
13193	052516	000004				SCOPE			;CALL THE SCOPE LOOP UTILITY
13194	052520	012700	000716			MOV	#716,R0		;LOAD R0 WITH TEST NUMBER
13195	052524	013701	052574			MOV	#2S,R1		;LOAD R1 WITH TEST INSTRUCTION WORD
13196									
13197	052530	032737	040000	063160		.SBTTL	USER CONTROLLED BREAKPOINT -- BIT14		
13198	052536	001401				BIT	#BIT14,#BPTLOC		;BREAKPOINT HALT SET ??
13199	052540	000000				BEQ	.+4		;BR IF NOT
13200	052542	010605				HALT			;BREAK-DEPRESS CONTINUE TO CONTINUE
13201	052544	010737	001010			MOV	SP,R5		;SAVE THE SP
13202	052550	010506				MOV	PC,#SLPERR		;FOR PROPER SP RESETTING ON ERROR LOOP
13203	052552	012737	052600	063242	1S:	MOV	R5,SP		;RESTORE SP FOR ERROR LOOPING
13204	052560	012746	000020			MOV	#3S,#MBUF1		;SET UP POINTER--DEST ADDR = 3S FOR JSR
13205	052564	012746	052574			MOV	#20,-(SP)		;SET "T" BIT IN THE NEW PSW
13206	052570	000257				MOV	#2S,-(SP)		;MAKE NEW PC = 2S
13207	052572	000006				CCC			;SCOPE SYNC
						RTT			;SET "T" BIT - GO TO 2S

E05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 240
 DQKDA.P11 08-FEB-77 16:17 USER CONTROLLED BREAKPOINT -- BIT14

```

13208
13209 052574 004777 010442      2$:   JSR   PC, JMBUF1      ; INSTRUCTION SHOULD SPRING TRAP
13210
13211 052600 104005              3$:   ERROR 5              ; BUT SERVICE IN XXX FAILED
13212
13213 052602 010506              MOV   R5, SP              ; RESTORE SP IF ALL OK OR NOT LOOPING
13214      ; *****
13215      ; *TEST 717 BUT SERVICE TEST - (JMP A)
13216      ; *****
13217      †ST717:
13218 052604 000004              SCOPE                      ; CALL THE SCOPE LOOP UTILITY
13219 052606 012700 000717      MOV   #717, R0             ; ; LOAD R0 WITH TEST NUMBER
13220 052612 013701 052632      MOV   J#2$, R1            ; ; LOAD R1 WITH TEST INSTRUCTION WORD
13221 052616 012746 000020      MOV   #20, -(SP)          ; ; SET "T" BIT IN THE NEW PSW
13222 052622 012746 052632      MOV   #2$, -(SP)          ; ; MAKE NEW PC = 2$
13223 052626 000257              CCC                        ; ; SCOPE SYNC
13224 052630 000006              RTT                        ; ; SET "T" BIT - GO TO 2$
13225
13226 052632 000167 000000      2$:   JMP   3$              ; JMP INSTRUCTION SHOULD SPRING TRAP
13227
13228 052636 104005              3$:   ERROR 5              ; BUT SERVICE IN XXX FAILED
13229
13230      ; *****
13231      ; *TEST 720 BUT SERVICE TEST - (JMP JA)
13232      ; *****
13233      †ST720:
13234 052640 000004              SCOPE                      ; CALL THE SCOPE LOOP UTILITY
13235 052642 012700 000720      MOV   #720, R0             ; ; LOAD R0 WITH TEST NUMBER
13236 052646 013701 052674      MOV   J#2$, R1            ; ; LOAD R1 WITH TEST INSTRUCTION WORD
13237 052652 012737 052700 063236  MOV   #3$, J#MBUFO        ; ; SET UP POINTER--DEST ADDR = 3$ FOR JMP
13238 052660 012746 000020      MOV   #20, -(SP)          ; ; SET "T" BIT IN THE NEW PSW
13239 052664 012746 052674      MOV   #2$, -(SP)          ; ; MAKE NEW PC = 2$
13240 052670 000257              CCC                        ; ; SCOPE SYNC
13241 052672 000006              RTT                        ; ; SET "T" BIT - GO TO 2$
13242
13243 052674 000177 010336      2$:   JMP   J#BUFO          ; JMP INSTRUCTION SHOULD SPRING TRAP
13244
13245 052700 104005              3$:   ERROR 5              ; BUT SERVICE IN XXX FAILED
13246
13247      ; *****
13248      ; *TEST 721 BUT SERVICE TEST - (RTS PC)
13249      ; *****
13250      †ST721:
13251 052702 000004              SCOPE                      ; CALL THE SCOPE LOOP UTILITY
13252 052704 012700 000721      MOV   #721, R0             ; ; LOAD R0 WITH TEST NUMBER
13253 052710 013701 052744      MOV   J#2$, R1            ; ; LOAD R1 WITH TEST INSTRUCTION WORD
13254 052714 010605              MOV   SP, R5              ; ; SAVE THE SP
13255 052716 010737 001010      MOV   PC, J#SLPERR        ; ; FOR PROPER SP RESETTING ON ERROR LOOP
13256 052722 010506              1$:   MOV   R5, SP          ; ; RESTORE SP FOR ERROR LOOPING
13257 052724 012746 052746      MOV   #3$, -(SP)          ; ; RTS WILL LOAD PC WITH 3$
13258 052730 012746 000020      MOV   #20, -(SP)          ; ; SET "T" BIT IN THE NEW PSW
13259 052734 012746 052744      MOV   #2$, -(SP)          ; ; MAKE NEW PC = 2$
13260 052740 000257              CCC                        ; ; SCOPE SYNC
13261 052742 000006              RTT                        ; ; SET "T" BIT - GO TO 2$
13262
13263 052744 000207              2$:   RTS   PC              ; RTS INSTRUCTION SHOULD SPRING TRAP

```

13264
13265 052746 104005
13266
13267
13268
13269
13270
13271
13272
13273
13274
13275
13276
13277
13278
13279
13280
13281
13282
13283
13284
13285
13286
13287
13288
13289
13290
13291
13292
13293
13294
13295
13296
13297
13298
13299 052750
13300 052750 000004
13301 052752 012700 000722
13302 052756 012705 063264
13303 052762 010737 001010
13304 052766 024545
13305
13306 052770 005725
13307 052772 022705 063342
13308 052776 001413
13309 053000 012501
13310 053002 012503
13311 053004 000257
13312
13313 053006 060103
13314
13315 053010 021503
13316 053012 001766
13317
13318 053014 011504
13319 053016 014502

3\$: ERROR 5 ;BUT SERVICE IN XXX FAILED

;TEST 722 ALU ADD FUNCTION TEST
;THIS TEST VERIFIES THAT THE ALU ADD FUNCTION CAN RESPOND CORRECTLY
;TO THE 8 POSSIBLE COMBINATIONS THAT COULD OCCUR AT THE INPUTS OF
;EACH OF THE 16 BIT POSITIONS AS DESCRIBED BELOW:

```

;      AIN      BIN      CIN
;
;      0        0        0
;      0        0        1
;      0        1        0
;      0        1        1
;      1        0        0
;      1        0        1
;      1        1        0
;      1        1        1

```

;THE TEST NO.S ALONG WITH THE CORRECT ANSWERS ARE STORED IN A TABLE
;TAGGED "ALUADD" AS SHOWN BELOW:

```

;ALUADD:      NULL
;              SRC OP1
;              DST OP1
;              SUM1
;              SRC OP2
;              DST OP2
;              SUM2
;              ETC.

```

;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING PAIR
;OF NO.S IF SW09=1 OR GO ON TO THE NEXT PAIR IF SW09=0.

;T722:

```

SCOPE      ;CALL THE SCOPE LOOP UTILITY
MOV        #722,R0      ;LOAD R0 WITH TEST NUMBER
1$: MOV     #ALUADD+4,R5 ;R5 POINTS TO TABLE OF NO.S
MOV        PC,2*SLPERR  ;LOOP ONLY ON FAILING PAIR OF #'S
CMP        -(R5),-(R5)  ;RESET R5 TO POINT TO BAD GUYS
                ;(OR NULL ENTRY FIRST TIME THROUGH)
4$: TST     (R5)+        ;POINT TO A SRC OP
CMP        #ALUADD+62,R5 ;DONE ALL NO.S IN TABLE ?
BEQ        5$           ;BR IF YES
MOV        (R5)+,R1     ;LOAD SRC OP
MOV        (R5)+,R3     ;LOAD DEST OP
CCC        ;SCOPE SYNC

2$: ADD     R1,R3        ;TEST THE ADD FUNCTION

CMP        (R5),R3      ;CORRECT SUM ?
BEQ        4$           ;GO ADD NEXT PAIR IF YES

MOV        (R5),R4      ;GET S / B SUM
MOV        -(R5),R2     ;GET DEST OP

```

```

13320 053020 104010 3$: ERROR 10 ;ALU ADD OPERATION FAILED
13321
13322 053022 005725 TST (R5)+ ;CORRECT R5 POINTER
13323 053024 000761 BR 4$ ;GO DO NEXT PAIR
13324
13325 053026 012737 052756 001010 5$: MOV #1$,2$SLPERR ;LOOP FROM BEGINNING ON ERROR
13326

```

```

;*****
;TEST 723 ALU SUB FUNCTION TEST
;THIS TEST VERIFIES THAT THE ALU ADD FUNCTION CAN RESPOND CORRECTLY
;TO THE 8 POSSIBLE COMBINATIONS THAT COULD OCCUR AT THE INPUTS OF
;EACH OF THE 16 BIT POSITIONS AS DESCRIBED BELOW:

```

	AIN	BIN	CIN
	0	0	0
	0	0	1
	0	1	0
	0	1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1

```

;THE TEST NO.S ALONG WITH THE CORRECT ANSWERS ARE STORED IN A TABLE
;TAGGED "ALUADD" AS SHOWN BELOW:

```

```

;ALUSUB: NULL
; SRC OP1
; DST OP1
; DIFF1
; SRC OP2
; DST OP2
; DIFF2
; ETC.

```

```

;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING PAIR
;OF NO.S IF SW09=1 OR GO ON TO THE NEXT PAIR IF SW09=0.

```

```

;*****

```

```

13359 053034
13360 053034 000004
13361 053036 012700 000723
13362 053042 012705 063504
13363 053046 010737 001010
13364 053052 024545
13365
13366 053054 005725
13367 053056 022705 063562
13368 053062 001413
13369 053064 012501
13370 053066 012503
13371 053070 000257
13372
13373 053072 160103 2$: SUB R1,R3 ;TEST THE SUB FUNCTION
13374
13375 053074 021503 CMP (R5),R3 ;CORRECT DIFF. ?

```

```

;CALL THE SCOPE LOOP UTILITY
;LOAD R0 WITH TEST NUMBER
;R5 POINTS TO TABLE OF NO.S
;LOOP ONLY ON FAILING PAIR OF #'S
;RESET R5 TO POINT TO BAD GUYS
;(OR NULL ENTRY FIRST TIME THROUGH)
;POINT TO A SRC OP
;DONE ALL NO.S IN TABLE ?
;BR IF YES
;LOAD SRC OP
;LOAD DEST OP
;SCOPE SYNC

```

```

13376 053076 001766          BEQ      4$          ;GO SUB NEXT PAIR IF YES
13377
13378 053100 011504          MOV      (R5),R4    ;GET S / B DIFF
13379 053102 014502          MOV      -(R5),R2   ;GET DEST OP
13380 053104 104010          3$:     ERROR      10 ;ALU SUB OPERATION FAILED
13381
13382 053106 005725          TST      (R5)+      ;CORRECT R5 POINTER
13383 053110 000761          BR       4$          ;GO DO NEXT PAIR
13384
13385 053112 012737 053042 001010 5$:     MOV      #1$,2$SLPERR ;LOOP FROM BEGINNING ON ERROR
13386

```

```

;*****
;TEST 724 ALU "AND" FUNCTION TEST USING BIC INSTRUCTION
;THIS TEST VERIFIES THAT THE ALU "AND" FUNCTION RESPONDS CORRECTLY
;TO ALL POSSIBLE COMBINATIONS FOR EACH OF THE 16 BIT POSITIONS
;IT EXECUTES THE BIC INSTRUCTION FOR THE FOLLOWING PAIRS OF
;OPERANDS AND TESTS FOR THE INDICATED RESULT:

```

;	SOURCE OP	DEST. OP	RESULT
;	000000	000000	000000
;	177777	177777	000000
;	000000	177777	177777
;	177777	000000	000000
;	125252	125252	000000
;	052525	052525	000000
;	125252	052525	052525
;	052525	125252	125252

```

;THE 8 PAIRS OF NO.S AND THE ANSWERS ARE STORED IN A TABLE TAGGED
;"ANDTAB" IN THE FOLLOWING PATTERN:

```

```

;ANDTAB:      NULL
;              SRC OP1
;              DST OP1
;              ANS1
;              SRC OP2
;              DST OP2
;              ANS2
;              ETC.

```

```

;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING
;PAIR OF NO.S IF SW09=1 OR GO ON TO TEST THE NEXT PAIR IF SW09=0
;*****

```

```

13419 053120
13420 053120 000004          SCOPE
13421 053122 012700 000724    MOV      #724,R0    ;CALL THE SCOPE LOOP UTILITY
13422 053126 012705 063344    1$:     MOV      #ANDTAB+4,R5 ;LOAD R0 WITH TEST NUMBER
13423 053132 010737 001010    MOV      PC,#SLPERR ;RS POINTS TO TABLE OF TEST NO.S
13424 053136 024545          CMP      -(R5),-(R5) ;LOOP ONLY ON FAILING PAIR OF #'S
13425
13426 053140 005725          4$:     TST      (R5)+      ;RESET R5 TO POINT TO BAD GUYS
13427 053142 022705 063422    CMP      #ANDTAB+62,R5 ; (OR NULL ENTRY FIRST TIME THROUGH)
13428 053146 001413          BEQ      5$          ;POINT TO A SOURCE OPR
13429 053150 012501          MOV      (R5)+,R1   ;DONE ALL COMBINATIONS ?
13430 053152 012503          MOV      (R5)+,R3   ;BR IF YES
13431 053154 000257          CCC          ;LOAD THE SRC OP
;              ;LOAD THE DEST OP
;              ;SCOPE SYNC

```


J05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 245
 ALU "OR" FUNCTION TEST USING BIS INSTRUCTION

```

13488 053232 001413      BEQ     5$           ;BR IF YES
13489 053234 012501      MOV     (R5)+,R1    ;LOAD THE SRC OP
13490 053236 012503      MOV     (R5)+,R3    ;LOAD THE DEST OP
13491 053240 000257      CCC           ;SCOPE SYNC
13492
13493 053242 050103      2$:     BIS     R1,R3      ;TEST THE "OR"
13494
13495 053244 020315      CMP     R3,(R5)     ;RESULT CORRECT ?
13496 053246 001766      BEQ     4$           ;BR IF YES - GET THE NEXT PAIR
13497
13498 053250 011504      MOV     (R5),R4     ;GET THE S / B DATA
13499 053252 014502      MOV     -(R5),R2    ;GET DEST OP
13500 053254 104010      3$:     ERROR   10       ;ALU "OR" FAILED
13501
13502 053256 005725      TST     (R5)+       ;CORRECT R5 POINTER
13503 053260 000761      BR      4$           ;GO GET NEXT PAIR
13504
13505 053262 012737 053212 001010 5$:     MOV     #1$,#$SLPERR ;LOOP FROM BEGINNING ON ERROR
13506
13507      ;*****
13508      ;#TEST 726      INC / DEC / ADD TEST - CYCLE NO.S 000000-077777
13509      ;THIS TEST COMBINES THE INC / DEC / ADD INSTRUCTIONS IN THE FOLLOWING
13510      ;TEST SEQUENCE:
13511
13512      ;1. BOTH SOURCE AND DEST OPS ARE ZEROED
13513      ;2. THE TWO NO.S ARE ADDED AND THE RESULT COMPARED WITH 000000
13514      ;3. THE SOURCE OP IS INCREMENTED
13515      ;4. THE DEST OP IS DECREMENTED
13516      ;5. STEPS 2,3, AND 4 ARE REPEATED UNTIL THE SOURCE OP GOES
13517      ;    NEGATIVE
13518
13519      ;ON DETECTION OF A NON-ZERO RESULT THE ERROR IS REPORTED AND THEN IF:
13520
13521      ;    1. SW09=0 THE TEST IS EXITED
13522      ;    2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS
13523
13524      ;*****
13525 053270      TST726:
13526 053270 000004      SCOPE           ;CALL THE SCOPE LOOP UTILITY
13527 053272 012700 000726 10$:     MOV     #726,R0    ;LOAD R0 WITH TEST NUMBER
13528 053276 005001      CLR     R1         ;INITIALIZE REGS TO 000000
13529 053300 005002      CLR     R2
13530 053302 005004      CLR     R4
13531 053304 010737 001010 1$:     MOV     PC,#$SLPERR ;LOOP ONLY ON FAILING PAIR OF #'S
13532 053310 010203      MOV     R2,R3     ;LOAD DEST OPERAND
13533 053312 000257      CCC           ;SCOPE SYNC
13534
13535 053314 060103      2$:     ADD     R1,R3     ;ADD THE TWO TEST NO.S
13536      ;RESULT S / B = 000000
13537
13538 053316 020403      CMP     R4,R3     ;RESULT = 000000 ?
13539 053320 001402      BEQ     4$         ;BR IF YES
13540
13541 053322 104010      3$:     ERROR   10       ;INCORRECT RESULT IN R3
13542
13543 053324 000407      BR      TST727    ;;EXIT TO NEXT TEST
  
```

K05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 246
INC / DEC / ADD TEST - CYCLE NO.S 000000-077777

```

13544
13545 053326 005201          4S:  INC      R1          ;ADD 1 TO SOURCE OP
13546 053330 100402          BMI      5S          ;GET OUT IF IT WENT NEGATIVE
13547 053332 005302          DEC      R2          ;SUB 1 FROM THE DEST OP
13548 053334 000765          BR       1S          ;GO ADD THE TWO NO.S
13549
13550 053336 012737 053276 001010 5S:  MOV      #10S,2#SLPERR ;LOOP FROM BEGINNING ON ERROR
13551
13552  ;*****
13553  ;#TEST 727      INC / DEC / ADD TEST - CYCLE NO.S 077777-000000
13554  ;THIS TEST COMBINES THE INC / DEC / ADD INSTRUCTIONS IN THE FOLLOWING
13555  ;TEST SEQUENCE:
13556
13557  ;1. BOTH SOURCE AND DEST OPS ARE ZEROED
13558  ;2. THE TWO NO.S ARE ADDED AND THE RESULT COMPARED WITH 000000
13559  ;3. THE SOURCE OP IS DECREMENTED
13560  ;4. THE DEST OP IS INCREMENTED
13561  ;5. STEPS 2, 3, AND 4 ARE REPEATED UNTIL THE DEST. OP GOES
13562  ;    NEGATIVE
13563
13564  ;ON DETECTION OF A NON-ZERO RESULT THE ERROR IS REPORTED AND THEN IF:
13565
13566  ;    1. SW09=0 THE TEST IS EXITED
13567  ;    2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS
13568  ;*****
13569  ;#TEST 727:
13570 053344 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13571 053346 012700 000727  MOV      #727,R0  ;LOAD R0 WITH TEST NUMBER
13572 053352 005001          CLR      R1          ;INITIALIZE REGS TO 000000
13573 053354 005002          CLR      R2
13574 053356 005004          CLR      R4
13575 053360 010737 001010  MOV      PC,2#SLPERR ;LOOP ONLY ON FAILING PAIR OF #'S
13576 053364 010203          MOV      R2,R3      ;LOAD DEST OPERAND
13577 053366 000257          CCC          ;SCOPE SYNC
13578
13579 053370 060103          2S:  ADD      R1,R3      ;ADD THE TWO TEST NO.S
13580  ;RESULT S / B = 000000
13581
13582 053372 020403          CMP      R4,R3      ;RESULT = 000000 ?
13583 053374 001402          BEQ      4S          ;BR IF YES
13584
13585 053376 104010          3S:  ERROR   10          ;INCORRECT RESULT IN R3
13586
13587 053400 000407          BR       TST730      ;;GO TO SCOPE EXIT
13588
13589 053402 005202          4S:  INC      R2          ;ADD 1 TO DEST. OP
13590 053404 100402          BMI      5S          ;GET OUT IF IT WENT NEGATIVE
13591 053406 005301          DEC      R1          ;SUB 1 FROM THE SOURCE OP
13592 053410 000765          BR       1S          ;GO ADD THE TWO NO.S
13593
13594 053412 012737 053352 001010 5S:  MOV      #10S,2#SLPERR ;LOOP FROM BEGINNING ON ERROR
13595
13596  ;*****
13597  ;#TEST 730      MUL RA,RB TEST ; N:C = 1111
13598  ;*****
13599 053420  ;#TEST 730:

```

```

13600 053420 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13601 053422 012700 000730  MOV      #730,R0      ;:LOAD R0 WITH TEST NUMBER
13602 053426 013737 053456 001076  MOV      @#25,@#STMPO ;GET TEST INSTRUCTION WORD
13603 053434 005001          CLR      R1          ;S/B RESULT IN R2
13604 053436 012704 000006  MOV      #6,R4       ;S/B RESULT IN R3
13605 053442 012702 000002  MOV      #2,R2       ;INITIALIZE REG
13606 053446 005003          CLR      R3          ;INITIALIZE REG + 1
13607 053450 012705 000003  MOV      #3,R5       ;INITIALIZE SRC
13608 053454 000277          SCC                    ;SCOPE SYNC
13609
13610 053456 070205          2$:      MUL      R5,R2      ;TEST THE MUL
13611
13612 053460 100403          BMI      3$          ;N:C=0000?
13613 053462 001402          BEQ      3$
13614 053464 102401          BVS      3$
13615 053466 103001          BCC      4$
13616
13617 053470 104044          3$:      ERROR    44          ;COND CODES SET IMPROPERLY
13618
13619 053472 020304          4$:      CMP      R3,R4      ;REG+1 CORRECT?
13620 053474 001002          BNE      5$          ;BR IF NOT
13621 053476 020102          CMP      R1,R2       ;REG CORRECT?
13622 053500 001401          BEQ      T$T731     ;;BR IF YES
13623
13624 053502 104045          5$:      ERROR    45          ;MUL DELIVERED WRONG RESULT
13625
13626
13627
13628
13629
13630 053504          ;:*****
13631 053504 000004          ;:TEST 731      MUL (RA),RB TEST ; N:C = 0000-SET C
13632 053512 013737 053546 001076  ;:*****
13633 053520 005001          T$T731:
13634 053522 012704 123450  MOV      #731,R0      ;CALL THE SCOPE LOOP UTILITY
13635 053526 012702 012345  MOV      @#25,@#STMPO ;:LOAD R0 WITH TEST NUMBER
13636 053532 005003          CLR      R1          ;GET TEST INSTRUCTION WORD
13637 053534 012705 063236  MOV      #123450,R4   ;S/B RESULT IN R2
13638 053540 012715 000010  MOV      #012345,R2   ;S/B RESULT IN R3
13639 053544 000257          CLR      R3          ;INITIALIZE REG
13640          ;INITIALIZE REG + 1
13641 053546 070215          2$:      MUL      (R5),R2    ;TEST THE MUL
13642          ;N:C=0001?
13643 053550 100403          BMI      3$
13644 053552 001402          BEQ      3$
13645 053554 102401          BVS      3$
13646 053556 103401          BCS      4$
13647
13648 053560 104044          3$:      ERROR    44          ;COND CODES SET IMPROPERLY
13649
13650 053562 020304          4$:      CMP      R3,R4      ;REG+1 CORRECT?
13651 053564 001002          BNE      5$          ;BR IF NOT
13652 053566 020102          CMP      R1,R2       ;REG CORRECT?
13653 053570 001401          BEQ      T$T732     ;;BR IF YES
13654
13655 053572 104045          5$:      ERROR    45          ;MUL DELIVERED WRONG RESULT
    
```


M05

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
 DQKDAAR.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 248
 T731 MUL (RA),RB TEST ; N:C = 0000-SET C

```

13656
13657
13658
13659
13660 053574
13661 053574 000004
13662 053576 012700 000732
13663 053602 013737 053634 001076
13664 053610 005001
13665 053612 005004
13666 053614 005002
13667 053616 012703 177777
13668 053622 012705 063236
13669 053626 012715 000010
13670 053632 000257
13671
13672 053634 070225 2$: MUL (R5)+,R2 ;TEST THE MUL
13673
13674 053636 100403 BMI 3$ ;N:C=0100?
13675 053640 001002 BNE 3$
13676 053642 102401 BVS 3$
13677 053644 103001 BCC 4$
13678
13679 053646 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
13680
13681 053650 020304 4$: CMP R3,R4 ;REG+1 CORRECT?
13682 053652 001002 BNE 5$ ;BR IF NOT
13683 053654 020102 CMP R1,R2 ;REG CORRECT?
13684 053656 001401 BEQ 6$ ;BR IF YES
13685
13686 053660 104045 5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
13687
13688 053662 022705 063240 6$: CMP #Mbuf0+2,R5 ;DID R5 GET AUTO-INCREMENTED?
13689 053666 001401 BEQ TST733 ;;BR IF YES
13690
13691 053670 104046 ERROR 46 ;AUTO INCREMENT DID NOT OCCUR
13692
13693
13694
13695
13696 053672
13697 053672 000004
13698 053674 012700 000733
13699 053700 013737 053740 001076
13700 053706 012701 177777
13701 053712 012704 177770
13702 053716 012702 000001
13703 053722 005003
13704 053724 012705 063232
13705 053730 012737 177770 063236
13706 053736 000257
13707
13708 053740 070235 2$: MUL @(R5)+,R2 ;TEST THE MUL
13709
13710 053742 100003 BPL 3$ ;N:C=1000?
13711 053744 001402 BEQ 3$

```

```

*****
;TEST 732 MUL (RA)+,RB TEST ; N:C = 0000-SET Z
*****

```

```

TST732:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #732,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,@#STMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RESULT IN R2
CLR R4 ;S/B RESULT IN R3
CLR R2 ;INITIALIZE REG
MOV #-1,R3 ;INITIALIZE REG + 1
MOV #Mbuf0,R5 ;SET UP POINTER TO SRC
MOV #10,(R5) ;INITIALIZE SRC
CCC ;SCOPE SYNC

2$:
MUL (R5)+,R2 ;TEST THE MUL

BMI 3$ ;N:C=0100?
BNE 3$
BVS 3$
BCC 4$

3$:
ERROR 44 ;COND CODES SET IMPROPERLY

4$:
CMP R3,R4 ;REG+1 CORRECT?
BNE 5$ ;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ 6$ ;BR IF YES

5$:
ERROR 45 ;MUL DELIVERED WRONG RESULT

6$:
CMP #Mbuf0+2,R5 ;DID R5 GET AUTO-INCREMENTED?
BEQ TST733 ;;BR IF YES

ERROR 46 ;AUTO INCREMENT DID NOT OCCUR

```

```

*****
;TEST 733 MUL @(RA)+,RB TEST ; N:C = 0000-SET N ; SRC,DST = -,+
*****

```

```

TST733:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #733,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,@#STMP0 ;GET TEST INSTRUCTION WORD
MOV #-1,R1 ;S/B RESULT IN R2
MOV #-10,R4 ;S/B RESULT IN R3
MOV #1,R2 ;INITIALIZE REG
CLR R3 ;INITIALIZE REG + 1
MOV #ATA+10,R5 ;SET UP POINTER TO POINTER TO Mbuf0
MOV #-10,@#Mbuf0 ;INITIALIZE SRC
CCC ;SCOPE SYNC

2$:
MUL @(R5)+,R2 ;TEST THE MUL

BPL 3$ ;N:C=1000?
BEQ 3$

```

N05

MAINDEC-11-DGKDA-A KD11-K BASIC LOGIC TESTS
 DGKDA.P11 08-FEB-77 16:17 T733

MACY11 27(1006) 08-FEB-77 16:23 PAGE 249
 MUL 2(RA)+,RB TEST ; N:C = 0000-SET N ; SRC,DST = -,+

13712	053746	102401			BVS	3S		
13713	053750	103001			BCC	4S		
13714								
13715	053752	104044			3S:	ERROR	44	;COND CODES SET IMPROPERLY
13716								
13717	053754	020304			4S:	CMP	R3,R4	;REG+1 CORRECT?
13718	053756	001002				BNE	5S	;BR IF NOT
13719	053760	020102				CMP	R1,R2	;REG CORRECT?
13720	053762	001401				BEQ	6S	;BR IF YES
13721								
13722	053764	104045			5S:	ERROR	45	;MUL DELIVERED WRONG RESULT
13723								
13724	053766	022705	063234		6S:	CMP	#ATA+12,R5	;DID R5 GET AUTO-INCREMENTED?
13725	053772	001401				BEQ	TST734	;BR IF YES
13726								
13727	053774	104046				ERROR	46	;AUTO INCREMENT DID NOT OCCUR
13728								
13729								
13730								
13731								
13732	053776							
13733	053776	000004			TST734:	SCOPE		;CALL THE SCOPE LOOP UTILITY
13734	054000	012700	000734			MOV	#734,R0	;LOAD R0 WITH TEST NUMBER
13735	054004	013737	054044	001076		MOV	2#2S,2#STMPD	;GET TEST INSTRUCTION WORD
13736	054012	012701	177777			MOV	#-1,R1	;S/B RESULT IN R2
13737	054016	012704	177770			MOV	#-10,R4	;S/B RESULT IN R3
13738	054022	012702	177777			MOV	#-1,R2	;INITIALIZE REG
13739	054026	005003				CLR	R3	;INITIALIZE REG + 1
13740	054030	012705	063240			MOV	#MBUFD+2,R5	;SET UP POINTER TO SRC
13741	054034	012737	000010	063236		MOV	#10,2#MBUFD	;INITIALIZE SRC
13742	054042	000277				SCC		;SCOPE SYNC
13743								
13744	054044	070245			2S:	MUL	-(R5),R2	;TEST THE MUL
13745								
13746	054046	100003				BPL	3S	;N:C=1000?
13747	054050	001402				BEQ	3S	
13748	054052	102401				BVS	3S	
13749	054054	103001				BCC	4S	
13750								
13751	054056	104044			3S:	ERROR	44	;COND CODES SET IMPROPERLY
13752								
13753	054060	020304			4S:	CMP	R3,R4	;REG+1 CORRECT?
13754	054062	001002				BNE	5S	;BR IF NOT
13755	054064	020102				CMP	R1,R2	;REG CORRECT?
13756	054066	001401				BEQ	6S	;BR IF YES
13757								
13758	054070	104045			5S:	ERROR	45	;MUL DELIVERED WRONG RESULT
13759								
13760	054072	022705	063236		6S:	CMP	#MBUFD,R5	;DID SRC REG GET AUTO-DECREMENTED?
13761	054076	001401				BEQ	TST735	;BR IF YES
13762								
13763	054100	104046				ERROR	46	;AUTO DECREMENT DID NOT OCCUR
13764								
13765								
13766								
13767								

```

*****
;TEST 734 MUL -(RA),RB TEST ; N:C = 1111-CLR ALL BUT N ; SRC,DSK = +,-
*****
TST734:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #734,R0 ;LOAD R0 WITH TEST NUMBER
MOV 2#2S,2#STMPD ;GET TEST INSTRUCTION WORD
MOV #-1,R1 ;S/B RESULT IN R2
MOV #-10,R4 ;S/B RESULT IN R3
MOV #-1,R2 ;INITIALIZE REG
CLR R3 ;INITIALIZE REG + 1
MOV #MBUFD+2,R5 ;SET UP POINTER TO SRC
MOV #10,2#MBUFD ;INITIALIZE SRC
SCC ;SCOPE SYNC

2S: MUL -(R5),R2 ;TEST THE MUL

BPL 3S ;N:C=1000?
BEQ 3S
BVS 3S
BCC 4S

3S: ERROR 44 ;COND CODES SET IMPROPERLY

4S: CMP R3,R4 ;REG+1 CORRECT?
BNE 5S ;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ 6S ;BR IF YES

5S: ERROR 45 ;MUL DELIVERED WRONG RESULT

6S: CMP #MBUFD,R5 ;DID SRC REG GET AUTO-DECREMENTED?
BEQ TST735 ;BR IF YES

ERROR 46 ;AUTO DECREMENT DID NOT OCCUR

*****
;TEST 735 MUL 2-(RA),RB TEST ; N:C = 1111-CLR ALL BUT C ; SRC,DST = -,-
*****

```

```

13768 054102          TST735:
13769 054102 000004          SCOPE
13770 054104 012700 000735  MOV      #735,R0          ;CALL THE SCOPE LOOP UTILITY
13771 054110 013737 054150 001076  MOV      @#25,@#STMP0    ;:LOAD R0 WITH TEST NUMBER
13772 054116 005001          CLR      R1              ;GET TEST INSTRUCTION WORD
13773 054120 012704 106420  MOV      #106420,R4      ;S/B RESULT IN R2
13774 054124 012702 177776  MOV      #-2,R2          ;/S/B RESULT IN R3
13775 054130 012703 177777  MOV      #-1,R3          ;INITIALIZE REG
13776 054134 012705 063234  MOV      #ATA+12,R5      ;INITIALIZE REG + 1
13777 054140 012737 134570 063236  MOV      #-43210,@#MBOFO ;SET UP POINTER TO POINTER TO MBOFO
13778 054146 000277          SCC                   ;INITIALIZE SRC
13779                                ;SCOPE SYNC
13780 054150 070255          2$:  MUL      2-(R5),R2    ;TEST THE MUL
13781
13782 054152 100403          BMI      3$              ;N:C=0001?
13783 054154 001402          BEQ      3$
13784 054156 102401          BVS      3$
13785 054160 103401          BCS      4$
13786
13787 054162 104044          3$:  ERROR    44          ;COND CODES SET IMPROPERLY
13788
13789 054164 020304          4$:  CMP      R3,R4        ;REG+1 CORRECT?
13790 054166 001002          BNE      5$              ;BR IF NOT
13791 054170 020102          CMP      R1,R2          ;REG CORRECT?
13792 054172 001401          BEQ      6$              ;BR IF YES
13793
13794 054174 104045          5$:  ERROR    45          ;MUL DELIVERED WRONG RESULT
13795
13796 054176 022705 063232  6$:  CMP      #ATA+10,R5   ;DID R5 GET AUTO-DECREMENTED?
13797 054202 001401          BEQ      TST736         ;;BR IF YES
13798
13799 054204 104046          ERROR    46          ;AUTO INCREMENT DID NOT OCCUR
13800
13801
13802
13803
13804 05 06
13805 054206 000004          ;:*****
13806 054210 012700 000736  MOV      #736,R0          ;*TEST 736 MUL X(RA),RB TEST ; N:C = 1111 TO 0100
13807 054214 013737 054250 001076  MOV      @#25,@#STMP0    ;:*****
13808 054222 005001          TST736:
13809 054224 005004          SCOPE
13810 054226 012702 012345  MOV      #012345,R2      ;CALL THE SCOPE LOOP UTILITY
13811 054232 012703 177777  MOV      #-1,R3          ;:LOAD R0 WITH TEST NUMBER
13812 054236 012705 063236  MOV      #MBOFO,R5      ;GET TEST INSTRUCTION WORD
13813 054242 005065 000002  CLR      2(R5)          ;S/B RESULT IN R2
13814 054246 000277          SCC                   ;S/B RESULT IN R3
13815                                ;INITIALIZE REG
13816 054250 070265 000002  2$:  MUL      2(R5),R2    ;INITIALIZE REG + 1
13817                                ;SET UP POINTER TO SRC
13818                                ;INITIALIZE SRC
13819                                ;SCOPE SYNC
13818 054254 100403          BMI      3$              ;N:C=0100?
13819 054256 001002          BNE      3$
13820 054260 102401          BVS      3$
13821 054262 103001          BCC      4$
13822
13823 054264 104044          3$:  ERROR    44          ;COND CODES SET IMPROPERLY

```

```

13824
13825 054266 020304      4$:  CMP      R3,R4      ;REG+1 CORRECT?
13826 054270 001002      BNE      5$          ;BR IF NOT
13827 054272 020102      CMP      R1,R2      ;REG CORRECT?
13828 054274 001401      BEQ      T$T737     ;;BR IF YES
13829
13830 054276 104045      5$:  ERROR    45          ;MUL DELIVERED WRONG RESULT
13831
13832      ;*****
13833      ;*TEST 737      MUL 2X(RA),RB TEST
13834      ;*****
13835      T$T737:
13836 054300 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
13837 054302 012700 000737  MOV      #737,R0    ;:LOAD R0 WITH TEST NUMBER
13838 054306 013737 054344 001076  MOV      @#2$,@#STMPD ;GET TEST INSTRUCTION WORD
13839 054314 005001      CLR      R1          ;S/B RESULT IN R2
13840 054316 012704 000100  MOV      #100,R4    ;S/B RESULT IN R3
13841 054322 012702 000010  MOV      #10,R2     ;INITIALIZE REG
13842 054326 005003      CLR      R3          ;INITIALIZE REG + 1
13843 054330 012705 063222  MOV      #ATA,R5    ;GET POINTER TO TABLE OF POINTERS
13844 054334 012737 000010 063236  MOV      #10,@#M$BUFO ;INITIALIZE SRC
13845 054342 000257      CCC          ;SCOPE SYNC
13846
13847 054344 070275 000010      2$:  MUL      @10(R5),R2 ;TEST THE MUL
13848
13849 054350 020304      CMP      R3,R4      ;REG+1 CORRECT?
13850 054352 001002      BNE      3$          ;BR IF NOT
13851 054354 020102      CMP      R1,R2      ;REG CORRECT?
13852 054356 001401      BEQ      T$T740     ;;BR IF YES
13853
13854 054360 104045      3$:  ERROR    45          ;MUL DELIVERED WRONG RESULT
13855
13856      ;*****
13857      ;*TEST 740      DIV #N,RA TEST ; N:C = 1111
13858      ;*****
13859      T$T740:
13860 054362 000004      SCOPE          ;CALL THE SCOPE LOOP UTILITY
13861 054364 012700 000740  MOV      #740,R0    ;:LOAD R0 WITH TEST NUMBER
13862 054370 013737 054416 001076  MOV      @#2$,@#STMPD ;GET COPY OF TEST INSTRUCTION
13863 054376 012701 010000  MOV      #010000,R1 ;S/B RES IN R2
13864 054402 012704 000001  MOV      #1,R4      ;S/B RES IN R3
13865 054406 005002      CLR      R2          ;SET UP REG OPERAND
13866 054410 012703 020001  MOV      #020001,R3 ;SET UP REG+1 OP
13867 054414 000277      SCC          ;SCOPE SYNC
13868
13869 054416 071227 000002      2$:  DIV      #2,R2      ;TEST DIV
13870
13871 054422 100403      BMI      3$          ;N:C=0000?
13872 054424 001402      BEQ      3$
13873 054426 102401      BVS      3$
13874 054430 103001      BCC      4$
13875
13876 054432 104044      3$:  ERROR    44          ;COND CODES SET IMPROPERLY
13877
13878 054434 020304      4$:  CMP      R3,R4      ;CORRECT RESULT IN REG+1?
13879 054436 001002      BNE      5$          ;BR IF NOT

```

```

13880 054440 020102          CMP      R1,R2          ;CORRECT RESULT IN REG?
13881 054442 001401          BEQ      TST741         ;;BR IF YES
13882
13883 054444 104045          5$:      ERROR      45          ;DIV DELIVERED WRONG RESULT
13884
13885          ;*****
13886          ;*TEST 741          DIV #N,RA TEST ; RA NEGATIVE ; N:C = 0000
13887          ;*****
13888 054446          TST741:
13889 054446 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13890 054450 012700 000741      MOV      #741,R0      ;:LOAD R0 WITH TEST NUMBER
13891 054454 013737 054504 001076  MOV      @#25,@#STMPO ;GET COPY OF TEST INSTRUCTION
13892 054462 012701 177775      MOV      #-3,R1      ;S/B RES IN R2
13893 054466 012704 177776      MOV      #-2,R4      ;S/B RES IN R3
13894 054472 012702 177777      MOV      #-1,R2      ;SET UP REG OPERAND
13895 054476 012703 177762      MOV      #-14.,R3    ;SET UP REG+1 OP
13896 054502 000257          CCC          ;SCOPE SYNC
13897
13898 054504 071227 000004      2$:      DIV      #4,R2          ;TEST DIV
13899
13900 054510 100003          BPL      3$           ;N:C=1000?
13901 054512 001402          BEQ      3$
13902 054514 102401          BVS      3$
13903 054516 103001          BCC      4$
13904
13905 054520 104044          3$:      ERROR      44          ;COND CODES SET IMPROPERLY
13906
13907 054522 020304          4$:      CMP      R3,R4          ;CORRECT RESULT IN REG+1?
13908 054524 001002          BNE      5$          ;BR IF NOT
13909 054526 020102          CMP      R1,R2          ;CORRECT RESULT IN REG?
13910 054530 001401          BEQ      TST742         ;;BR IF YES
13911
13912 054532 104045          5$:      ERROR      45          ;DIV DELIVERED WRONG RESULT
13913
13914          ;*****
13915          ;*TEST 742          DIV #N,RA TEST ; N:C = 0000 TO 0100
13916          ;*****
13917 054534          TST742:
13918 054534 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13919 054536 012700 000742      MOV      #742,R0      ;:LOAD R0 WITH TEST NUMBER
13920 054542 013737 054566 001076  MOV      @#25,@#STMPO ;GET COPY OF TEST INSTRUCTION
13921 054550 005001          CLR      R1          ;S/B RES IN R2
13922 054552 012704 000001      MOV      #1,R4        ;S/B RES IN R3
13923 054556 005002          CLR      R2          ;SET UP REG OPERAND
13924 054560 012703 000001      MOV      #1,R3        ;SET UP REG+1 OP
13925 054564 000257          CCC          ;SCOPE SYNC
13926
13927 054566 071227 000002      2$:      DIV      #2,R2          ;TEST DIV
13928
13929 054572 100403          BMI      3$           ;N:C=0100?
13930 054574 001002          BNE      3$
13931 054576 102401          BVS      3$
13932 054600 103001          BCC      4$
13933
13934 054602 104044          3$:      ERROR      44          ;COND CODES SET IMPROPERLY
13935

```

13936 054604 020304
13937 054606 001002
13938 054610 020102
13939 054612 001401
13940
13941 054614 104045
13942
13943
13944
13945
13946 054616
13947 054616 000004
13948 054620 012700 000743
13949 054624 013737 054652 001076
13950 054632 012701 177775
13951 054636 012704 000002
13952 054642 005002
13953 054644 012703 000016
13954 054650 000257
13955
13956 054652 071227 177774
13957
13958 054656 020304
13959 054660 001002
13960 054662 020102
13961 054664 001401
13962
13963 054666 104045
13964
13965
13966
13967
13968
13969
13970
13971
13972 054670
13973 054670 000004
13974 054672 012700 000744
13975 054676 013701 054716
13976 054702 012704 000002
13977 054706 005037 177776
13978 054712 012702 000050
13979
13980 054716 071227 000005
13981
13982 054722 100424
13983 054724 001423
13984 054726 102022
13985 054730 103421
13986
13987 054732 012702 177777
13988 054736 005003
13989
13990 054740 071227 177776
13991

4\$: CMP R3,R4 ;CORRECT RESULT IN REG+1?
BNE 5\$;BR IF NOT
CMP R1,R2 ;CORRECT RESULT IN REG?
BEQ TST743 ;;BR IF YES

5\$: ERROR 45 ;DIV DELIVERED WRONG RESULT

;*****
;TEST 743 DIV #N,RA TEST ; RA POS
;*****
TST743:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #743,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #25,#STMP0 ;GET COPY OF TEST INSTRUCTION
MOV #-3,R1 ;S/B RES IN R2
MOV #2,R4 ;S/B RES IN R3
CLR R2 ;SET UP REG OPERAND
MOV #14.,R3 ;SET UP REG+1 OP
CCC ;SCOPE SYNC

2\$: DIV #-4,R2 ;TEST DIV

CMP R3,R4 ;CORRECT RESULT IN REG+1?
BNE 3\$;BR IF NOT
CMP R1,R2 ;CORRECT RESULT IN REG?
BEQ TST744 ;;BR IF YES

3\$: ERROR 45 ;DIV DELIVERED WRONG RESULT

;*****
;TEST 744 DIV TEST - V BIT GETS SET
; THIS TEST TESTS THAT THE V BIT CAN BE SET IN ALL THE
; POSSIBLE WAYS. SINCE THE INSTRUCTION SHOULD BE ABORTED, THE
; RESULTS CANNOT BE GUARANTEED. FOR THIS REASON, ONLY
; THE CONDITION CODES ARE CHECKED.
;*****
TST744:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #744,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #2,R4 ;S/B PSW
CLR #PSW ;CLEAR OUT OTHER PSW BITS
MOV #50,R2 ;SET UP REG OP

2\$: DIV #5,R2 ;TEST DIV -- SHOULD ABORT

BMI 3\$;N:C=0010?
BEQ 3\$
BVC 3\$
BCS 3\$

MOV #-1,R2 ;INITIALIZE REG OP
CLR R3 ;INITIALIZE REG+1 OP

DIV #-2,R2 ;TEST DIV -- SHOULD ABORT

```

13992 054744 100413      BMI      3$          ;N:C=0010?
13993 054746 001412      BEQ      3$
13994 054750 102011      BVC      3$
13995 054752 103410      BCS      3$
13996
13997 054754 012704 000003      MOV      #3,R4      ;S/B PSW
13998
13999 054760 071227 000000      DIV      #0,R2      ;TEST DIV BY 0 -- SHOULD ABORT
14000
14001 054764 100403      BMI      3$          ;N:C=0010?
14002 054766 001402      BEQ      3$
14003 054770 102001      BVC      3$
14004 054772 103405      BCS      TST745     ;;IF ALL OK, THEN EXIT TEST
14005
14006 054774 013703 177776      3$:     MOV      @#PSW,R3 ;GET WAS PSW
14007 055000 012702 177776      MOV      #PSW,R2    ;DESTINATION IS PSW
14008
14009 055004 104001      ERROR   1          ;CONDITION CODES SET WRONG
14010
14011
14012
14013
14014 055006
14015 055006 000004      ;*****
14016 055010 012700 000745      ;*TEST 745      ASH #N,RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010
14017 055014 013701 055032      ;*****
14018 055020 012704 123450      TST745:
14019 055024 012703 112345      SCOPE
14020 055030 000257      MOV      #745,R0    ;CALL THE SCOPE LOOP UTILITY
14021
14022 055032 072327 000003      2$:     MOV      @#2$,R1    ;;LOAD R0 WITH TEST NUMBER
14023
14024 055036 100003      MOV      #123450,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
14025 055040 001402      MOV      #112345,R3 ;S/B RESULT
14026 055042 102001      MOV      #112345,R3 ;INITIAL REG
14027 055044 103001      CCC
14028
14029 055046 104002      3$:     ERROR   2          ;SCOPE SYNC
14030
14031 055050 020304      2$:     ASH      #3,R3    ;TEST THE ASH
14032 055052 001401      BPL      3$          ;N:C=1010?
14033 055054 104002      BEQ      3$
14034
14035
14036
14037
14038 055056
14039 055056 000004      3$:     ERROR   2          ;INCORRECT CONDITION CODES
14040 055060 012700 000746      4$:     CMP      R3,R4    ;CORRECT RESULT?
14041 055064 013701 055102      BEQ      TST746     ;;BR IF YES
14042 055070 005004      ERROR   2          ;ASH DELIVERED WRONG RESULT
14043 055072 012703 000004      ;*****
14044 055076 000257      ;*TEST 746      ASH #N,RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
14045 055100 000270      ;*****
14046
14047 055102 072327 177775      2$:     TST746:
SCOPE
MOV      #746,R0    ;CALL THE SCOPE LOOP UTILITY
MOV      @#2$,R1    ;;LOAD R0 WITH TEST NUMBER
CLR      R4         ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #4,R3     ;S/B RESULT
CCC        ;INITIAL REG
SEN        ;SCOPE SYNC
           ;CODES = 1000
           ;TEST THE ASH

```

G06

MAINDEC-11-DOKDA-A KD11-K BASIC LOGIC TESTS
DOKDAA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 255
ASH #N,RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101

```

14048
14049 055106 100403          BMI      3$          ;N:C=0101?
14050 055110 001002          BNE      3$
14051 055112 102401          BVS      3$
14052 055114 103401          BCS      4$
14053
14054 055116 104002          3$:      ERROR      2          ;INCORRECT CONDITION CODES
14055
14056 055120 020304          4$:      CMP        R3,R4          ;CORRECT RESULT?
14057 055122 001401          BEQ      TST747        ;:BR IF YES
14058 055124 104002          ERROR    2          ;ASH DELIVERED WRONG RESULT
14059
14060
14061
14062
14063 055126
14064 055126 000004          TST747:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
14065 055130 012700 000747          MOV      #747,R0          ;:LOAD R0 WITH TEST NUMBER
14066 055134 013701 055152          MOV      @#25,R1          ;:LOAD R1 WITH TEST INSTRUCTION WORD
14067 055140 012704 177234          MOV      #177234,R4        ;S/B RESULT
14068 055144 012703 123432          MOV      #123432,R3        ;INITIAL REG
14069 055150 000277          SCC          ;SCOPE SYNC
14070
14071 055152 072327 177772          2$:      ASH      #-6,R3          ;TEST THE ASH
14072
14073 055156 100003          BPL      3$          ;N:C=1000?
14074 055160 001402          BEQ      3$
14075 055162 102401          BVS      3$
14076 055164 103001          BCC      4$
14077
14078 055166 104002          3$:      ERROR      2          ;INCORRECT CONDITION CODES
14079
14080 055170 020304          4$:      CMP        R3,R4          ;CORRECT RESULT?
14081 055172 001401          BEQ      TST750        ;:BR IF YES
14082 055174 104002          ERROR    2          ;ASH DELIVERED WRONG RESULT
14083
14084
14085
14086
14087 055176
14088 055176 000004          TST750:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
14089 055200 012700 000750          MOV      #750,R0          ;:LOAD R0 WITH TEST NUMBER
14090 055204 013737 055234 001076          MOV      @#25,@#STMPO      ;GET TEST INSTRUCTION WORD
14091 055212 012701 123456          MOV      #123456,R1        ;S/B RES IN R2
14092 055216 012704 076530          MOV      #076530,R4        ;S/B RES IN R3
14093 055222 012702 112345          MOV      #112345,R2        ;INITIALIZE COMBINED
14094 055226 012703 147653          MOV      #147653,R3        ;:REGISTERS
14095 055232 000257          CCC          ;SCOPE SYNC
14096
14097 055234 073227 000003          2$:      ASHC     #3,R2          ;TEST ASHC
14098
14099 055240 100003          BPL      3$          ;N:C=1010?
14100 055242 001402          BEQ      3$
14101 055244 102001          BVC      3$
14102 055246 103001          BCC      4$
14103

```


H06

MAINDEC-11-DGKDA-A KD11-K BASIC LOGIC TESTS
DGKDA.P11 08-FEB-77 16:17

MACY11 27(1006) 08-FEB-77 16:23 PAGE 256
ASHC #N,RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010

```

14104 055250 104044      3$:  ERROR  44          ;COND CODES WRONG
14105
14106 055252 020102      4$:  CMP     R1,R2      ;TOP HALF OF RESULT CORRECT?
14107 055254 001002          BNE     5$           ;BR IF NOT
14108 055256 020403          CMP     R4,R3      ;LOWER HALF OF RESULT CORRECT?
14109 055260 001401          BEQ     T$T751     ;BR IF YES
14110 055262 104045      5$:  ERROR  45          ;ASHC DELIVERED WRONG RES
14111
14112
14113  ;*****
14114  ;*TEST 751      ASHC #N,RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
14115  ;*****
14116 055264 000004      T$T751:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
14117 055266 012700 000751  MOV     #751,R0    ;LOAD R0 WITH TEST NUMBER
14118 055272 013737 055316 001076  MOV     @#2$,@#STMPD ;GET TEST INSTRUCTION WORD
14119 055300 005001          CLR     R1         ;S/B RES IN R2
14120 055302 005004          CLR     R4         ;S/B RES IN R3
14121 055304 005002          CLR     R2         ;INITIALIZE COMBINED
14122 055306 012703 000005  MOV     #5,R3      ;REGISTERS
14123 055312 000257          CCC          ;SCOPE SYNC
14124 055314 000270          SEN          ;CODES = 1000
14125
14126 055316 073227 177775      2$:  ASHC   #-3,R2    ;TEST ASHC
14127
14128 055322 100403          BMI     3$         ;N:C=0101?
14129 055324 001002          BNE     3$
14130 055326 102401          BVS     3$
14131 055330 103401          BCS     4$
14132
14133 055332 104044      3$:  ERROR  44          ;COND CODES WRONG
14134
14135 055334 020102      4$:  CMP     R1,R2      ;TOP HALF OF RESULT CORRECT?
14136 055336 001002          BNE     5$           ;BR IF NOT
14137 055340 020403          CMP     R4,R3      ;LOWER HALF OF RESULT CORRECT?
14138 055342 001401          BEQ     T$T752     ;BR IF YES
14139 055344 104045      5$:  ERROR  45          ;ASHC DELIVERED WRONG RES
14140
14141
14142  ;*****
14143  ;*TEST 752      ASHC #N,RA TEST ; SHIFT RIGHT ; N:C = 1111 TO 1000
14144  ;*****
14145 055346 000004      T$T752:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
14146 055350 012700 000752  MOV     #752,R0    ;LOAD R0 WITH TEST NUMBER
14147  ;SBTTL USER CONTROLLED BREAKPOINT -- BIT15
14148 055354 032737 100000 063160  BIT     #BIT15,@#BPTLOC ;BREAKPOINT HALT SET ??
14149 055362 001401          BEQ     .+4        ;BR IF NOT
14150 055364 000000          HALT          ;BREAK-DEPRESS CONTINUE TO CONTINUE
14151 055366 013737 055416 001076  MOV     @#2$,@#STMPD ;GET TEST INSTRUCTION WORD
14152 055374 012701 177234  MOV     #177234,R1 ;S/B RES IN R2
14153 055400 012704 135275  MOV     #135275,R4 ;S/B RES IN R3
14154 055404 012702 123456  MOV     #123456,R2 ;INITIALIZE COMBINED
14155 055410 012703 127542  MOV     #127542,R3 ;REGISTERS
14156 055414 000257          CCC          ;SCOPE SYNC
14157
14158 055416 073227 177772      2$:  ASHC   #-6,R2    ;TEST ASHC
14159

```

```

14160 055422 100003          BPL      3$          ;N:C=1000?
14161 055424 001402          BEQ      3$
14162 055426 102401          BVS      3$
14163 055430 103401          BCS      4$
14164
14165 055432 104044          3$:      ERROR     44          ;COND CODES WRONG
14166
14167 055434 020102          4$:      CMP      R1,R2          ;TOP HALF OF RESULT CORRECT?
14168 055436 001002          BNE      5$          ;BR IF NOT
14169 055440 020403          CMP      R4,R3          ;LOWER HALF OF RESULT CORRECT?
14170 055442 001401          BEQ      TST753        ;BR IF YES
14171 055444 104045          5$:      ERROR     45          ;ASHC DELIVERED WRONG RES
14172
14173
14174
14175
14176
14177
14178
14179
14180
14181
14182
14183
14184
14185
14186
14187
14188
14189
14190
14191
14192
14193 055446
14194 055446 012700 000752          TST753:  MOV      #752,R0          ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
14195 055452 000004          SCOPE          ;:CALL THE SCOPE LOOP UTILITY
14196 055454 012737 000304 177770  MED1:  MOV      #304,#UBREAK        ;:SET SCOPE SYNC FOR MED INSTR
14197 055462 012737 140000 177776  MOV      #140000,#PSW        ;:GO TO USER MODE
14198 055470 012706 001000          MOV      #STACK,#SP          ;:SETUP USER STACK PTR.
14199 055474 012737 055526 000004  MOV      #2$,#ERRVEC          ;:SET ERROR TRAP VECTOR TO 2$ BELOW
14200 055502 012737 055526 000010  MOV      #2$,#RESVEC          ;:LOAD RESERVED INST. TRAP VECTOR
14201 055510 012701 177777          MOV      #-1,R1              ;:LOAD R1 WITH A -1
14202 055514 005000          CLR      R0                  ;:CLEAR R0
14203 055516 076600          MED          ;:TRY TO DO MAINT. EXAMINE
14204 055520 000041          .WORD    041                  ;:MED READ CODE FOR R1
14205 055522 104012          ERROR    12                  ;:ERROR - MED INST. NOT ILLEGAL IN USER
14206 055524 000404          BR      4$
14207 055526 005700          2$:      TST      R0              ;:IS R0 UNCHANGED?
14208 055530 001401          BEQ      3$                  ;:BRANCH IF YES
14209 055532 104013          ERROR    13                  ;:ERROR - MED INSTRUCTION WAS EXECUTED
14210
14211 055534 022626          3$:      CMP      (SP)+,(SP)+        ;:BEFORE TRAPPING
14212 055536 012737 061144 000004  4$:      MOV      #BERR,#ERRVEC        ;:CLEAN UP STACK
14213 055544 012737 061046 000010  MOV      #RSERR,#RESVEC        ;:RESTORE ERROR TRAP VECTOR
14214
14215 055552 005037 177776          MED0:  CLR      #PSW                ;:RESTORE RESERVED INST. TRAP VECTOR
;:GO TO KERNEL MODE,CLEAR COND. CODES

```

```

: *
: * THIS SECTION OF THE MED TESTS EXERCISES CERTAIN SCRATCH
: * PAD REGISTERS USING MED READS AND WRITES. THEIR ORIGINAL
: * CONTENTS ARE RESTORED BUT:
: *
: * ***** IMPORTANT NOTE *****
: *
: * THE CONSOLE MUST NOT !!! BE USED DURING THESE MED
: * TESTS. NO INTERRUPTS OR TRAPS CAN BE ALLOWED EITHER*
: *
: * *****

```

```

: *****
: *TEST 753 CHECK MED IS ILLEGAL IN USER - EXECUTES IN KERNAL
: * THE NEXT TEST BELOW CHECKS TO SEE THAT THE "MED"
: * (MAINTENANCE, EXAM, AND DEPOSIT) INSTRUCTION WILL EXECUTE
: * WHEN IN KERNEL MODE WITHOUT AFFECTING THE PSW AND
: * THAT IT IS ILLEGAL IN USER MODE
: *****

```

J06

MAINDEC-11-DOKDA-A KD11-K BASIC LOGIC TESTS
DOKDAA.P11 08-FEB-77 16:17 T753

MACY11 27(1006) 08-FEB-77 16:23 PAGE 258
CHECK MED IS ILLEGAL IN USER - EXECUTES IN KERNAL

14216 055556 076600
14217 055560 000041
14218 055562 103403
14219 055564 102402
14220 055566 100401
14221 055570 001001
14222 055572 104014

MED ;DO MAINT. EXAMINE OF R1
.WORD 041 ;MED READ CODE FOR R1
BCS MEDHLT
BVS MEDHLT
BMI MEDHLT
BNE +4
MEDHLT: ERROR 14 ;ERROR CC-BITS IN PSW AFFECTED BY MED

*TEST 754 MED TEST - R/W DATA PATTERNS TO REGS
* THIS PARTICULAR MED TEST WRITES DATA PATTERNS
* TO THOSE INTERNAL REGS. WHICH CAN BE WRITTEN
* AND READ WITHOUT SPECIAL CONSIDERATIONS. REGISTERS
* REQUIRING SPECIAL TESTS ARE TESTED IN LATER
* MED TESTS.
* TABLE II CONTAINS THE REGISTER ADDRESSES.
*
* A MAX. OF 3 ERRORS ARE REPORTED FOR EACH LOC.

14223
14224
14225
14226
14227
14228
14229
14230
14231
14232
14233
14234
14235
14236 055574
14237 055574 012700 000753
14238 055600 000004
14239 055602 012737 000340 177776
14240 055610 012701 064112
14241 055614 012737 125252 001102
14242 055622 111137 055670
14243 055626 112137 055712
14244 055632 111137 055650
14245 055636 112137 055676
14246 055642 005037 001106
14247 055646 076600
14248 055650 000000
14249 055652 010037 001076
14250 055656 010137 001100
14251 055662 013700 001102
14252 055666 076600
14253 055670 000000
14254 055672 005000
14255 055674 076600
14256 055676 000000
14257 055700 010037 001104
14258 055704 013700 001076
14259 055710 076600
14260 055712 000000
14261 055714 023737 001102 001104
14262 055722 001412
14263 055724 013737 055676 001100
14264 055732 022737 000003 001106
14265 055740 002401
14266 055742 104022
14267 055744 005237 001106
14268 055750 005137 001102
14269 055754 013701 001100
14270 055760 022737 125252 001102
14271 055766 001327

ST754:
MOV #753,RO ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;:CALL THE SCOPE LOOP UTILITY
MEDT1: MOV #340,PSW ;:KERNEL MODE-PRIORITY 7
MOV #TBL2,R1 ;:INITIALIZE ADDRESS POINTER
15: MOV #125252,STMP2 ;:PUT WRITE CODE BY "WRITE-MED'S"
MOV (R1),R1S ;:AND POINT R1 TO READ CODE
MOV (R1)+,R1S3 ;:PUT READ CODE BY "READ-MED'S"
MOV (R1)+,R1S ;:R1 NOW POINTS TO NEXT REG.
CLR STMP4 ;:CLEAR ERROR COUNTER
25: MED ;:MED-READ THE INTERNAL REG.
105: .WORD 0 ;:MED-READ CODE
MOV RO,STMP0 ;:SAVE ITS ORIGINAL CONTENTS
MOV R1,STMP1 ;:SAVE ADDR. PTR. VALUE
MOV STMP2,RO ;:LOAD RO WITH DATA TO BE WRITTEN
MED ;:MED-WRITE THE TEST DATA
115: .WORD 0 ;:MED-WRITE CODE
CLR RO ;:CLEAR RO
MED ;:MED-READ THE DATA BACK
125: .WORD 0 ;:MED-READ CODE
MOV RO,STMP3 ;:SAVE DATA READ FOR COMPARISON
MOV STMP0,RO ;:LOAD ORIGINAL DATA IN RO
MED ;:MED-WRITE ORG. DATA TO REG.
135: .WORD 0 ;:MED-WRITE CODE
CMP STMP2,STMP3 ;:DID DATA READ=DATA WRITTEN?
BEQ 35 ;:BRANCH IF YES
MOV #125,STMP1 ;:SAVE MED-CODE FOR ERROR
CMP #3,STMP4 ;:MAX. ERROR REPORTS YET?
BLT 145 ;:BRANCH IF YES
ERROR 22 ;:INT. REG. READ BACK WRONG DATA
145: INC STMP4 ;:INCREMENT ERROR COUNTER
35: COM STMP2 ;:CHANGE DATA PATTERN
MOV STMP1,R1 ;:RESTORE ADDR. POINTER
CMP #125252,STMP2 ;:BOTH DATA PATTERNS BEEN USED?
BNE 25 ;:BRANCH IF NO

14272 055770 005711
14273 055772 001310

TST (R1) ;END OF ADDR. TABLE?
BNE 15 ;BRANCH IF NO

14274
14275
14276
14277
14278
14279
14280
14281
14282
14283
14284
14285
14286

:TEST 755 MED TEST - VERIFY NOPS; READ R7 IN A & B SP
: THIS TEST CHECKS ALL OF THE "NOP" OPERATION CODES
: TO ENSURE THEY WILL EXECUTE AS NOP'S AND
: NOT RESULT IN A PROCESSOR HANG. THE "NOPS"
: TABLE (TABLE III) HOLDS THESE CODES.
: THIS TEST ALSO READS THE PROGRAM COUNTER (R7) VALUES
: STORED IN A & B SCRATCH PADS TO SEE THAT THEY
: READ PROPERLY. THE R7 ADDRESSES ARE IN TABLE IV.

14287 055774
14288 055774 012700 000754
14289 056000 000004
14290 056002 012701 064330
14291 056006 112137 056014
14292
14293 056012 076600
14294 056014 000000
14295 056016 123711 056014
14296
14297 056022 103003
14298 056024 005237 056014
14299 056030 000770
14300 056032 105721
14301 056034 005711
14302 056036 001363
14303
14304 056040 113737 064351 056052
14305 056046 005000
14306 056050 076600
14307 056052 000000
14308 056054 020027 056054
14309 056060 001411
14310 056062 013737 056052 001100
14311 056070 012737 056054 001102
14312 056076 010037 001104
14313 056102 104022
14314 056104 023727 056052 000047
14315 056112 001404
14316 056114 113737 064355 056052
14317 056122 000751
14318 056124

TST755:
MOV #754,R0 ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;:CALL THE SCOPE LOOP UTILITY
MEDT3: MOV #TBL3,R1 ;:INITIALIZE NOP TABLE PTR. (R1)
15: MOVB (R1)+,2#105 ;:PLACE FIRST "NOP-CODE" AFTER MED
;:AND POINT R1 TO LAST CODE IN GROUP
55: MED ;:EXECUTE MED WITH NOP OP-CODE
105: .WORD 0
CMPB 2#105,(R1) ;:HAVE ALL NOPS IN THAT GROUP
;:BEEN TESTED?
;:BRANCH IF YES
;:NEXT NOP IN GROUP
65: BHS 65
INC 2#105
BR 55
TSTB (R1)+ ;:POINT R1 TO NEXT NOP GROUP
TST (R1) ;:HAVE ALL GROUPS BEEN TESTED
BNE 15 ;:BRANCH IF NO
MEDT4: MOVB 2#R7A+1,2#55 ;:LOAD R7A READ CODE AFTER MED
45: CLR R0 ;:CLEAR R0
MED ;:MED READ R7 IN THE ASP
55: .WORD 0 ;:READ CODE FOR R7A
CMP R0,#55+2 ;:DID R7A READ CORRECTLY?
BEQ 65 ;:BRANCH IF YES
MOV 2#55,2#STMP1 ;:SAVE MED-CODE FOR ERROR
MOV #55+2,2#STMP2 ;:SAVE DATA EXPECTED
MOV R0,2#STMP3 ;:SAVE DATA RECEIVED
ERROR 22 ;:R7A DID NOT READ THE RIGHT VALUE
65: CMP 2#55,#47 ;:HAS R7B BEEN CHECKED?
BEQ 85 ;:BRANCH IF YES
MOVB 2#R7B+1,2#55 ;:LOAD R7B READ CODE AFTER MED
BR 45 ;:TEST R7 BSP
85:

14319
14320
14321
14322
14323
14324
14325
14326
14327

:TEST 756 MED TEST - CSP CONSTANTS CHECK
: THIS TEST CHECKS THE CONSTANT VALUES LOCATED
: IN THE C SCRATCH PAD. THE CONSTANTS ARE READ
: WITH A MED INSTRUCTION AND COMPARED TO THEIR
: EXPECTED VALUE. THE ADDRESSES OF THESE CONSTANTS

14328
14329
14330
14331 056124
14332 056124 012700 000755
14333 056130 000004
14334
14335 056132 170000
14336
14337 056134 012701 064462
14338 056140 012167 000006
14339 056144 001414
14340 056146 005000
14341 056150 076600
14342 056152 000000
14343 056154 020021
14344 056156 001770
14345 056160 013737 056152 001100
14346 056166 016137 177776 001102
14347 056174 104021
14348 056176
14349
14350
14351
14352
14353
14354
14355
14356
14357
14358
14359
14360
14361
14362
14363
14364
14365
14366
14367
14368 056176
14369 056176 012700 000756
14370 056202 000004
14371 056204 012737 000071 177770
14372 056212 012737 061030 000004
14373 056220 012737 000340 000006
14374 056226 005037 061036
14375 056232 076600
14376 056234 000022
14377 056236 052700 001000
14378 056242 076600
14379 056244 000222
14380 056246 076600
14381 056250 000144
14382 056252 052700 100000
14383 056256 076600

AND THE VALUES EXPECTED ARE IN TABLE VII.

TST756: MOV #755,RO ;: SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;: CALL THE SCOPE LOOP UTILITY
MEDT10: CFCC ;: EXECUTE FLT. PT INST. SO FLT. PT.
;: CONSTANTS ARE LOADED INTO CSP
10S: MOV #TBL7,R1 ;: SETUP TABLE POINTER
MOV (R1)+,1S ;: LOAD MED READ CODE AT 1S
BEQ 11S ;: BR IF END OF TABLE
CLR RO
MED ;: READ INTERNAL CONTENTS INTO RO
1S: .WORD 0
CMP RO,(R1)+ ;: WAS THE CONSTANT READ THE ONE EXPECTED
BEQ 10S ;: BRANCH IF YES
MOV @#1S,@#STMP1 ;: SAVE MEDCODE FOR ERROR
MOV -2(R1),@#STMP2 ;: SAVE CONSTANT VALUE EXPECTED
ERROR 21 ;: CSP LOCATION HELD WRONG VALUE
11S:

TEST 757 MED TEST - MICROBK CHECK OF MICRO-POINTS

THIS TEST USES THE MICROBREAK REGISTER AND THE
INFORMATION IN TABLE V TO CHECK THAT THE
CORRECT MED-FLOW IS ENTERED WHEN EACH
REGISTER IS ACCESSED BY A MED INSTRUCTION.
THE MICROBREAK REG. IS SETUP TO CAUSE A TRAP TO
LOC. 4 WHEN ITS CONTENTS EQUAL THE ADDRESS
OF THE MIRCOWORD BEING EXECUTED.
NOTE: THE MICRO BREAK - TRAP-TO-4 CAPABILITY
IS TRIED AT THE BEGINNING OF THE TEST.
IF IT DOESN'T WORK, AN ERROR IS PRINTED
AND THE TEST IS SKIPPED

TST757: MOV #756,RO ;: SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;: CALL THE SCOPE LOOP UTILITY
MEDT11: MOV #SWB01,@#UBREAK ;: LOAD MICROBK. REG. WITH AN MICRO ADDR.
MOV #BKROUT,@#4 ;: LOAD ADDR. OF MICROBK. ROUTINE IN 4
MOV #340,@#6 ;: LOAD KERNEL PSM - PRIORITY 7 IN 6
CLR @#BKFLAG ;: CLEAR MICROBK. TRAP FLAG
MED ;: GET WHAMI INTO RO
RDWHAMI
BIS #BIT9,RO ;: SET BIT 9
MED ;: MED-WRITE THE WHAMI REG TO
10S: WRWHAMI ;: ENABLE MICROBK-TRAP-TO-4
MED ;: GET FLAG REGISTER
RDFLAG
BIS #BIT15,RO ;: SET BIT 15 IN RO
MED ;: MED-WRITE THE FLAG REG TO

M06

MAINDEC-11-DOKDA-A KD11-K BASIC LOGIC TESTS
 DOKDAA.P11 08-FEB-77 16:17 T757

MACY11 27(1006) 08-FEB-77 16:23 PAGE 261
 MED TEST - MICROBK CHECK OF MICRO-POINTS

```

14384 056260 000344      11S:  WRFLAG      ;ENABLE MICROBK TRAPPING
14385 056262 000300      SWAB        RO      ;MICROBK TRAP SHOULD OCCUR ON SWAB
14386 056264 005737 061036  TST        @BKFLAG ;DID TRAP TO 4 OCCUR?
14387 056270 001007      BNE        15      ;BRANCH IF YES
14388 056272 005037 001076  CLR        @STMP0
14389 056276 016737 121567 001100  MOV        SWB01,@STMP1 ;SAVE EXPECTED UBREAK ADDR
14390 056304 104015      ERROR      15      ;MICROBREAK TRAP DIDN'T WORK
14391 056306 000453      BR         50S     ;SKIP TO END OF TEST
14392
14393 056310 012701 000710  1S:   MOV        #SWB01#10,R1 ;GET CORRECT U-ADDR
14394 056314 076600      MED        ;GET LOG CUA REG
14395 056316 000103      RDL CUA
14396 056320 042700 100007  BIC        #100007,RO ;GET RID OF IRRELEVANT BITS
14397 056324 020001      CMP        RO,R1   ;WAS CORRECT UADDR LOGGED?
14398 056326 001401      BEQ        3S     ;BR IF YES
14399 056330 104025      ERROR      25      ;CUA CONTAINS INCORRECT U-ADDR
14400 056332 012701 064362  3S:   MOV        @TBL5,R1   ;INITIALIZE TABLE PTR. (R1)
14401 056336 012702 064410  MOV        @TBL6,R2
14402 056342 010737 001010  MOV        PC,@SLPERR ;SET ERROR LOOP RETURN TO 2S
14403 056346 111137 056404  2S:   MOVB      (R1),@12S ;LOAD WRITE CODE AFTER MED
14404 056352 001431      BEQ        50S     ;BR IF END OF TABLE
14405 056354 011237 177770  4S:   MOV        (R2),@UBREAK ;LOAD MICROBK REG. WITH MICROADDR.
14406 056360 005037 061036  CLR        @BKFLAG   ;CLEAR MICROBK TRAP-TO-4 FLAG
14407 056364 076600      MED        ;GET FLAG REGISTER
14408 056366 000144      RDL FLAG
14409 056370 052700 100000  BIS        #BIT15,RO ;SET BIT 15 IN RO
14410 056374 076600      MED        ;MED WRITE TO FLAG REG TO
14411 056376 000344  15S:  WRFLAG      ;ENABLE MICROBK TRAPPING
14412 056400 005000      CLR        RO      ;IN CASE U-BREAK TRAP DOESN'T OCCOR
14413
14414 056402 076600      MED        ;USUALLY BETTER TO WRITE 0'S
14415 056404 000000  12S:  .WORD      0
14416 056406 005737 061036  TST        @BKFLAG   ;DID WE TRAP-TO-4? (FLAG NOT = 0)
14417 056412 001006      BNE        20S     ;BRANCH IF YES TO NEXT ENTRY
14418 056414 013737 056404 001076  MOV        @12S,@STMP0 ;SAVE MED-CODE FOR ERROR
14419 056422 011237 001100  MOV        (R2),@STMP1 ;SAVE EXPECTED U-ADDR FOR ERROR
14420 056426 104015      ERROR      15      ;MICROBK. TRAP-TO-4 DID NOT OCCUR
14421
14422 056430 105721  20S:  TSTB      (R1)+    ;INCREMENT TO NEXT TABLE
14423 056432 005722      TST        (R2)+    ;ENTRIES AND
14424 056434 000744      BR         2S     ;CONTINUE
14425
14426 056436 076600  50S:  MED        ;GET WHAMI INTO RO
14427 056440 000022      RDWHAMI
14428 056442 042700 001000  BIC        #BIT9,RO  ;CLEAR BIT 9
14429 056446 076600      MED        ;CLEAR THE FLAG REG. TO
14430 056450 000344  13S:  WRFLAG      ;DISABLE MICROBK. TRAPPING
14431 056452 076600      MED        ;CLEAR THE WHAMI REG. TO
14432 056454 000222  14S:  WRWHAMI     ;DISABLE MICROBK. TRAP-TO-4
14433 056456 012737 056204 001010  MOV        #MEDT11,@SLPERR ;RESET LOOP ON ERROR POINTER
14434 056464 012737 061144 000004  MOV        #BERR,@4   ;RESTORE NORMAL ERROR ROUTINE
14435 056472 012737 000304 177770  MOV        #304,@UBREAK ;GENERATE SYNC PULSE ON MED INSTR
14436
14437
14438
14439
;*****
; *TEST 760      PHYSICAL ADDRESS & ODD ADDRESS ERROR LOGGING
    
```

```

14440 ;* THIS TEST CHECKS THAT THE PROPER PHYSICAL ADDRESS BITS
14441 ;* <17:00> ARE LOGGED UPON ERROR. THE ERROR IS CAUSED BY
14442 ;* FORCING AN ODD ADDRESS TRAP. THE ERROR LOG MODE USED
14443 ;* IS "LOG FIRST". ALSO, THE ODD ADDRESS ERROR BITS IN
14444 ;* THE LOG JAM AND CPU ERROR REGISTER ARE CHECKED.
14445 ;* *****
14446 056500 TST760: MOV #757,R0 ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
14447 056500 012700 000757 SCOPE ;:CALL THE SCOPE LOOP UTILITY
14448 056504 000004 1S: MOV #25,2#4 ;:SETUP PC FOR ODD ADDR SERVICE
14449 056506 012737 056546 000004 MOV #340,2#6
14450 056514 012737 000340 000006 MOV #BIT15+BIT0,R0 ;:SETUP "LOG FIRST" MODE
14451 056522 012700 100001 MED
14452 056526 076600 WRMHAMI
14453 056530 000222 MOV #15+1,R2 ;:SAVE ADDRESS OF ODD ADDR. INSTRUCTION
14454 056532 012702 056507 ;:DO ODD ADDRESS INSTRUCTION TO FORCE
14455 056536 005767 177745 TST 15+1 ;:A JAMUPP & TRAP TO 4
14456 ;:*** ODD ADDR. TRAP DID NOT OCCUR
14457 056542 104023 ERROR 23 ;:EXIT TEST
14458 056544 000441 BR 10$ ;:RESTORE STACK
14459 056546 022626 2S: CMP (SP)+,(SP)+ ;:RESTORE OLD PC & PSW
14460 056550 012737 061144 000004 MOV #BERR,2#4
14461 056556 076600 MED
14462 056560 000100 RDLJAM
14463 056562 013701 177766 MOV 2#CPUERR,R1
14464 056566 032701 000100 BIT #BIT6,R1 ;:WAS ODD ADDR. ERROR RECORDED BY
14465 ;:THE CPU ERROR REGISTER?
14466 056572 001001 BNE 3$ ;:BRANCH IF YES
14467 056574 104024 ERROR 24 ;:*** CPU ERROR REG. DID NOT
14468 ;:REPORT ODD ADDRESS ERROR
14469 ;:READ THE LOG JAM REGISTER
14470 056576 032700 100004 3S: BIT #BIT15+BIT2,R0 ;:WAS ODD ADDR. ERROR LOGGED BY LOG JAM
14471 056602 001001 BNE 4$ ;:BRANCH IF YES
14472 056604 104024 ERROR 24 ;:*** LOG JAM REG. DID NOT LOG
14473 ;:ODD ADDRESS ERROR CORRECTLY
14474 ;:
14475 056606 005005 4S: CLR R5 ;:CLR ERROR FLAG
14476 056610 076600 MED ;:READ THE LOG PBA REGISTER
14477 056612 000102 RDLPBA
14478 056614 010003 MOV R0,R3 ;:SAVE RECEIVED PHYS ADDR <15:0>
14479 056616 020002 CMP R0,R2 ;:WERE BITS <15:00> OF THE PHYSICAL
14480 ;:BUS ADDR. LOGGED CORRECTLY?
14481 056620 001401 BEQ 5$ ;:BRANCH IF YES
14482 056622 005205 INC R5 ;:SET ERROR FLAG
14483 056624 076600 5S: MED ;:READ THE LOG SERVICE REGISTER
14484 056626 000101 RDLSERVICE
14485 056630 000300 SWAB R0 ;:GET "PBA 17&16" DOWN TO BIT POSITION 0&1
14486 056632 042700 177774 BIC #177774,R0
14487 056636 001002 BNE 11$ ;:BR IF PHYS ADDR BITS <17:16> LOGGED CORRECTLY
14488 056640 005705 TST R5 ;:PREVIOUS ERROR?
14489 056642 001402 BEQ 10$ ;:BR IF NOT
14490 056644 005001 11S: CLR R1 ;:SET UP EXPECTED PA<17:16>
14491 056646 104026 ERROR 26 ;:*** PHYSICAL BUS ADDR. <17:00>
14492 ;:NOT LOGGED CORRECTLY WHEN
14493 ;:ODD ADDRESS TRAP OCCURRED
14494 056650 005000 10S: CLR R0
14495 056652 076600 MED ;:DISABLE "LOG FIRST" MODE

```

WRUHAMI

```

14496 056654 000222
14497
14498
14499
14500
14501
14502
14503
14504
14505
14506
14507
14508
14509 056656
14510 056656 012700 000760
14511 056662 000004
14512
14513 056664 012701 063772
14514 056670 005711
14515 056672 012737 000100 177746
14516 056700 012711 125252
14517 056704 012737 000001 177746
14518
14519 056712 012737 056752 000114
14520 056720 012737 000340 000116
14521 056726 005000
14522 056730 076600
14523 056732 000302
14524 056734 076600
14525 056736 000306
14526 056740 076600
14527 056742 000307
14528 056744 005767 005022
14529 056750 000406
14530 056752 012700 000200
14531 056756 076600
14532 056760 000352
14533 056762 022626
14534 056764 104030
14535
14536 056766 012700 000200
14537 056772 076600
14538 056774 000352
14539 056776 012711 125252
14540 057002 012737 000116 000114
14541 057010 005037 000116
14542 057014 005005
14543 057016 076600
14544 057020 000102
14545 057022 010003
14546
14547 057024 001401
14548 057026 005205
14549 057030 076600
14550 057032 000106
14551 057034 010001

```

```

*****
; *TEST 761 CHECK DISABLE PARITY ERROR TRAP
; *THIS TEST CHECKS THAT PARITY ERROR TRAPS TO LOCATION 114
; *ARE DISABLED WHEN BIT0 OF THE CACHE CONTROL REGISTER IS
; *SET (=1). A TRAP TO 114 SHOULD NOT OCCUR AND ERROR
; *INFORMATION SHOULD NOT BE LOGGED IN THE LOG PBA, LOG
; *CACHE DATA, OR LOG TAG DATA REGISTERS. WRONG PARITY IS
; *WRITTEN INTO A TEST LOCATION TO CAUSE THE PARITY ERROR
; *NEEDED IN THIS TEST.
*****

```

```

;TST761:
MOV #760,R0 ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;:CALL THE SCOPE LOOP UTILITY

MOV #TLOC1,R1 ;:GET POINTER TO TEST LOCATION
TST (R1) ;:MAKE IT A HIT
MOV #WMP,@CCR ;:SET WRITE WRONG PARITY BIT
MOV #125252,(R1) ;:WRITE TO TEST LOC. WITH WRONG PARITY
MOV #DPTRP,@CCR ;:DISABLE PARITY ERROR TRAPS
;AND CLEAR WMP
;SETUP PARITY ERROR VECTOR

MOV #15,@#114
MOV #340,@#116
CLR RO
MED ;:CLEAR LOG PBA REGISTER
WRLPBA ;:CLEAR LOG CACHE DATA REGISTER
MED ;:CLEAR LOG CACHE TAG REGISTER
WRLDATA
MED
WRLTAG
TST TLOC1 ;:READ TEST LOC0 TO FORCE PARITY ERROR
BR 2$ ;:BRANCH IF NO TRAP OCCURS

1$: MOV #200,R0
MED ;:CLEAN UP THE CACHE
352 ;:INITIALIZATION CODE
CMP (SP)+,(SP)+ ;:CLEAN UP STACK
ERROR 30 ;:*** PARITY TRAP TO 114 OCCURRED
;WHEN IT SHOULD HAVE BEEN DISABLED

2$: MOV #200,R0
MED ;:CLEAN UP THE CACHE
352 ;:INITIALIZATION CODE
MOV #125252,(R1) ;:WRITE BAK GOOD PARITY IN TST LOC
MOV #116,@#114 ;:RESTORE ORIGINAL PARITY HANDLER & PSW
CLR @#116
CLR R5 ;:CLEAR ERROR FLAG
MED ;:READ LOG PBA REGISTER
RDLPBA
MOV RO,R3 ;:SAVE COPY
;LOG PBA REG. STILL CLEAR?
BEQ 3$ ;:BRANCH IF YES
INC R5 ;:OTHERWISE SET ERROR FLAG
MED ;:READ LOG CACHE DATA REG.
RDLDATA
MOV RO,R1 ;:SAVE COPY

```



```

14552                                     ; LOG CACHE DATA REG. STILL CLEAR?
14553 057036 001401                       BEQ      4$
14554 057040 005205                       INC      R5
14555 057042 076600                       4$:    MED
14556 057044 000107                       RDLTAG
14557 057046 010002                       MOV      R0,R2
14558                                     ; SAVE COPY
14559 057050 001401                       BEQ      5$
14560 057052 005205                       INC      R5
14561 057054 005705                       5$:    TST      R5
14562 057056 001401                       BEQ      6$
14563 057060 104027                       ERROR    27
14564                                     ; LOG CACHE TAG REG. STILL CLEAR?
14565                                     ; BRANCH IF YES
14566                                     ; OTHERWISE SET ERROR FLAG
14567 057062 005037 177746                6$:    CLR      @#CCR
14568                                     ; WERE ANY OF LOG REGISTERS CHANGED
14569                                     ; BRANCH IF NO
14570                                     ; *** ONE OF LOG REGISTERS CHANGED
14571                                     ; WHEN ERROR SHOULD NOT HAVE BEEN LOGGED
14572                                     ; LOG PBA, LOG DATA & LOG TAG
14573                                     ; REGISTER SHOULD BE CLEAR.
14574                                     ; ENABLE PARITY ERROR TRAPS
14575                                     ;*****
14576                                     ;*TEST 762      CHECK PARITY ERROR BITS IN MEMERR REG. IN BACKUP MODE OF CACHE (TRAP)
14577                                     ;*****
14578                                     ;*THIS TEST CHECKS THAT ALL OF THE PARITY ERROR BITS (5,6,7)
14579                                     ;*OF THE MEMORY ERROR REGISTER ARE SET TO "1" WHEN A CACHE
14580                                     ;*PARITY ERROR OCCURS IN THE BACKUP MODE.
14581                                     ;*****
14582 057066 012700 000761                †ST762: MOV      #761,R0
14583 057066 000004                       SCOPE
14584 057072 012701 063772                MOV      @TLOC1,R1
14585 057100 005711                       TST      (R1)
14586 057102 012737 000100 177746        MOV      @WMP,@#CCR
14587 057110 012711 125252                MOV      #125252,(R1)
14588 057114 042737 000100 177746        BIC      @WMP,@#CCR
14589 057122 012737 057156 000114        MOV      #15,@#114
14590 057130 012737 000340 000116        MOV      #340,@#116
14591 057136 005737 063772                TST      @#TLOC1
14592 057142 012700 000200                MOV      #200,R0
14593 057146 076600                       MED
14594 057150 000352                       352
14595 057152 104031                       ERROR    31
14596 057154 000405                       BR      2$
14597 057156 012700 000200                1$:    MOV      #200,R0
14598 057162 076600                       MED
14599 057164 000352                       352
14600 057166 022626                       CMP      (SP)+,(SP)+
14601 057170 022737 000340 177744        2$:    CMP      #000340,@#MEMERR
14602                                     ; WERE PARITY ERROR BITS (5,6,7) SET
14603                                     ; AND CPU ABORT BIT (15) LEFT CLEAR
14604                                     ; IN MEMORY ERROR REGISTER?
14605                                     ; BRANCH IF YES
14606 057176 001403                       BEQ      3$
14607 057200 013700 177744                MOV      @#MEMERR,R0
14608 057204 104032                       ERROR    32
14609                                     ; *** MEMORY ERROR REGISTER BITS
14610                                     ; WERE SET INCORRECTLY
14611 057206 012737 000116 000114        3$:    MOV      #116,@#114
14612 057214 005037 000116                CLR      @#116
14613                                     ;*****
14614                                     ;*TEST 763      CHECK UNIBUS TIMEOUT, ODD ADDRESS AND LOG CONTINUOUS MODE
14615                                     ;*****

```

14608
14609
14610
14611
14612
14613
14614
14615
14616
14617
14618
14619
14620
14621
14622
14623
14624
14625
14626
14627
14628
14629
14630
14631
14632
14633
14634
14635
14636
14637
14638
14639
14640
14641
14642
14643
14644
14645
14646
14647
14648
14649
14650
14651
14652
14653
14654
14655
14656
14657
14658
14659
14660
14661
14662
14663

057220
057220 012700 000762
057224 000004
057226 012737 057250 000004
057234 012737 000340 000006
057242 005737 160000
057246 000461
057250 022626
057252 012737 061144 000004
057260 076600
057262 000100
057264 013701 177766
057270 022701 000020
057274 001401
057276 104033
057300 022700 021200
057304 001401
057306 104033
057310 076600
057312 000102
057314 020027 160000
057320 001403
057322 012701 160000
057326 104020
057330 012737 057352 000004
057336 012737 000340 000006
057344 005767 177741
057350 000420
057352 022626
057354 012737 061144 000004
057362 076600
057364 000100
057366 013701 177766
057372 022701 000100
057376 001401
057400 104024

```

; *THIS TEST CHECKS THAT THE "UNIBUS TIMEOUT" BIT (BIT4)
; *GETS SET IN THE CPU ERROR REGISTER WHEN A TIMEOUT OCCURS.
; *A TIMEOUT TRAP IS FORCED BY REFERENCING BUS ADDRESS 760000.
; *THEN AN ODD ADDRESS ERROR IS FORCED AND IT
; *IS CHECKED IF ONLY BIT (6)-ODD ADDRESS ERROR IS SET
; *(IN CPUERR). THIS CHECKS THAT THE ERROR LOG IS
; *CONTINUOUSLY UPDATED IN THE "LOG CONTINUOUS" MODE.
*****
†ST763:
MOV      #762,R0      ;: SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE
MOV      #15,2#4     ;: CALL THE SCOPE LOOP UTILITY
MOV      #340,2#6    ;: SETUP NEW PC & PSW FOR THE
TST      2#160000    ;: TIMEOUT SERVICE ROUTINE
;: FORCE A TIMEOUT TRAP TO 4 BY
;: REFERENCING NON-EXISTENT ADDRESS
BR       6$
1$:      CMP      (SP)+,(SP)+ ;: RESTORE STACK
MOV      #BERR,2#4   ;: RESTORE OLD PC & PSW FOR TIMEOUT
MED
RDLJAM
MOV      2#CPUERR,R1 ;: SAVE CPU ERR REG
CMP      #BIT4,R1    ;: DID "UNIBUS TIMEOUT" BIT IN CPU ERROR
;: REGISTER GET SET?
BEQ      2$
ERROR   33          ;: BRANCH IF YES
;: *** "UNIBUS TIMEOUT" BIT (BIT4) IN CPU
;: ERROR REG. DID NOT SET WHEN A
;: TIMEOUT WAS FORCED
;: READ THE LOG JAM REGISTER
2$:      CMP      #BIT13+BIT9+BIT7,R0 ;: DID "UNIBUS TIMEOUT" BIT (BIT7) SET?
;: BIT 9= POWER STATUS, ALWAYS SET
BEQ      3$
ERROR   33          ;: BRANCH IF YES
;: *** "UNIBUS TIMEOUT" BIT (BIT7)
;: DID NOT SET IN LOG JAM REGISTER
;: WHEN UNIBUS TIMEOUT WAS FORCED
;: READ LOG PBA
3$:      MED
RDLPBA
CMP      R0,#160000  ;: WAS PHYS BA LOGGED CORRECTLY?
BEQ      5$
MOV      #160000,R1
ERROR   20          ;: PHYSICAL BUS ADDRESS WAS
;: LOGGED WRONG ON A UNIBUS
;: TIMEOUT
5$:      MOV      #45,2#4 ;: SET UP PC,PSW FOR ODD ADDRESS
MOV      #340,2#6
TST      3$+1
;: FORCE ODD ADDRESS ERROR
BR       6$
4$:      CMP      (SP)+,(SP)+ ;: RESTORE STACK
MOV      #BERR,2#4
MED
RDLJAM
MOV      2#CPUERR,R1
CMP      #BIT6,R1    ;: ODD ADDR. BUT SET 3
BEQ      7$
ERROR   24          ;: ODD ADDRESS BIT WAS
;: NOT SET IN THE CPU

```

E07

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS
DQKDA.P11 08-FEB-77 16:17 T763

MACY11 27(1006) 08-FEB-77 16:23 PAGE 266
CHECK UNIBUS TIMEOUT, ODD ADDRESS AND LOG CONTINUOUS MODE

```

14664                                     ;ERROR REGISTER. IN LOG
14665                                     ;CONTINUOUS MADE THE
14666                                     ;LATEST ERROR SHOULD
14667                                     ;BE LOGGED
14668 057402 032700 000004          7$: BIT      #BIT2,RO      ;ODD ADR. BIT SET IN
14669 057406 001001                BNE      6$          ;LOG JAM?
14670 057410 104024                ERROR   24          ;ODD ADDRESS BIT WAS
14671                                     ;NOT SET IN THE LOG
14672                                     ;JAM REGISTER ON A
14673                                     ;ODD ADDRESS ERROR
14674 057412 076600          6$: MED                ;CHECK IF LAST INTERRUPT VECTOR
14675 057414 000104                RDLFGINT          ;WAS LOGGED?
14676 057416 120027 000004        CMPB    RO,#4
14677 057422 001401                BEQ     8$
14678 057424 104036                ERROR   36          ;LAST ERROR VECTOR WS NOT LOGGED
14679
14680 057426          8$:
14681
14682
14683 ::*****
14684 ;*TEST 764      CHECK ILLEGAL INTERNAL ADDRESS TRAP
14685
14686 ;*THIS TEST CHECKS THAT A TRAP OCCURS UPON REFERENCING AN
14687 ;*ILLEGAL INTERNAL ADDRESS AND THAT "ILLEGAL INTERNAL ADDRESS"
14688 ;*BIT (BIT0) OF THE CPU ERROR REGISTER AND BITS OF LOG JAM
14689 ;*REGISTER GET SET. IT ALSO CHECKS IF THE INTERRUPT VECTOR
14690 ;*(4) IS SAVED AS THE "LAST INTERRUPT VECTOR" IN THE LOG
14691 ;*FLAG/INTERRUPT REG.
14692 ::*****
14693 †ST764:
14694 057426 012700 000763          MOV     #763,RO      ;: SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
14695 057432 000004                SCOPE
14696 057434 012737 057464 000004    MOV     #15,2#4     ;: CALL THE SCOPE LOOP UTILITY
14697 057442 012737 000340 000006    MOV     #340,2#6   ;: SETUP NEW HANDLER PC & PSW
14698 057450 005037 177746                CLR     2#CCR
14699 057454 012707 177746                MOV     2#CCR,PC
14700 057460 104034                ERROR   34          ;: ILLEGAL INTERNAL ADDRESS TRAP SHOULD OCCUR
14701                                     ;: *** ILLEGAL INTERNAL ADDRESS
14702 057462 000420                BR      3$          ;: DID NOT RESULT IN A TRAP
14703 057464 022626          1$: CMP     (SP)+,(SP)+ ;: BRANCH TO EXIT IF NO TRAP
14704 057466 012737 061144 000004    MOV     #BERR,2#4  ;: RESTORE STACK
14705 057474 076600                MED
14706 057476 000100                RDLJAM
14707 057500 013701 177766                MOV     2#CPUERR,R1
14708 057504 032701 000001                BIT     #BIT0,R1   ;: DID "ILLEGAL INTERNAL ADDRESS" BIT (0)
14709                                     ;: IN CPU ERROR REGISTER GET SET?
14710 057510 001001                BNE    2$          ;: BRANCH IF YES
14711 057512 104035                ERROR   35          ;: *** ILLEGAL INTERNAL ADDRESS
14712                                     ;: BIT DID NOT SET IN CPU ERROR REG.
14713                                     ;: READ THE LOG JAM REG.
14714 057514 032700 000040          2$: BIT     #BITS,RO ;: DID "ILLEGAL INTERNAL ADDRESS" BIT (5)
14715                                     ;: IN LOG JAM REG. GET SET
14716 057520 001001                BNE    3$          ;: BRANCH IF YES
14717 057522 104035                ERROR   35          ;: *** ILLEGAL INTERNAL ADDRESS BIT
14718                                     ;: DID NOT SET IN LOG JAM REG.
14719          3$:

```

14720
14721
14722
14723
14724
14725
14726
14727
14728
14729
14730
14731
14732
14733
14734 057524
14735 057524 012700 000764
14736 057530 000004
14737
14738 057532 012737 000201 177746
14739 057540 005037 001062
14740 057544 012701 063772
14741 057550 005711
14742 057552 052737 000100 177746
14743 057560 012711 125252
14744 057564 042737 000100 177746
14745 057572 012700 100001
14746 057576 076600
14747 057600 000222
14748 057602 042737 000001 177746
14749 057610 012737 057636 000114
14750 057616 016737 004150 001062
14751 057624 012700 000200
14752 057630 076600
14753 057632 000352
14754 057634 104031
14755
14756
14757
14758
14759
14760 057636 012700 000200 PTRP1:
14761 057642 076600
14762 057644 000352
14763 057646 012737 000001 177746
14764 057654 012737 000116 000114
14765 057662 005037 000116
14766 057666 022626
14767 057670 005737 001062
14768
14769 057674 001401
14770 057676 104041
14771
14772
14773 057700 076600 15:
14774 057702 000101
14775 057704 010004

```

*****
; *TEST 765 CHECK LOG SERVICE & MEMERR LOGS LO-HI BYTE & TAG, IN CACHE ABORT MODE
; TEST CHECKS THAT "LO BYTE PARITY" "HI BYTE PARITY" AND "TAG PARITY"
; BITS CAN SET IN "LOG SERVICE" REGISTERS. IT IS ALSO
; CHECKED THAT THE PROPER TAG AND DATA BITS GET STORED
; IN THE "LOG CACHE DATA," "LOG CACHE TAG/CPU" AND THE
; "MEMORY ADDRESS REGISTER" WHEN A PARITY ERROR IS
; FORCED.
; IT IS CHECKED IF THE INSTRUCTION WAS ABORTED AND THE
; LOG FLAG/INTERRUPT REGISTER LOGGED THE LAST INTERRUPT
; VECTOR.
*****
; *ST765:
MOV #764,RO ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;:CALL THE SCOPE LOOP UTILITY
MOV #DPTRP+PABORT,2#CCR ;:DISABLE PARITY TRAPS (CACHE)
CLR 2#SREG0
MOV #TLOC1,R1 ;:GET POINTER TO TEST LOC.
TST (R1) ;:MAKE IT A HIT
BIS #WMP,2#CCR ;:WRITE WRONG PARITY SET
MOV #125252,(R1) ;:WRITE TEST LOCATION WITH WRONG PARITY
BIC #WMP,2#CCR ;:CLEAR WMP
MOV #BIT15+BIT0,RO
MED ;:ENABLE "LOG FIRST" MODE, AND
WRWHAMI ;:ERROR LOGGING
BIC #DPTRP,2#CCR ;:ENABLE CACHE PARITY TRAPS
MOV #PTRP1,2#114 ;:NEW PARITY TRAP SERVICE
MOV TLOC1,2#SREG0 ;:READ TEST LOC, FORCE PARITY ERROR
MOV #200,RO
MED ;:CLEAN UP THE CACHE
352 ;:INITIALIZATION CODE
ERROR 31 ;:*** CACHE PARITY ERROR TRAP
;:DID NOT OCCUR WHEN
;:TEST LOC WITH BAD PARITY
;:WAS READ
;:ENTER HERE IF PARITY TRAP OCCURRED
PTRP1: MOV #200,RO
MED ;:CLEAN UP THE CACHE
352 ;:INITIALIZATION CODE
MOV #DPTRP,2#CCR ;:DISABLE CACHE PARITY ERROR TRAPS
MOV #116,2#114 ;:REESTABLISH OLD SERVICE VECTORS
CLR 2#116
CMP (SP)+,(SP)+
TST 2#SREG0
;:WAS THE INSTRUCTION ABORTED ON
;:CACHE PARITY ERROR (ABORT MODE)?
;:YES
BEQ 15
ERROR 41 ;:INSTRUCTION HAVING CACHE PARITY
;:ERROR WAS NOT ABORTED, IN THE
;:CACHE ABORT MODE.
;:READ THE "LOG SERVICE" REGISTER
15: MED
RDLSERVICE
MOV RO,R4 ;:COPY

```

14776	057706	042704	177435		BIC	#1C<LO+HI+TAG+BIT1>,R4	;MASK ALL BUT LO,HI,TAG BITS
14777	057712	022704	000342		CMP	#342,R4	;LO,HI ,TAG, CACHE PARITY BITS SET? IN "SERVICE"
14778	057716	001401			BEQ	25	;YES
14779	057720	104042			ERROR	42	*** "LO BYTE" PARITY ERROR
14780							AND "TAG" PARITY ERROR BITS
14781							WERE NOT LOGGED CORRECTLY IN "LOG
14782							"SERVICE" REGISTER, WHEN PARITY
14783							ERROR TRAP WAS FORCED.
14784							CLEAR BITS ARE ACTIVE.
14785	057722	013700	177744	25:	MOV	2#MEMERR,RO	;GET MEM ERR REG
14786	057726	022700	100340		CMP	#HI+LO+TAG+BIT15	;RO :DID"LO BYTE" "HI BYTE" AND "TAG"
14787							PARITY ERROR BITS SET IN
14788							THE MEMORY ERROR REGISTER?
14789	057732	001401			BEQ	35	;YES
14790	057734	104043			ERROR	43	*** "LO BYTE" "HI BYTE" AND "TAG" PARITY
14791							ERROR BITS DID NOT SET
14792							CORRECTLY IN THE MEMORY
14793							ERROR REGISTER
14794	057736	076600		35:	MED		;READ "LOG PBA" REGISTER
14795	057740	000102			RDL PBA		
14796	057742	020027	063772		CMP	RO,#TLOC1	;DID "LOG PBA" CONTAIN CORRECT
14797							PHYSICAL BUS ADDRESS ,WHERE
14798							THE PARITY ERROR OCCURRED?
14799	057746	001403			BEQ	45	;YES
14800	057750	012701	063772		MOV	#TLOC1,R1	;EXPECTED PBA
14801	057754	104020			ERROR	20	*** PHYSICAL BUS ADDRESS
14802							(WHERE PARITY ERROR OCCURRED)
14803							WAS NOT LOGGED CORRECTLY
14804							WHEN CACHE PARITY ERROR WAS FORCED
14805	057756	076600		45:	MED		;READ "LOG CACHE TAG" REGISTER
14806	057760	000107			RDL TAG		
14807	057762	000300			SWAB	RO	
14808	057764	012701	063772		MOV	#TLOC1,R1	;SHIFT RIGHT (3 TIMES) THE 16 BIT
14809	057770	000301			SWAB	R1	
14810	057772	106201			ASRB	R1	;PHYSICAL BUS ADDRESS OF THE
14811	057774	106201			ASRB	R1	;TEST LOCATION
14812	057776	106201			ASRB	R1	
14813	060000	052701	000200		BIS	#BIT7,R1	;FUDGE TAGE BIT
14814	060004	120100			CMPB	R1,RO	;WAS THE CORRECT TAG LOGGED?
14815	060006	001401			BEQ	55	;YES
14816	060010	104017			ERROR	17	;TAG BITS WERE NOT LOGGED
14817							CORRECTLY, WHEN CACHE
14818							PARITY ERROR WAS FORCED
14819	060012	076600		55:	MED		;READ CACHE DATA
14820	060014	000106			RDL DATA		
14821	060016	020027	125252		CMP	RO,#125252	;CACHE DATA LOGGED CORRECTLY?
14822	060022	001403			BEQ	65	
14823	060024	012701	125252		MOV	#125252,R1	;EXPECTED DATA
14824	060030	104016			ERROR	16	
14825							
14826	060032	012700	000001	65:	MOV	#BIT0,RO	;SET UP LOG CONTINUOUS
14827	060036	076600			MED		
14828	060040	000222			WRWHAMI		
14829	060042	012737	060054 000004		MOV	#75,#4	;SETUP CPU VECTOR
14830	060050	005737	160000		TST	2#160000	;FORCE TIMEOUT & TRAP TO 75
14831	060054	022626		75:	CMP	(SP)+,(SP)+	

H07

MAINDEC-11-DGKDA-A KD11-K BASIC LOGIC TESTS
DGKDA.P11 08-FEB-77 16:17 T765

MACY11 27(1006) 08-FEB-77 16:23 PAGE 269
CHECK LOG SERVICE & MEMERR LOGS LO-HI BYTE & TAG, IN CACHE ABORT MODE

14832 060056 012737 061144 000004
14833 060064 076600
14834 060066 000104
14835 060070 120027 000114
14836 060074 001403
14837 060076 010037 001062
14838 060102 104036
14839
14840
14841
14842 060104
14843
14844
14845
14846
14847
14848
14849
14850
14851
14852
14853
14854
14855
14856
14857
14858
14859 060104
14860 060104 012700 000765
14861 060110 000004
14862
14863 060112 012700 100001
14864 060116 076600
14865 060120 000222
14866 060122 012737 060144 000004
14867 060130 012737 000340 000006
14868 060136 005737 160000
14869 060142 000462
14870
14871 060144 022626
14872
14873
14874 060146 012737 060162 000004
14875 060154 005767 177765
14876 060160 000453
14877
14878 060162 022626
14879 060164 012737 061144 000004
14880 060172 076600
14881 060174 000100
14882 060176 013701 177766
14883 060202 022701 000020
14884
14885 060206 001402
14886 060210 104033
14887

MOV #BERR,2#4 ;RESTORE CPU VECTOR
MED ;READ LOG FLAG/INTERRUPT REGISTER
RDLFGINT
CMPB R0,#114 ;DID LO BYTE CONTAIN VECTOR 114?
BEQ B\$
MOV R0,2#\$REGO
ERROR 36 ;LAST INTERRUPT VECTOR WAS NOT
LOGGED CORRECTLY IN FLAG REGISTER
WHEN A CACHE PARITY ERROR WAS
FORCED.

B\$:

;TEST 766 CHECK "LOG FIRST" MODE OF ERROR LOGGING
;THIS TEST CHECKS THE "LOG FIRST" MODE OF ERROR LOGGING.
;THE "LOG FIRST" MODE IS ENABLED. THEN A TIME-OUT TRAP
;IS FORCED, BIT 4 OF CPU ERROR REGISTER SHOULD BE SET.
;THEN AN ODD ADDRESS TRAP IS FORCED. HOWEVER, THIS
;TIME THE ERROR SHOULD NOT BE LOGGED; BIT 6 (ODD
;ADDRESS) SHOULD NOT BE SET BECAUSE THE ERROR LOG
;IS LOCKED UP AFTER THE FIRST ERROR.
;THEN, THE ERROR LOG IS ENABLED (BY SETTING BIT 0 OF
;WHAMI). AN ODD ADDRESS ERROR IS FORCED AGAIN AND IT IS
;CHECKED THAT THIS TIME THE ERROR IS LOGGED. (BIT 6-ODD
;ADDRESS SHOULD BE SET IN CPU ERROR REGISTER).

TST766:

MOV #765,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #BIT15+BIT0,R0 ;SET UP "LOG FIRST MODE"
MED
WRWHAMI
MOV #15,2#4 ;SET UP NEW PC & PSW FOR
MOV #340,2#6 ;TIMEOUT
TST 2#160000 ;FORCE A TIMEOUT
BR 5\$;SKIP TEST IF NO TIMEOUT

1\$:

CMP (SP)+,(SP)+ ;RESTORE STACK
;BIT 4 OF CPU ERROR REGISTER
;SHOULD HAVE SET
MOV #25,2#4 ;SET UP NEW PC FOR ODD ADDRESS
TST 1\$+1 ;FORCE ODD ADDRESS TRAP
BR 5\$;SKIP TEST IF NO ODD ADDRESS TRAP

2\$:

CMP (SP)+,(SP)+ ;RESTORE STACK
MOV #BERR,2#4
MED
RDLJAM
MOV 2#CPUERR,R1
CMP #BIT4,R1 ;"TIMEOUT" BIT SHOULD BE STILL
;SET, CHECK?
BEQ 3\$
ERROR 33 ;*** SECOND ERROR (ODD ADDRESS)
;UPDATED THE ERROR LOG IN

```

14888                                     ; THE LOG FIRST MODE. BIT 4
14889                                     ; (UNIBUS TIMEOUT) SHOULD BE
14890                                     ; STILL SET FROM THE FIRST
14891                                     ; ERROR
14892 060212 000436 100004 3$: BR 5$ ; SKIP THE REST
14893 060214 032700 BIT #BIT2+BIT15,RO ; CHECK THAT ODD ADRES ERROR BITS NOT
14894 060220 001401 BEQ 6$ ; SET IN LOG JAM. NOTE LOG FIRST
14895                                     ; MODE SHOULD INHIBIT FURTHER
14896                                     ; ERROR LOGGING
14897 060222 104037 ERROR 37 ; ODD ADDRESS ERROR BITS GOT SET IN LOG JAM
14898                                     ; THEY SHOULD NOT BE SINCE LOG FIRST MODE
14899                                     ; INHIBITS ERROR LOGGING AFTER THE FIRST ERROR
14900 060224 012700 100001 6$: MOV #BIT15+BIT0,RO ; ENABLE ERROR LOG AGAIN IN
14901                                     ; LOG FIRST MODE
14902 060230 076600 MED
14903 060232 000222 WRWAMI
14904 060234 012737 060256 000004 MOV #4$,2#4 ; SET UP NEW PC & PSW FOR
14905 060242 012737 000340 000006 MOV #340,2#6 ; ODD ADDRESS ERROR
14906 060250 005767 177741 TST 3$+1 ; FORCE ODD ADDRESS TRAP
14907 060254 000415 BR 5$ ; SKIP IF NO TRAP
14908 060256 022626 4$: CMP (SP)+,(SP)+ ; RESTORE STACK
14909                                     ; RESTORE OLD PC(4), PSW(6)
14910 060260 012737 061144 000004 MOV #BERR,2#4
14911 060266 022737 000100 177766 CMP #BIT6,2#CPUERR ; THE ERROR LOG FROM PREVIOUS
14912                                     ; ERROR SHOULD BE OVER WRITTEN.
14913                                     ; ODD ADDRESS BIT SHOULD
14914                                     ; BE SET BECAUSE THE ERROR
14915 060274 001405 BEQ 5$ ; LOG WAS ENABLED.
14916                                     ; OK, IF YES
14917 060276 076600 MED
14918 060300 000100 RDLJAM
14919 060302 013701 177766 MOV 2#CPUERR,R1
14920 060306 104040 ERROR 40 ; THE ERROR LOG WAS NOT UPDATED
14921                                     ; (UPON AN ODD ADDRESS ERROR)
14922                                     ; AFTER THE LOG WAS ENABLED.
14923                                     ; AT THIS FORMAT BIT 6 OF
14924                                     ; CPU ERROR REGISTER SHOULD
14925                                     ; BE SET. IT WAS NOT.
14926 060310 012737 061144 000004 5$: MOV #BERR,2#4 ; RESTORE OLD PC(4), PSW(6)
14927 060316 012700 000001 MOV #BIT0,RO
14928 060322 076600 MED
14929 060324 000222 WRWAMI ; PUT THE LOGGING BACK INTO
14930                                     ; "CONTINUOUS" MODE
14931
14932 ; *****
14933 ; *TEST 767 CHECK LAST INTERRUPT VECTOR IS LOGGED IN FLAG REG.
14934 ; *****
14935 060326 012700 000766 TST767: MOV #766,RO ; SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
14936 060332 000004 SCOPE ; CALL THE SCOPE LOOP UTILITY
14937
14938 060334 012737 060344 000030 MOV #1$,2#30 ; LOAD EMT VECTOR WITH 1$
14939 060342 104000 EMT ; FIRST INTERRUPT -- EMT
14940 060344 022626 1$: CMP (SP)+,(SP)+ ; CLEAN UP STACK
14941 060346 012737 061544 000030 MOV #ERROR,2#30 ; RESTORE VECTOR
14942 060354 012737 060366 000004 MOV #2$,2#4 ; SET UP CPU VECTOR
14943 060362 005737 160000 TST 2#160000 ; FORCE TIMEOUT

```

```

14944 060366 022626          2$:  CMP      (SP)+,(SP)+      ;CLEAN UP STACK
14945 060370 012737 061144 000004  MOV      #BERR,#4         ;RESTORE BUS ERROR VECTOR
14946 060376 076600          MED          ;CHECK FLAG
14947 060400 000104          RDLFGINT
14948 060402 120027 000030  CMPB     RO,#30          ;EMT VECTOR LAST LOGGED?
14949 060406 001401          BEQ      3$              ;BR IF YES
14950 060410 104036          ERROR     36            ;LOG FLAG/INT REG DID NOT LOG VECTOR
14951                                     ;LO BYTE OF LOG FLAG/INT REG S/B=30
14952
14953 060412 012737 060422 000020  3$:  MOV      #4$,#20        ;LOAD IOT VECTOR WITH 4$
14954 060420 000004          IOT          ;SECOND INTERRUPT-SHOULD LOAD LOG FLAG REG
14955 060422 022626          4$:  CMP      (SP)+,(SP)+      ;CLEANUP STACK
14956 060424 012737 061204 000020  MOV      #SCOPE,#20      ;RESTORE IOT VECTOR
14957 060432 012737 060444 000004  MOV      #5$,#4         ;SET UP CPU VECTOR
14958 060440 005737 160000          TST      @#160000      ;FORCE TIMEOUT
14959 060444 022626          5$:  CMP      (SP)+,(SP)+      ;CLEAN UP STACK
14960 060446 012737 061144 000004  MOV      #BERR,#4         ;RESTORE BUS ERROR VECTOR
14961 060454 076600          MED          ;CHECK FLAG
14962 060456 000104          RDLFGINT
14963 060460 120027 000020  CMPB     RO,#20          ;IOT VECTOR LAST LOGGED?
14964 060464 001401          BEQ      6$              ;BR IF YES
14965 060466 104036          ERROR     36            ;LOG FLAG/INT REG DID NOT LOG VECTOR
14966                                     ;LOW BYTE S/B = 20
14967
14968 060470 012700 000767  6$:  MOV      #STN-1,RO      ;SET UP FOR MISSED TEST CHECK AND
14969                                     ;FULL WORD TEST NUMBER FOR APT
14970
14971                                     .ENABLE AMA
14972
14973                                     .SBTTL END OF PASS ROUTINE
14974
14975                                     ;*****
14976                                     ;*INCREMENT THE PASS NUMBER ($PASS)
14977                                     ;*IF THERES A MONITOR GO TO IT
14978                                     ;*IF THERE ISN'T JUMP TO INIT
14979
14980 SEOP:
14981 060474 000004          SCOPE
14982 060474 005037 001110  CLR      $TIMES          ;; ZERO THE NUMBER OF ITERATIONS
14983 060476 005237 001126  INC      $PASS           ;; INCREMENT THE PASS NUMBER
14984 060502 042737 100000 001126  BIC      #100000,$PASS   ;; DON'T ALLOW A NEG. NUMBER
14985 060506 005327          DEC      (PC)+          ;; LOOP?
14986 060514 000001          SEOPCT: .WORD 1
14987 060516 003027          BGT      $DOAGN          ;; YES
14988 060520 012737          MOV      (PC)+,@(PC)+   ;; RESTORE COUNTER
14989 060522 000001          SENDCT: .WORD 1
14990 060524 065027          TYPE    EOP1            ;TYPE "END PASS #"
14991 060526 001126          MOV      $PASS,-(SP)    ;SAVE $PASS FOR TYP0UT
14992 060530 104401 065045          TYPOC   EOP2            ;TYPE PASS NUMBER IN OCTAL
14993 060534 013746 001012          TYPE    "ERROR COUNT =" ;TYPE "ERROR COUNT ="
14994 060540 104401 001012          MOV      $ERTTL,-(SP)   ;SAVE ERROR TOTAL FOR TYP0UT
14995 060542 013746 001012          TYPOC   $CRLF           ;TYPE ERROR TOTAL
14996 060546 104401 001115          TYPE    #42,RO          ;; GET MONITOR ADDRESS
14997 060552 104401 000042          SGET42: MOV
14998 060554 104401 000042          MOV
14999 060560 013700 000042          MOV

```



```

15011 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
15012 .SBTTL / / / / / UTILITIES / / / / /
15013 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
15014
15015 .SBTTL POWER DOWN AND UP ROUTINES
15016
15017 ;*****
15018 :POWER DOWN ROUTINE
15019 060610 012737 060762 000024 $PWRDN: MOV $SILLUP, @PWRVEC ;;SET FOR FAST UP
15020 060616 012737 000340 000026 MOV @340, @PWRVEC+2 ;;PRIO:7
15021 060624 010046 MOV R0, -(SP) ;;PUSH R0 ON STACK
15022 060626 010146 MOV R1, -(SP) ;;PUSH R1 ON STACK
15023 060630 010246 MOV R2, -(SP) ;;PUSH R2 ON STACK
15024 060632 010346 MOV R3, -(SP) ;;PUSH R3 ON STACK
15025 060634 010446 MOV R4, -(SP) ;;PUSH R4 ON STACK
15026 060636 010546 MOV R5, -(SP) ;;PUSH R5 ON STACK
15027 060640 017746 120174 MOV @SWR, -(SP) ;;PUSH @SWR ON STACK
15028 060644 010637 060766 MOV SP, $SAVR6 ;;SAVE SP
15029 060650 012737 060662 000024 MOV @SPWRUP, @PWRVEC ;;SET UP VECTOR
15030 060656 000000 HALT
15031 060660 000776 BR .-2 ;;HANG UP
15032
15033 ;*****
15034 :POWER UP ROUTINE
15035 060662 012737 060762 000024 $PWRUP: MOV $SILLUP, @PWRVEC ;;SET FOR FAST DOWN
15036 060670 013706 060766 MOV $SAVR6, SP ;;GET SP
15037 060674 005037 060766 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
15038 060700 005237 060766 1$: INC $SAVR6 ;;WAIT FOR THE INC
15039 060704 001375 BNE 1$ ;;OF WORD
15040 060706 011600 MOV (SP), R0 ;;GET OLD SWR VALUE
15041 060710 076600 MED ;;WRITE BACK ORIGINAL SWR VALUE
15042 060712 000226 WCNSSW ;;INTO HARDWARE SWITCH REGISTER
15043 060714 012677 120120 MOV (SP)+, @SWR ;;POP STACK INTO @SWR
15044 060720 012605 MOV (SP)+, R5 ;;POP STACK INTO R5
15045 060722 012604 MOV (SP)+, R4 ;;POP STACK INTO R4
15046 060724 012603 MOV (SP)+, R3 ;;POP STACK INTO R3
15047 060726 012602 MOV (SP)+, R2 ;;POP STACK INTO R2
15048 060730 012601 MOV (SP)+, R1 ;;POP STACK INTO R1
15049 060732 012600 MOV (SP)+, R0 ;;POP STACK INTO R0
15050 060734 012737 060610 000024 MOV @SPWRDN, @PWRVEC ;;SET UP THE POWER DOWN VECTOR
15051 060742 012737 000340 000026 MOV @340, @PWRVEC+2 ;;PRIO:7
15052 060750 104401 TYPE ;;REPORT THE POWER FAILURE
15053 060752 060770 $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
15054 060754 012716 MOV (PC)+, (SP) ;;RESTART AT PWRUP
15055 060756 061000 $PWRAD: .WORD PWRUP ;;RESTART ADDRESS
15056 060760 000002 RTI
15057 060762 000000 $SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
15058 060764 000776 BR .-2 ;;BEFORE THE POWER DOWN WAS COMPLETE
15059 060766 000000 $SAVR6: 0 ;;PUT THE SP HERE
15060 060770 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
15061 060776 000122 .EVEN
15062
15063
15064 061000 012706 001000 PWRUP: MOV @STACK, SP ;;RESET SP
15065 061004 005037 177776 CLR @PSW ;;PRIORITY 0 -- CLEAR CODES
15066 061010 000137 001630 JMP @START ;;RESTART PROGRAM

```

```

15067
15068 ; *****
15069 ; .SBTTL "T" BIT SERVICE ROUTINE
15070 ; *****
15071
15072 061014 062716 000002          TBSER: ADD    #2,(SP)          ;MOVE RETURN PC AROUND ERROR CALL
15073 061020 042766 000020 000002  BIC    #20,2(SP)         ;TURN OFF THE "T" BIT
15074 061026 000006                RTT                          ;RETURN TO THE CALLING TEST
15075
15076 ;.SBTTL MICROBREAK TRAP SERVICE ROUTINE
15077 ; *****
15078 ; THIS ROUTINE MERELY SETS A FLAG
15079 ; WHEN THE ROUTINE HAS BEEN ENTERED
15080 ; *
15081 061030 005237 061036          BKROUT: INC    BKFLAG          ;SET MICROBREAK FLAG TO
15082 ;INDICATE TRAP TO 4 OCCURRED
15083 061034 000002                RTI                          ;RETURN FROM TRAP
15084 061036 000000          BKFLAG: .WORD    0          ;MICROBREAK TRAP FLAG
15085
15086 ; *****
15087 ;.SBTTL RSVD INSTRUCTION TRAP SERVICE ROUTINE
15088 ; *****
15089
15090 ;THIS ROUTINE SERVICES UNEXPECTED RESERVED INSTRUCTION TRAP ERRORS
15091 ;IT RESULTS IN PRINTING THE ERROR MESSAGE: "TRAPPED TO 10 PC=XXXXXX"
15092 ;WHERE XXXXXX IS THE ADDRESS CONTAINING THE INSTRUCTION WORD THAT
15093 ;SPRUNG THE TRAP. AFTER PRINTING THE ERROR MESSAGE AN ATTEMPT IS
15094 ;MADE TO RESTART THE PROGRAM AT THE BEGINNING.
15095
15096 ;IF THE TRAP IS SPRUNG WHILE IN THE PROCESS OF TRYING TO SERVICE A
15097 ;PREVIOUS RSVD INSTRUCTION TRAP OR AN UNEXPECTED BUS ERROR THE PROGRAM
15098 ;WILL HALT. AFTER THE HALT THE STACK WILL CONTAIN INFORMATION RELATIVE
15099 ;TO THE TWO SUCCESSIVE TRAPS AS SHOWN BELOW:
15100
15101 ;[SP] PC+2 OF 2ND TRAP
15102 ;[SP]+2 PSW
15103 ;[SP]+4 PC+2 OF 1ST TRAP
15104 ;[SP]+6 PSW
15105
15106 ;LOCATION "CATERR" CAN BE EXAMINED TO OBTAIN THE FOLLOWING
15107 ;INFORMATION:
15108
15109 ;[CATERR]=401 RSVD INSTR TRAP COMBINED WITH A BUS ERROR
15110 ;TRAP (PC AT TIME OF ERROR HALT INDICATES
15111 ;WHICH OCCURRED FIRST)
15112 ;[CATERR]=2 TWO SUCCESSIVE BUS ERROR TRAPS
15113 ;[CATERR]=1000 TWO SUCCESSIVE RSVD INSTR TRAPS
15114
15115 ;THE CONTENTS OF RD AT THE TIME OF THE
15116 ;HALT PROVIDES FURTHER INFORMATION AS TO THE LAST TEST BEING EXECUTED
15117 ;WHEN THE TRAPS OCCURRED.
15118
15119 ;THESE TWO INSTRUCTIONS ARE USED BY THE BASIC INSTRUCTION
15120 ;TESTS TO VERIFY THE RSVD INSTR TRAP MECHANISM PRIOR TO ACTIVATING THE SERVICE
15121 ;ROUTINE
15122

```

```

15123 061040 005137 063172      RSVTST: COM      RSVFLG      ;SET RSVD INSTR TRAP TEST FLAG
15124 061044 000002              RTI              ;RETURN TO BASIC TEST
15125
15126 061046 005737 063176      RSERR:  TST      @#CATERR    ;ANY PENDING CATASTROPHIC ERRORS
15127 061052 001025              BNE      INCRSV           ;BE IF YES
15128 061054 105237 063177      INCB     @#1+CATERR        ;SET RSVD INSTR FLAG
15129 061060 032777 010000 117752  BIT      @SW12,@SWR        ;INHIBIT ERROR PRINT ?
15130 061066 001015              BNE      RESTAR          ;BR IF YES
15131 061070 104401              TYPE     RESTAR           ;GO TYPE "TRAPPED TO 10 PC="
15132 061072 065167              RSMMSG
15133 061074 011646      RSBERT: MOV      (SP),-(SP)  ;GET ERROR PC ON STACK FOR PRINTING
15134 061076 104402              TYPOC
15135 061100 104401              TYPE
15136 061102 001115              $CRLF
15137 061104 005237 001012      INC      @#SERTTL         ;COUNT THE ERROR
15138 061110 032777 100000 117722  BIT      @BIT15,@SWR        ;HALT ON ERROR?
15139 061116 001401              BEQ      RESTAR          ;BR IF NOT
15140 061120 000000              HALT
15141 061122 000137 003236      RESTAR: JMP      @#INIT     ;GO ATTEMPT RESTART
15142 061126 105237 063177      INCRSV: INCB     @#1+CATERR  ;INCREMENT RSVD INSTR FLAG
15143 061132 000000              HALT
15144 061134 000772              BR       RESTAR          ;CATASTROPHIC ERROR HALT
15145
15146
15147
15148
15149
15150
15151
15152
15153
15154
15155
15156
15157
15158
15159
15160
15161
15162
15163
15164
15165
15166
15167
15168
15169
15170
15171
15172
15173
15174
15175
15176
15177
15178
    
```

```

; *****
; .SBTTL  BUS ERROR TRAP SERVICE ROUTINE
; *****
    
```

```

; THIS ROUTINE SERVICES UNEXPECTED BUS ERROR TRAPS (BUS TIMEOUT, ODD ADDRESS
; ERRORS, STACK OVERFLOW, AND ILLEGAL INSTRUCTIONS). IT RESULTS IN PRINTING THE
; ERROR MESSAGE: "TRAPPED TO 4 PC =XXXXXX" WHERE XXXXXX IS THE
; CONTENTS OF THE PC WHEN THE TRAP WAS SPRUNG. AFTER PRINTING THE
; ERROR MESSAGE AN ATTEMPT IS MADE TO RESTART THE PROGRAM AT
; THE BEGINNING.
    
```

```

; IF THE TRAP IS SPRUNG WHILE IN THE PROCESS OF TRYING TO SERVICE A PREVIOUS
; RSVD INSTR TRAP OR A PREVIOUS BUS ERROR, THE PROGRAM WILL HALT.
; AFTER THE HALT THE STACK WILL CONTAIN INFORMATION RELATIVE TO THE
; TWO SUCCESSIVE TRAPS AS SHOWN BELOW:
    
```

```

;[SP] PC+2    OF 2ND TRAP
;[SP]+2 PSW
;[SP]+4 PC+2    OF 1ST TRAP
;[SP]+6 PSW
    
```

```

; LOCATION "CATERR" CAN BE EXAMINED TO OBTAIN THE FOLLOWING
; INFORMATION:
    
```

```

;[CATERR]=401  RSVD INSTR TRAP COMBINED WITH A BUS ERROR
;              TRAP (PC AT TIME OF ERROR HALT
;              INDICATES WHICH OCCURRED FIRST)
;[CATERR]=2    TWO SUCCESSIVE BUS ERRORS
;[CATERR]=1000 TWO SUCCESSIVE RSVD INSTR TRAPS
    
```

```

; THE CONTENTS OF RO AT THE TIME OF
; THE HALT PROVIDED FURTHER INFORMATION AS TO THE TEST IN PROGRESS
    
```

15179
15180
15181
15182
15183
15184
15185
15186
15187
15188
15189
15190
15191
15192
15193
15194
15195
15196
15197
15198
15199
15200
15201
15202
15203
15204
15205
15206
15207
15208
15209
15210
15211
15212
15213
15214
15215
15216
15217
15218
15219
15220
15221
15222
15223
15224
15225
15226
15227
15228
15229
15230
15231
15232
15233
15234

061136 005137 063174
061142 000002
061144 005737 063176
061150 001011
061152 105237 063176
061156 032777 010000 117654
061164 001356
061166 104401
061170 065142
061172 000740
061174 105237 063176
061200 000000
061202 000747
061204
061210 001406
061212 012737 061222 001112
061220 104011
061222 005037 001112
061226 110037 001002
061232 032777 002000 117600
061240 001411
061242 017737 117572 063166
061250 042737 177000 063166
061256 020037 063166
061262 001510

; WHEN THE TRAPS OCCURRED.
; THE CONTENTS OF THE SP CAN BE USED TO INDICATE IF STACK OVERFLOW CAUSED
; THE BUSS ERROR TRAP(S) AS SHOWN BELOW:
; 400>[SP]>336 YELLOW ZONE
; [SP]=0 RED ZONE
; THESE TWO INSTRUCTIONS ARE USED BY THE BASIC INSTRUCTION TESTS TO
; VERIFY THAT THE BUS ERROR TRAP MECHANISM WORKS PRIOR TO ACTIVATING
; THE SERVICE ROUTINE
BETST: COM BERFLG ; SET BUS ERROR TRAP TEST FLAG
RTI ; RETURN TO BASIC TEST
BERR: TST @CATERR ; ANY CATASTROPHIC ERRORS PENDING?
BNE 2\$; BR IF YES
INCB @CATERR ; SET CATASTROPHIC ERROR FLAG
BIT @SW12,@SWR ; INHIBIT ERROR PRINT
BNE RESTAR ; BR IF YES
TYPE ; PRINT "TRAP TO 4" MESSAGE
BR RSBERT ; TYPE REST OF BUS ERROR MESSAGE
2\$: INCB @CATERR ; SET CATASTROPHIC ERROR FLAG
HALT ; CATASTROPHIC ERROR HALT-SCHOOLS OUT
BR RESTAR ; DEPRESS CONTINUE TO ATTEMPT RESTART

.SBTTL SCOPE HANDLER ROUTINE

; THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
; AND LOAD THE TEST NUMBER(\$TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
; AND LOAD THE ERROR FLAG (\$ERFLG) INTO DISPLAY<15:08>
; THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
; *SW14=1 LOOP ON TEST
; *SW11=1 INHIBIT ITERATIONS
; *SW09=1 LOOP ON ERROR
; *CALL
; * SCOPE ;; SCOPE=IOT

SSCOPE: CMP RO,@\$TSTNM ; ANY MISSED TESTS ?
BEQ 10\$; BR IF NOT
MOV @12\$,@SESCAPE ; NO ERROR LOOPING
ERROR 11 ; MISSED TESTS ERROR CALL
12\$: CLR @SESCAPE ; NORMAL ERROR LOOPING
10\$: MOVB RO,@\$TSTNM ; INSURE TSTNUM IS CORRECT
BIT @SW10,@SWR ; LOOP ON SELECTED TEST?
BEQ 11\$; BR IF NO
MOV @SWR,@SELTST ; GET CONTENTS OF SWITCHES
BIC @177000,@SELTST ; MASK OUT SWR<15:9>
CMP RO,@SELTST ; IS THIS THE SELECTED TEST?
BEQ SOVER ; BR IF YES

```

15235 061264 11S:
15236 061264 032777 040000 117546 1S: BIT #BIT14,SWR ;; LOOP ON PRESENT TEST?
15237 061272 001104 BNE $OVER ;; YES IF SW14=1
15238 :####START OF CODE FOR THE XOR TESTER####
15239 061274 000416 $XTSTR: BR 6S ;; IF RUNNING ON THE "XOR" TESTER CHANGE
15240 ;; THIS INSTRUCTION TO A "NOP" (NOP=240)
15241 061276 013746 000004 MOV 2#ERRVEC, -(SP) ;; SAVE THE CONTENTS OF THE ERROR VECTOR
15242 061302 012737 061322 000004 MOV 8SS, 2#ERRVEC ;; SET FOR TIMEOUT
15243 061310 005737 177060 TST 2#177060 ;; TIME OUT ON XOR?
15244 061314 012637 000004 MOV (SP)+, 2#ERRVEC ;; RESTORE THE ERROR VECTOR
15245 061320 000453 BR $SVLAD ;; GO TO THE NEXT TEST
15246 061322 022626 5S: CMP (SP)+, (SP)+ ;; CLEAR THE STACK AFTER A TIME OUT
15247 061324 012637 000004 MOV (SP)+, 2#ERRVEC ;; RESTORE THE ERROR VECTOR
15248 061330 000413 BR 7S ;; LOOP ON THE PRESENT TEST
15249 061332 6S: ;####END OF CODE FOR THE XOR TESTER####
15250 061332 105737 001003 2S: TSTB SERFLG ;; HAS AN ERROR OCCURRED?
15251 061336 001421 BEQ 3S ;; BR IF NO
15252 061340 123737 001015 001003 CMPB SERMAX, SERFLG ;; MAX. ERRORS FOR THIS TEST OCCURRED?
15253 061346 101015 BHI 3S ;; BR IF NO
15254 061350 032777 001000 117462 BIT #BIT09,SWR ;; LOOP ON ERROR?
15255 061356 001404 BEQ 4S ;; BR IF NO
15256 061360 013737 001010 001006 7S: MOV $LPERR, $LPADR ;; SET LOOP ADDRESS TO LAST SCOPE
15257 061366 000446 BR $OVER
15258 061370 105037 001003 4S: CLRB SERFLG ;; ZERO THE ERROR FLAG
15259 061374 005037 001110 CLR $TIMES ;; CLEAR THE NUMBER OF ITERATIONS TO MAKE
15260 061400 000415 BR 1S ;; ESCAPE TO THE NEXT TEST
15261 061402 032777 004000 117430 3S: BIT #BIT11,SWR ;; INHIBIT ITERATIONS?
15262 061410 001011 BNE 1S ;; BR IF YES
15263 061412 005737 001126 TST $PASS ;; IF FIRST PASS OF PROGRAM
15264 061416 001406 BEQ 1S ;; INHIBIT ITERATIONS
15265 061420 005237 001004 INC $ICNT ;; INCREMENT ITERATION COUNT
15266 061424 023737 001110 001004 CMP $TIMES, $ICNT ;; CHECK THE NUMBER OF ITERATIONS MADE
15267 061432 002024 BGE $OVER ;; BR IF MORE ITERATION REQUIRED
15268 061434 012737 000001 001004 1S: MOV #1, $ICNT ;; REINITIALIZE THE ITERATION COUNTER
15269 061442 013737 061534 001110 MOV $MXCNT, $TIMES ;; SET NUMBER OF ITERATIONS TO DO
15270 061450 105237 001002 $SVLAD: INCB $STSTNM ;; COUNT TEST NUMBERS
15271 061454 113737 001002 001124 MOVB $STSTNM, $TESTN ;; SET TEST NUMBER IN APT MAILBOX
15272 061462 011637 001006 MOV (SP), $LPADR ;; SAVE SCOPE LOOP ADDRESS
15273 061466 011637 001010 MOV (SP), $LPERR ;; SAVE ERROR LOOP ADDRESS
15274 061472 005037 001112 CLR $ESCAPE ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
15275 061476 112737 000001 001015 MOVB #1, SERMAX ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
15276 061504 013777 001002 117330 $OVER: MOV $STSTNM, $DISPLAY ;; DISPLAY TEST NUMBER
15277 061512 013716 001006 MOV $LPADR, (SP) ;; FUDGE RETURN ADDRESS
15278 061516 120037 001002 CMPB RO, 2#$STSTNM ;; WAS $STSTNM INCREMENTED?
15279 061522 001401 BEQ 10S ;; BR IF NOT
15280 061524 005200 INC RO ;; INCREMENT TEST NUMBER
15281 061526 010037 001124 10S: MOV RO, 2#$TESTN ;; FIX $TESTN TO BE WORD COUNT, NOT BYTE
15282 061532 000002 RTI
15283 061534 000200 $MXCNT: 200 ;; MAX. NUMBER OF ITERATIONS
15284
15285 061536 005137 063170 SCOPEA: COM 2#SCOFLG ;; THESE TWO INSTRUCTIONS ARE
15286 061542 000002 RTI ;; USED IN THE BASIC TESTS TO
15287 ;; VERIFY THE IOT LINKAGE
15288
15289
15290

```

.SBTTL ERROR HANDLER ROUTINE

```

15291
15292
15293
15294
15295
15296
15297
15298
15299
15300
15301
15302
15303
15304 061544
15305 061544 010546
15306 061546 012705 001060
15307 061552 016625 000004
15308 061556 010025
15309 061560 010125
15310 061562 010225
15311 061564 010325
15312 061566 010425
15313 061570 022715 177777
15314 061574 001001
15315 061576 010615
15316 061600 012605
15317 061602 105237 001003
15318 061606 001775
15319 061610 013777 001002 117224
15320 061616 005237 001012
15321 061622 011637 001016
15322 061626 162737 000002 001016
15323 061634 117737 117156 001014
15324 061642 032777 020000 117170
15325 061650 001004
15326 061652 004737 061772
15327 061656 104401 001115
15328 061662
15329 061662 122737 000001 001140
15330 061670 001007
15331 061672 113737 001014 061704
15332 061700 004737 062662
15333 061704 000
15334 061705 000
15335 061706 000777
15336 061710 005777 117124
15337 061714 100001
15338 061716 000000
15339 061720 032777 001000 117112
15340 061726 001402
15341 061730 013716 001010
15342 061734 005737 001112
15343 061740 001402
15344 061742 013716 001112
15345 061746
15346 061746 012737 177777 001074

```

```

*****
THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
AND GO TO SERRTYP ON ERROR
THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
SW15=1 HALT ON ERROR
SW13=1 INHIBIT ERROR TYPEOUTS
SW09=1 LOOP ON ERROR
CALL
* ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER

SERROR:
MOV R5, -(SP) ;SAVE R5 ON STACK
MOV #SREGAD, R5 ;GET POINTER
MOV 4(SP), (R5)+ ;SAVE ERROR PSW IN $REGAD FOR TYP0UT
MOV R0, (R5)+ ;SAVE R0 FOR TYPEOUTS
MOV R1, (R5)+ ;SAVE R1 IN $REG1
MOV R2, (R5)+ ;SAVE R2 IN $REG2, ETC.
MOV R3, (R5)+
MOV R4, (R5)+
CMP #-1, (R5) ;IS SP ALREADY STORED IN $REG5?
BNE 10$ ;BR IF YES
MOV SP, (R5) ;PUT SP IN $REG5 FOR TYP0UT
MOV (SP)+, R5 ;RESTORE R5
10$: INCB $ERFLG ;SET THE ERROR FLAG
7$: BEQ 7$ ;DON'T LET THE FLAG GO TO ZERO
MOV $STNM, @DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
INC $ERTTL ;INC THE ERROR COUNT
MOV (SP), $ERRPC ;GET ADDRESS OF ERROR INSTRUCTION
SUB #2, $ERRPC
MOVVB @ERRPC, $ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
BIT #BIT13, @SWR ;SKIP TYPEOUT IF SET
BNE 20$ ;SKIP TYPEOUTS
JSR PC, SERRTYP ;GO TO USER ERROR ROUTINE
TYPE , $CRLF

20$: CMPB #APTENV, $ENV ;RUNNING IN APT MODE
BNE 2$ ;NO SKIP APT ERROR REPORT
MOVVB $ITEMB, 21$ ;SET ITEM NUMBER AS ERROR NUMBER
JSR PC, $ATY4 ;REPORT FATAL ERROR TO APT

21$: .BYTE 0
.BYTE 0
22$: BR 22$ ;APT ERROR LOOP
2$: TST @SWR ;HALT ON ERROR
BPL 3$ ;SKIP IF CONTINUE
HALT ;HALT ON ERROR!
3$: BIT #BIT09, @SWR ;LOOP ON ERROR SWITCH SET?
BEQ 4$ ;BR IF NO
MOV $LPERR, (SP) ;FUDGE RETURN FOR LOOPING
4$: TST $ESCAPE ;CHECK FOR AN ESCAPE ADDRESS
BEQ 5$ ;BR IF NONE
MOV $ESCAPE, (SP) ;FUDGE RETURN ADDRESS FOR ESCAPE
5$: MOV #-1, @SREG5 ;FLAG CURRENT STACK POINTER TO BE TYPED

```

```

15347 061754 042766 000020 000002      BIC      #20,2(SP)      ;CLEAR T BIT IN CASE ERROR OCCURED
15348                                     ;IN T BIT TESTS
15349 061762 000002                                     RTI
15351 061764 005137 063164      ERRA:    COM      @#ERRFLG      ;THESE TWO INSTRUCTIONS ARE USED
15352 061770 000002                                     RTI      ;IN THE BASIC TESTS TO VERIFY THE EMT
    
```

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE

```

;*****
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
    
```

\$ERRTYP:

```

15361 061772                                     TYPE      $SCLF      ;;"CARRIAGE RETURN" & "LINE FEED"
15362 061772 104401 001115      MOV      RO,-(SP)      ;SAVE RO
15363 061776 010046                                     CLR      RO      ;PICKUP THE ITEM INDEX
15364 062000 005000                                     BISB     @#$ITEMB,RO
15365 062002 153700 001014      BNE     1$      ;IF ITEM NUMBER IS ZERO, JUST
15366 062006 001004                                     ;TYPE THE PC OF THE ERROR
15367                                     ;SAVE $ERRPC FOR TYPEOUT
15368 062010 013746 001016      MOV      $ERRPC,-(SP) ;ERROR ADDRESS
15369                                     ;GO TYPE--OCTAL ASCII(ALL DIGITS)
15370 062014 104402                                     TYPOC
15371 062016 000426      BR      6$      ;GET OUT
15372 062020 005300      1$:    DEC      RO      ;ADJUST THE INDEX SO THAT IT WILL
15373 062022 006300      ASL     RO      ;WORK FOR THE ERROR TABLE
15374 062024 006300      ASL     RO
15375 062026 006300      ASL     RO
15376 062030 062700 001150      ADD     @#$ERRTB,RO   ;FORM TABLE POINTER
15377 062034 012037 062044      MOV     (RO)+,2$     ;PICKUP "ERROR MESSAGE" POINTER
15378 062040 001404      BEQ     3$      ;SKIP TYPEOUT IF NO POINTER
15379 062042 104401      TYPE   ;TYPE THE "ERROR MESSAGE"
15380 062044 000000      2$:    .WORD 0      ;"ERROR MESSAGE" POINTER GOES HERE
15381 062046 104401 001115      TYPE   $SCLF      ;;"CARRIAGE RETURN" & "LINE FEED"
15382 062052 012037 062062      3$:    MOV     (RO)+,4$ ;PICKUP "DATA HEADER" POINTER
15383 062056 001404      BEQ     5$      ;SKIP TYPEOUT IF 0
15384 062060 104401      TYPE   ;TYPE THE "DATA HEADER"
15385 062062 000000      4$:    .WORD 0      ;"DATA HEADER" POINTER GOES HERE
15386 062064 104401 001115      TYPE   $SCLF      ;;"CARRIAGE RETURN" & "LINE FEED"
15387 062070 011000      5$:    MOV     (RO),RO  ;PICKUP "DATA TABLE" POINTER
15388 062072 001004      BNE     7$      ;GO TYPE THE DATA
15389 062074 012600      6$:    MOV     (SP)+,RO  ;RESTORE RO
15390 062076 104401 001115      TYPE   $SCLF      ;;"CARRIAGE RETURN" & "LINE FEED"
15391 062102 000207      RTS     PC      ;RETURN
15392 062104      7$:
15393 062104 013046      MOV     @2(RO)+,-(SP) ;SAVE @2(RO)+ FOR TYPEOUT
15394 062106 104402      TYPOC ;GO TYPE--OCTAL ASCII(ALL DIGITS)
15395 062110 005710      TST    (RO)      ;IS THERE ANOTHER NUMBER?
15396 062112 001770      BEQ     6$      ;BR IF NO
15397 062114 104401 062122      TYPE   8$      ;TYPE TWO(2) SPACES
15398 062120 000771      BR      7$      ;LOOP
15399 062122 020040 000      8$:    .ASCIZ / /      ;TWO(2) SPACES
15400      .EVEN
    
```

; *****

15403
 15404
 15405
 15406 062126 005137 063162
 15407 062132 000002
 15408
 15409
 15410
 15411
 15412
 15413
 15414
 15415
 15416
 15417
 15418
 15419
 15420
 15421
 15422
 15423
 15424
 15425
 15426
 15427 062134 105737 001057
 15428 062140 100002
 15429 062142 000000
 15430 062144 000430
 15431 062146 010046
 15432 062150 017600 000002
 15433 062154 122737 000001 001140
 15434 062162 001011
 15435 062164 132737 000100 001141
 15436 062172 001405
 15437 062174 010037 062204
 15438 062200 004737 062652
 15439 062204 000000
 15440 062206 132737 000040 001141
 15441 062214 001003
 15442 062216 112046
 15443 062220 001005
 15444 062222 005726
 15445 062224 012600
 15446 062226 062716 000002
 15447 062232 000002
 15448 062234 122716 000011
 15449 062240 001430
 15450 062242 122716 000200
 15451 062246 001006
 15452 062250 005726
 15453 062252 104401
 15454 062254 001115
 15455 062256 105037 062412
 15456 062262 000755
 15457 062264 004737 062346
 15458 062270 123726 001056

.SBTTL PRINT ROUTINES
 ; *****
 PRINA: COM 2#PRIFLG ; THESE TWO INSTRUCTIONS ARE
 RTI ; USED BY THE BASIC TESTS TO VERIFY
 ; THE TRAP INSTRUCTION

.SBTTL TYPE ROUTINE
 ; *****
 ; *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 ; *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 ; *NOTE1: SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 ; *NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 ; *NOTE3: SFILLC CONTAINS THE CHARACTER TO FILL AFTER.

*CALL:
 ; *1) USING A TRAP INSTRUCTION
 ; * TYPE ,MESADR ; ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 ; *OR
 ; * TYPE
 ; * MESADR
 ; *

\$TYPE: TSTB \$TPFLG ; IS THERE A TERMINAL?
 BPL 1\$; BR IF YES
 HALT ; HALT HERE IF NO TERMINAL
 BR 3\$; LEAVE
 1\$: MOV RO, -(SP) ; SAVE RO
 MOV 22(SP), RO ; GET ADDRESS OF ASCIZ STRING
 CMPB #APTENV, \$ENV ; RUNNING IN APT MODE
 BNE 62\$; NO GO CHECK FOR APT CONSOLE
 BITB #APTSPool, \$ENV ; SPOOL MESSAGE TO APT
 BEQ 62\$; NO GO CHECK FOR CONSOLE
 MOV RO, 61\$; SETUP MESSAGE ADDRESS FOR APT
 JSR PC, \$ATY3 ; SPOOL MESSAGE TO APT
 61\$: .WORD 0 ; MESSAGE ADDRESS
 62\$: BITB #APTCSUP, \$ENV ; APT CONSOLE SUPPRESSED
 BNE 60\$; YES, SKIP TYPE OUT
 2\$: MOVB (RO)+, -(SP) ; PUSH CHARACTER TO BE TYPED ONTO STACK
 TST (SP)+ ; BR IF IT ISN'T THE TERMINATOR
 60\$: MOV (SP)+, RO ; IF TERMINATOR POP IT OFF THE STACK
 3\$: ADD #2, (SP) ; RESTORE RO
 RTI ; ADJUST RETURN PC
 4\$: CMPB #HT, (SP) ; RETURN
 BEQ 8\$; BRANCH IF <HT>
 CMPB #CRLF, (SP) ; ;BRANCH IF NOT <CRLF>
 BNE 5\$
 TST (SP)+ ; ;POP <CR><LF> EQUIV
 TYPE ; ;TYPE A CR AND LF
 \$CRLF
 CLR B \$CHARCNT ; ;CLEAR CHARACTER COUNT
 BR 2\$; ;GET NEXT CHARACTER
 5\$: JSR PC, \$TYPEC ; ;GO TYPE THIS CHARACTER
 6\$: CMPB \$FILLC, (SP)+ ; ;IS IT TIME FOR FILLER CHARS.?

```

15459 062274 001350          BNE      2$          ;; IF NO GO GET NEXT CHAR.
15460 062276 013746 001054  MOV      $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED
15461                                     ;; AND THE NULL CHAR.
15462 062302 105366 000001  7$:     DECB     1(SP)  ;; DOES A NULL NEED TO BE TYPED?
15463 062306 002770          BLT      6$          ;; BR IF NO--GO POP THE NULL OFF OF STACK
15464 062310 004737 062346  JSR     PC,$TYPEC  ;; GO TYPE A NULL
15465 062314 105337 062412  DECB     $CHARCNT  ;; DO NOT COUNT AS A COUNT
15466 062320 000770          BR       7$          ;; LOOP
    
```

;HORIZONTAL TAB PROCESSOR

```

15470 062322 112716 000040  8$:     MOVB     #' (SP)  ;; REPLACE TAB WITH SPACE
15471 062326 004737 062346  9$:     JSR     PC,$TYPEC  ;; TYPE A SPACE
15472 062332 132737 000007 062412  BITB     #7,$CHARCNT  ;; BRANCH IF NOT AT
15473 062340 001372          BNE     9$          ;; TAB STOP
15474 062342 005726          TST     (SP)+       ;; POP SPACE OFF STACK
15475 062344 000724          BR      2$          ;; GET NEXT CHARACTER
15476 062346 105777 116476  $TYPEC: TSTB     @STPS  ;; WAIT UNTIL PRINTER IS READY
15477 062352 100375          BPL     $TYPEC
15478 062354 116677 000002 116470  MOVB     2(SP),@STPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.
15479 062362 122766 000015 000002  CMPB     #CR,2(SP)  ;; IS CHARACTER A CARRIAGE RETURN?
15480 062370 001003          BNE     1$          ;; BRANCH IF NO
15481 062372 105037 062412  CLRB     $CHARCNT  ;; YES--CLEAR CHARACTER COUNT
15482 062376 000406          BR      $TYPEX
15483 062400 122766 000012 000002  1$:     CMPB     #LF,2(SP) ;; IS CHARACTER A LINE FEED?
15484 062406 001402          BEQ     $TYPEX  ;; BRANCH IF YES
15485 062410 105227          INCB     (PC)+     ;; COUNT THE CHARACTER
15486 062412 000000  $CHARCNT: .WORD  0  ;; CHARACTER COUNT STORAGE
15487 062414 000207  $TYPEX: RTS      PC
    
```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```

15488
15489
15490
15491
15492
15493
15494
15495
15496
15497
15498
15499
15500
15501
15502
15503
15504
15505
15506
15507
15508
15509
15510
15511
15512
15513
15514

;*****
;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
;OCTAL (ASCII) NUMBER AND TYPE IT.
;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
;$CALL:
;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
;      TYPOS   N              ;; CALL FOR TYPEOUT
;      .BYTE   N              ;; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;      .BYTE   M              ;; M=1 OR 0
;                               ;; 1=TYPE LEADING ZEROS
;                               ;; 0=SUPPRESS LEADING ZEROS
;$STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
;$TYPOS OR $TYPOC
;$CALL:
;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
;      TYPON   N              ;; CALL FOR TYPEOUT
;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
;$CALL:
;      MOV      NUM,-(SP)      ;; NUMBER TO BE TYPED
;      TYPOC   N              ;; CALL FOR TYPEOUT
    
```

```

15515 062416 017646 000000          STYPOS: MOV      2(SP),-(SP)      ;; PICKUP THE MODE
15516 062422 116637 000001 062641  MOVB     1(SP),SOFILL      ;; LOAD ZERO FILL SWITCH
15517 062430 112637 062643          MOVB     (SP)+,SOMODE+1    ;; NUMBER OF DIGITS TO TYPE
15518 062434 062716 000002          ADD      #2,(SP)          ;; ADJUST RETURN ADDRESS
15519 062440 000406          BR       STYPON
15520 062442 112737 000001 062641  STYPOC: MOVB     #1,SOFILL      ;; SET THE ZERO FILL SWITCH
15521 062450 112737 000006 062643  MOVB     #6,SOMODE+1      ;; SET FOR SIX(6) DIGITS
15522 062456 112737 000005 062640  STYPON: MOVB     #5,SOCNT      ;; SET THE ITERATION COUNT
15523 062464 010346          MOV      R3,-(SP)        ;; SAVE R3
15524 062466 010446          MOV      R4,-(SP)        ;; SAVE R4
15525 062470 010546          MOV      R5,-(SP)        ;; SAVE R5
15526 062472 113704 062643          MOVB     SOMODE+1,R4      ;; GET THE NUMBER OF DIGITS TO TYPE
15527 062476 005404          NEG      R4
15528 062500 062704 000006          ADD      #6,R4           ;; SUBTRACT IT FOR MAX. ALLOWED
15529 062504 110437 062642          MOVB     R4,SOMODE        ;; SAVE IT FOR USE
15530 062510 113704 062641          MOVB     SOFILL,R4        ;; GET THE ZERO FILL SWITCH
15531 062514 016605 000012          MOV      12(SP),R5       ;; PICKUP THE INPUT NUMBER
15532 062520 005003          CLR      R3              ;; CLEAR THE OUTPUT WORD
15533 062522 006105          1$:     ROL      R5        ;; ROTATE MSB INTO "C"
15534 062524 000404          BR       3$              ;; GO DO MSB
15535 062526 006105          2$:     ROL      R5        ;; FORM THIS DIGIT
15536 062530 006105          ROL      R5
15537 062532 006105          ROL      R5
15538 062534 010503          MOV      R5,R3
15539 062536 006103          3$:     ROL      R3        ;; GET LSB OF THIS DIGIT
15540 062540 105337 062642          DECB     SOMODE           ;; TYPE THIS DIGIT?
15541 062544 100016          BPL      7$              ;; BR IF NO
15542 062546 042703 177770          BIC      #177770,R3      ;; GET RID OF JUNK
15543 062552 001002          BNE     4$              ;; TEST FOR 0
15544 062554 005704          TST     R4              ;; SUPPRESS THIS 0?
15545 062556 001403          BEQ     5$              ;; BR IF YES
15546 062560 005204          4$:     INC      R4        ;; DON'T SUPPRESS ANYMORE 0'S
15547 062562 052703 000060          BIS     #'0,R3          ;; MAKE THIS DIGIT ASCII
15548 062566 052703 000040          5$:     BIS     #' ,R3    ;; MAKE ASCII IF NOT ALREADY
15549 062572 110337 062636          MOVB     R3,8$          ;; SAVE FOR TYPING
15550 062576 104401 062636          TYPE     8$            ;; GO TYPE THIS DIGIT
15551 062602 105337 062640          7$:     DECB     SOCNT      ;; COUNT BY 1
15552 062606 003347          BGT     2$              ;; BR IF MORE TO DO
15553 062610 002402          BLT     6$              ;; BR IF DONE
15554 062612 005204          INC     R4              ;; INSURE LAST DIGIT ISN'T A BLANK
15555 062614 000744          BR      2$              ;; GO DO THE LAST DIGIT
15556 062616 012605          6$:     MOV      (SP)+,R5   ;; RESTORE R5
15557 062620 012604          MOV      (SP)+,R4       ;; RESTORE R4
15558 062622 012603          MOV      (SP)+,R3       ;; RESTORE R3
15559 062624 016666 000002 000004  MOV      2(SP),4(SP)     ;; SET THE STACK FOR RETURNING
15560 062632 012616          MOV      (SP)+,(SP)
15561 062634 000002          RTI
15562 062636          8$:     .BYTE    0        ;; RETURN
15563 062637          .BYTE    0        ;; STORAGE FOR ASCII DIGIT
15564 062640          .BYTE    0        ;; TERMINATOR FOR TYPE ROUTINE
15565 062641          .BYTE    0        ;; OCTAL DIGIT COUNTER
15566 062642 000000          .WORD    0        ;; ZERO FILL SWITCH
15567          .WORD    0        ;; NUMBER OF DIGITS TO TYPE
15568          .SBTTL  APT COMMUNICATIONS ROUTINE
15569
15570          ;;*****

```

15571	062644	112737	000001	063110	\$ATY1:	MOVB	#1,\$FFLG	:: TO REPORT FATAL ERROR
15572	062652	112737	000001	063106	\$ATY3:	MOVB	#1,\$MFLG	:: TO TYPE A MESSAGE
15573	062660	000403				BR	\$ATYC	
15574	062662	112737	000001	063110	\$ATY4:	MOVB	#1,\$FFLG	:: TO ONLY REPORT FATAL ERROR
15575	062670				\$ATYC:			
15576	062670	010046				MOV	RO,-(SP)	:: PUSH RO ON STACK
15577	062672	010146				MOV	R1,-(SP)	:: PUSH R1 ON STACK
15578	062674	105737	063106			TSTB	\$MFLG	:: SHOULD TYPE A MESSAGE?
15579	062700	001450				BEQ	5\$:: IF NOT: BR
15580	062702	122737	000001	001140		CMPB	#APTENV,\$ENV	:: OPERATING UNDER APT?
15581	062710	001031				BNE	3\$:: IF NOT: BR
15582	062712	132737	000100	001141		BITB	#APTPOOL,\$ENVM	:: SHOULD SPOOL MESSAGES?
15583	062720	001425				BEQ	3\$:: IF NOT: BR
15584	062722	017600	000004			MOV	24(SP),RO	:: GET MESSAGE ADDR.
15585	062726	062766	000002	000004		ADD	#2,4(SP)	:: BUMP RETURN ADDR.
15586	062734	005737	001120		1\$:	TST	\$MSGTYPE	:: SEE IF DONE W/ LAST XMISSION?
15587	062740	001375				BNE	1\$:: IF NOT: WAIT
15588	062742	010037	001134			MOV	RO,\$MSGAD	:: PUT ADDR IN MAILBOX
15589	062746	105720			2\$:	TSTB	(RO)+	:: FIND END OF MESSAGE
15590	062750	001376				BNE	2\$	
15591	062752	163700	001134			SUB	\$MSGAD,RO	:: SUB START OF MESSAGE
15592	062756	006200				ASR	RO	:: GET MESSAGE LNTH IN WORDS
15593	062760	010037	001136			MOV	RO,\$MSGLGT	:: PUT LENGTH IN MAILBOX
15594	062764	012737	000004	001120		MOV	#4,\$MSGTYPE	:: TELL APT TO TAKE MSG.
15595	062772	000413				BR	5\$	
15596	062774	017637	000004	063020	3\$:	MOV	24(SP),4\$:: PUT MSG ADDR IN JSR LINKAGE
15597	063002	062766	000002	000004		ADD	#2,4(SP)	:: BUMP RETURN ADDRESS
15598	063010	013746	177776			MOV	177776,-(SP)	:: PUSH 177776 ON STACK
15599	063014	004737	062134			JSR	PC,\$TYPE	:: CALL TYPE MACRO
15600	063020	000000			4\$:	.WORD	0	
15601	063022				5\$:			
15602	063022	105737	063110		10\$:	TSTB	\$FFLG	:: SHOULD REPORT FATAL ERROR?
15603	063026	001416				BEQ	12\$:: IF NOT: BR
15604	063030	005737	001140			TST	\$ENV	:: RUNNING UNDER APT?
15605	063034	001413				BEQ	12\$:: IF NOT: BR
15606	063036	005737	001120		11\$:	TST	\$MSGTYPE	:: FINISHED LAST MESSAGE?
15607	063042	001375				BNE	11\$:: IF NOT: WAIT
15608	063044	017637	000004	001122		MOV	24(SP),\$FATAL	:: GET ERROR #
15609	063052	062766	000002	000004		ADD	#2,4(SP)	:: BUMP RETURN ADDR.
15610	063060	005237	001120			INC	\$MSGTYPE	:: TELL APT TO TAKE ERROR
15611	063064	105037	063110		12\$:	CLRB	\$FFLG	:: CLEAR FATAL FLAG
15612	063070	105037	063107			CLRB	\$LFLG	:: CLEAR LOG FLAG
15613	063074	105037	063106			CLRB	\$MFLG	:: CLEAR MESSAGE FLAG
15614	063100	012601				MOV	(SP)+,R1	:: POP STACK INTO R1
15615	063102	012600				MOV	(SP)+,RO	:: POP STACK INTO RO
15616	063104	000207				RTS	PC	:: RETURN
15617	063106	000			\$MFLG:	.BYTE	0	:: MESSG. FLAG
15618	063107	000			\$LFLG:	.BYTE	0	:: LOG FLAG
15619	063110	000			\$FFLG:	.BYTE	0	:: FATAL FLAG
15620		063112				.EVEN		
15621		000200			APTSIZE=	200		
15622		000001			APTENV=	001		
15623		000100			APTPOOL=	100		
15624		000040			APTCSUP=	040		
15625								
15626					.SBTTL	TRAP	DECODER	

```

15627
15628
15629
15630
15631
15632
15633
15634 063112 010046
15635 063114 016600 000002
15636 063120 005740
15637 063122 111000
15638 063124 006300
15639 063126 016000 063146
15640 063132 000200
15641
15642
15643
15644
15645 063134 011646
15646 063136 016666 000004 000002
15647 063144 000002
15648
15649
15650
15651
15652
15653
15654
15655
15656 063146 063134
15657 063150 062134
15658 063152 062442
15659 063154 062416
15660 063156 062456
15661
15662
15663
15664
15665
15666 063160 000000
15667
15668 063162 000000
15669 063164 000000
15670 063166 000000
15671 063170 000000
15672 063172 000000
15673 063174 000000
15674 063176 000000
15675
15676 063200 000000
15677
15678
15679 063202 177400
15680 063204 177400
15681 063206 177400
15682 063210 177400

```

```

;*****
;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;GO TO THAT ROUTINE.
STRAP:  MOV    RO, -(SP)           ;; SAVE RO
        MOV    2(SP), RO         ;; GET TRAP ADDRESS
        TST    -(RO)             ;; BACKUP BY 2
        MOVB   (RO), RO          ;; GET RIGHT BYTE OF TRAP
        ASL    RO                ;; POSITION FOR INDEXING
        MOV    STRPAD(RO), RO     ;; INDEX TO TABLE
        RTS    RO                ;; GO TO ROUTINE

;; THIS IS USE TO HANDLE THE "GETPRI" MACRO
STRAP2: MOV    (SP), -(SP)        ;; MOVE THE PC DOWN
        MOV    4(SP), 2(SP)      ;; MOVE THE PSW DOWN
        RTI                      ;; RESTORE THE PSW

.SBTTL  TRAP TABLE

; THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
; BY THE "TRAP" INSTRUCTION.
;
; ROUTINE
;-----
STRPAD: .WORD  STRAP2
        $TYPE  ;; CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
        $TYPOC ;; CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS ;; CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON ;; CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)

; FLAGS, CONSTANTS, AND VARIABLES
BPTLOC: 0 ;; STORES 16 USER DEFINED MAINTENANCE
          ;; BREAKPOINTS
PRIFLG: 0 ;; FLAG USED BY BASIC TESTS FOR TRAP TEST
ERRFLG: 0 ;; FLAG USED BY BASIC TESTS FOR EMT TEST
SELTST: 0 ;; STORES SR<8:0> FOR LOOP ON SELECTED TEST
SCOFLG: 0 ;; USED BY BASIC TESTS FOR IOT TEST
RSVFLG: 0 ;; FLAG USED BY BASIC TEST OF RSVD INSTR TRAP
BERFLG: 0 ;; FLAG USED BY BASIC TEST OF BUS ERROR TRAPS
CATERR: 0 ;; FLAGS USED BY BUS ERROR AND RSVD INSTR TRAP
          ;; SERVICE ROUTINES
ONCE: 0 ;; FLAGS PROGRAM TITLE HAS BEEN PRINTED

; COMMON DATA STRUCTURES AND MISCELLANEOUS TABLES
OBUF: 177400 ;; DL11 OUTPUT TEST BUFFER
      177400
      177400
      177400

```

15683					
15684	063212	000004	IBUF:	.BLKW 4	;DL11 INPUT TEST BUFFER
15685					
15686	063222	063246	ATA:	DWTA	
15687	063224	063756		DWTB	
15688	063226	064554		DBTA	
15689	063230	064560		DBTB	
15690	063232	063236		MBUFO	
15691	063234	063242		MBUF1	
15692					
15693	063236	000000	MBUFO:	0	
15694	063240	000000		0	
15695	063242	000000	MBUF1:	0	
15696	063244	000000		0	
15697	063246	000000	DWTA:	0	
15698	063250	177777		-1	
15699	063252	177400		177400	
15700	063254	000377		377	
15701	063256	125252		125252	
15702	063260	052525	ALUADD:	052525	;ALSO SERVES AS NULL ENTRY FOR ALUADD
15703					
15704					
15705					
15706					
15707	063262	000000		000000	;SRC OP1
15708	063264	000000		000000	;DST OP1
15709	063266	000000		000000	;ANS1
15710	063270	177777		177777	;SRC OP2
15711	063272	177777		177777	;DST OP2
15712	063274	177776		177776	;ANS2
15713	063276	125252		125252	;SRC OP3
15714	063300	052525		052525	;DST OP3
15715	063302	177777		177777	;ANS3
15716	063304	052525		052525	;SRC OP4
15717	063306	125252		125252	;DST OP4
15718	063310	177777		177777	;ANS4
15719	063312	125252		125252	;SRC OP5
15720	063314	125252		125252	;DST OP5
15721	063316	052524		052524	;ANS5
15722	063320	052525		052525	;SRC OP6
15723	063322	052525		052525	;DST OP6
15724	063324	125252		125252	;ANS6
15725	063326	052525		052525	;SRC OP7
15726	063330	125253		125253	;DST OP7
15727	063332	000000		000000	;ANS7
15728	063334	125253		125253	;SRC OP8
15729	063336	052525		052525	;DST OP8
15730	063340	000000	ANDTAB:	000000	;ANS8 -- ALSO NULL ENTRY FOR ANDTAB
15731					
15732					
15733					
15734					
15735	063342	000000		000000	;SRC OP1
15736	063344	000000		000000	;DST OP1
15737	063346	000000		000000	;ANS1
15738	063350	177777		177777	;SRC OP2

;THIS TABLE OF 8 ENTRIES IS USED BY THE ALU ADD TEST IN THE
;COMBINED INSTRUCTION TESTS

;THIS TABLE OF 8 ENTRIES IS USED BY THE ALU "AND" TESTS IN THE
;COMBINED INSTRUCTION EXERCISER TESTS

15739	063352	177777	177777	:DST OP2
15740	063354	000000	000000	:ANS2
15741	063356	000000	000000	:SRC OP3
15742	063360	177777	177777	:DST OP3
15743	063362	177777	177777	:ANS3
15744	063364	177777	177777	:SRC OP4
15745	063366	000000	000000	:DST OP4
15746	063370	000000	000000	:ANS4
15747	063372	125252	125252	:SRC OP5
15748	063374	125252	125252	:DST OP5
15749	063376	000000	000000	:ANS5
15750	063400	052525	052525	:SRC OP6
15751	063402	052525	052525	:DST OP6
15752	063404	000000	000000	:ANS6
15753	063406	125252	125252	:SRC OP7
15754	063410	052525	052525	:DST OP7
15755	063412	052525	052525	:ANS7
15756	063414	052525	052525	:SRC OP8
15757	063416	125252	125252	:DST OP8
15758	063420	125252	125252	:ANS8 -- ALSO NULL ENTRY FOR ORTAB

ORTAB:

:THIS TABLE OF 8 ENTRIES IS USED BY THE ALU "OR" TEST IN THE
 :COMBINED INSTRUCTION EXERCISER TEST

15763	063422	000000	000000	:SRC OP1
15764	063424	000000	000000	:DEST OP1
15765	063426	000000	000000	:ANS1
15766	063430	177777	177777	:SRC OP2
15767	063432	177777	177777	:DST OP2
15768	063434	177777	177777	:ANS2
15769	063436	000000	000000	:SRC OP3
15770	063440	177777	177777	:DST OP3
15771	063442	177777	177777	:ANS3
15772	063444	177777	177777	:SRC OP4
15773	063446	000000	000000	:DST OP4
15774	063450	177777	177777	:ANS4
15775	063452	125252	125252	:SRC OP5
15776	063454	125252	125252	:DST OP5
15777	063456	125252	125252	:ANS5
15778	063460	052525	052525	:SRC OP6
15779	063462	052525	052525	:DST OP6
15780	063464	052525	052525	:ANS6
15781	063466	125252	125252	:SRC OP7
15782	063470	052525	052525	:DST OP7
15783	063472	177777	177777	:ANS7
15784	063474	052525	052525	:SRC OP8
15785	063476	125252	125252	:DST OP8
15786	063500	177777	177777	:ANS8 -- ALSO NULL ENTRY FOR ALUSUB

ALUSUB:

:THIS TABLE OF 8 ENTRIES IS USED BY THE ALU SUB TEST IN THE
 :COMBINED INSTRUCTION EXERCISER TESTS

15791	063502	000000	000000	:SRC OP1
15792	063504	000000	000000	:DST OP1
15793	063506	000000	000000	:ANS1
15794	063510	177777	177777	:SRC OP2

15795	063512	177777	177777	:DST OP2
15796	063514	000000	000000	:ANS2
15797	063516	125252	125252	:SRC OP3
15798	063520	052525	052525	:DST OP3
15799	063522	125253	125253	:ANS3
15800	063524	052525	052525	:SRC OP4
15801	063526	125252	125252	:DST OP4
15802	063530	052525	052525	:ANS4
15803	063532	125252	125252	:SRC OP5
15804	063534	125252	125252	:DST OP5
15805	063536	000000	000000	:ANS5
15806	063540	052525	052525	:SRC OP6
15807	063542	052525	052525	:DST OP6
15808	063544	000000	000000	:ANS6
15809	063546	052525	052525	:SRC OP7
15810	063550	125253	125253	:DST OP7
15811	063552	052526	052526	:ANS7
15812	063554	125253	125253	:SRC OP8
15813	063556	052525	052525	:DST OP8
15814	063560	125252	125252	:ANS8

INSTAB: TST R2
 CLR R2
 COM R2
 INC R2
 DEC R2
 ADC R2
 SBC R2
 ASR R2
 ASL R2
 CLRB R2
 COMB R2
 INCB R2
 DECB R2
 ADCB R2
 ADCB R2
 SBCB R2
 TSTB R2
 ASRB R2
 ASLB R2
 BISB (R3), R2
 XOR R3, R2
 CMPB (R3), R2
 BITB (R3), R2
 BICB (R3), R2
 MOVB (R3), R2
 CMP (R3), R2
 BIT (R3), R2
 BIC (R3), R2
 BIS (R3), R2
 SXT R2
 NEG R2
 SUB (R3), R2
 CMP R3, (R2)
 BIT R3, (R2)
 CMPB R3, (R2)

:BEGINNING OF INSTRUCTION TABLE OF INSTRUCTIONS
 :THAT TEST BUT SERVICE IN VARIOUS ROM LOCATIONS

15907				;	*	PAD LIMITS" ARE MAINTAINED.
15908				;	*	
15909						
15910	064112			TBL2:		
15911						
15912	064112			ASP1:		;A SCRATCH PAD - LO
15913	064112	201	001	R1A: .BYTE	201,001	;LOBYTE, HIBYTE=WRITE CODE, READ CODE
15914	064114	202	002	R2A: .BYTE	202,002	
15915	064116	203	003	R3A: .BYTE	203,003	
15916	064120	204	004	R4A: .BYTE	204,004	
15917	064122	205	005	R5A: .BYTE	205,005	
15918	064124	206	006	R6A: .BYTE	206,006	
15919	064126	210	010	FAC3.0: .BYTE	210,010	
15920	064130	211	011	FAC3.1: .BYTE	211,011	
15921	064132	212	012	FAC3.2: .BYTE	212,012	
15922	064134	213	013	FAC3.3: .BYTE	213,013	
15923	064136	214	014	FAC3.4: .BYTE	214,014	
15924	064140	215	015	FAC3.5: .BYTE	215,015	
15925	064142	216	016	UR6A: .BYTE	216,016	
15926	064144	217	017	FDST3: .BYTE	217,017	
15927	064146	220	020	WCSA.0: .BYTE	220,020	;A SCRATCH PAD-HI
15928	064150	221	021	WCSA.1: .BYTE	221,021	
15929	064152	222	022	GNUMAM: .BYTE	222,022	
15930	064154	223	023	CNSTSW: .BYTE	223,023	
15931	064156	226	026	CNSSW: .BYTE	226,026	
15932	064160	227	027	CNSCDR: .BYTE	227,027	
15933	064162	230	030	FAC1.0: .BYTE	230,030	
15934	064164	231	031	FAC1.1: .BYTE	231,031	
15935	064166	232	032	FAC1.2: .BYTE	232,032	
15936	064170	233	033	FAC1.3: .BYTE	233,033	
15937	064172	234	034	FAC1.4: .BYTE	234,034	
15938	064174	235	035	FAC1.5: .BYTE	235,035	
15939	064176	236	036	FPSHI: .BYTE	236,036	
15940	064200	237	037	ASP2: FDST1: .BYTE	237,037	
15941						
15942	064202			BSP1:		
15943	064202	241	041	R1B: .BYTE	241,041	;B SCRATCH PAD - LO
15944	064204	242	042	R2B: .BYTE	242,042	
15945	064206	243	043	R3B: .BYTE	243,043	
15946	064210	244	044	R4B: .BYTE	244,044	
15947	064212	245	045	R5B: .BYTE	245,045	
15948	064214	246	046	R6B: .BYTE	246,046	
15949	064216	250	050	FAC2.0: .BYTE	250,050	
15950	064220	251	051	FAC2.1: .BYTE	251,051	
15951	064222	252	052	FAC2.2: .BYTE	252,052	
15952	064224	253	053	FAC2.3: .BYTE	253,053	
15953	064226	254	054	FAC2.4: .BYTE	254,054	
15954	064230	255	055	FAC2.5: .BYTE	255,055	
15955	064232	256	056	UR6B: .BYTE	256,056	
15956	064234	257	057	FDST2: .BYTE	257,057	
15957	064236	260	060	WCSB.0: .BYTE	260,060	;B SCRATCH PAD - HI
15958	064240	261	061	WCSB.1: .BYTE	261,061	
15959	064242	262	062	WCSADR: .BYTE	262,062	
15960	064244	263	063	RZERO: .BYTE	263,063	
15961	064246	266	066	RVECT: .BYTE	266,066	
15962	064250	270	070	FACO.0: .BYTE	270,070	

15963	064252	272	072
15964	064254	273	073
15965	064256	274	074
15966	064260	275	075
15967	064262	276	076
15968	064264	277	077
15969			
15970	064266		
15971	064266	300	100
15972	064270	301	101
15973	064272	302	102
15974	064274	303	103
15975	064276	304	104
15976	064300	305	105
15977	064302	307	107
15978	064304	310	110
15979	064306	311	111
15980	064310	312	112
15981	064312	313	113
15982	064314	316	116
15983	064316	224	024
15984	064320	225	025
15985	064322	264	064
15986	064324	265	065
15987	064326	000000	
15988			
15989			
15990			
15991			
15992			
15993			
15994			
15995			
15996	064330		
15997	064330	120	137
15998	064332	145	145
15999	064334	150	151
16000	064336	156	177
16001	064340	320	343
16002	064342	353	357
16003	064344	000000	
16004			
16005			
16006			
16007			
16008			
16009			
16010			
16011			
16012			
16013	064346		
16014	064346	200	000
16015	064350	207	007
16016	064352	240	040
16017	064354	247	047
16018	064356	314	114

FACO.1:	.BYTE	272,072	
FACO.2:	.BYTE	273,073	
FACO.4:	.BYTE	274,074	
FACO.5:	.BYTE	275,075	
FEA:	.BYTE	276,076	
BSP2:	FDSTO:	.BYTE	277,077
CSP1:			
LJAM:	.BYTE	300,100	
LSERV:	.BYTE	301,101	
LPBA:	.BYTE	302,102	
LCUA:	.BYTE	303,103	
LFGIN:	.BYTE	304,104	
LWHAM:	.BYTE	305,105	
LTAG:	.BYTE	307,107	
CNSCO:	.BYTE	310,110	
CNSC1:	.BYTE	311,111	
CNSC2:	.BYTE	312,112	
CST200:	.BYTE	313,113	
CSP2:	CNSTO:	.BYTE	316,116
RT1A:	.BYTE	224,024	
RT2A:	.BYTE	225,025	
RT1B:	.BYTE	264,064	
RT2B:	.BYTE	265,065	
	.WORD	0	

;C SCRATCH PAD

```

;*
;* TABLE III
;*
;* THE FOLLOWING IS A LIST OF "NOP" OPERATION CODES
;* THAT WILL BE USED WITH A MED IN MED TEST 3 TO
;* ENSURE THAT A MED WITH THESE CODES WILL NOT HANG.

```

TBL3:				
NOPS:	.BYTE	120,137		;GROUP A
	.BYTE	145,145		;GROUP B
	.BYTE	150,151		;GROUP C
	.BYTE	156,177		;GROUP D
	.BYTE	320,343		;GROUP E
	.BYTE	353,357		;FROUP G
	.WORD	0		;A 0 TERMINATES TABLE

```

;*
;* TABLE IV
;*
;* THE LIST BELOW CONTAINS THOSE OPERATION CODES
;* CORRESPONDING TO THE INTERNAL REGISTERS WHICH MUST
;* BE TESTED SEPERATELY BECAUSE THEY ARE READ-ONLY,
;* WRITE-ONLY, OR USED IN MACRO CODE EXECUTION, ETC. . .

```

TBL4:				
ROA:	.BYTE	200,000		;LOBYTE HYBYTE - WRITE CODE, READ CODE
R7A:	.BYTE	207,007		;0 REPLACES ANY NON EXSISTENT CODES
ROB:	.BYTE	240,040		;EXCEPT IN THE CASE OF ROA
R7B:	.BYTE	247,047		
CNST2:	.BYTE	314,114		

16019	064360	317	117	CNST1: .BYTE	317,117
16020				;*	TABLE V
16021				;*	
16022	064362			TBL5:	
16023					
16024	064362	306		LCDTA: .BYTE	306
16025	064363	106		.BYTE	106
16026	064364	315		MD: .BYTE	315
16027	064365	115		.BYTE	115
16028	064366	267		CNSCTL: .BYTE	267
16029	064367	067		.BYTE	067
16030	064370	140		JAM: .BYTE	140
16031	064371	141		SERV: .BYTE	141
16032	064372	142		PBA: .BYTE	142
16033	064373	143		CUA: .BYTE	143
16034	064374	344		FLAG: .BYTE	344
16035	064375	144		.BYTE	144
16036	064376	345		DREG: .BYTE	345
16037	064377	146		REV: .BYTE	146
16038	064400	346		SREG: .BYTE	346
16039	064401	147		COUNT: .BYTE	147
16040	064402	347		NJA: .BYTE	347
16041	064403	351		RES: .BYTE	351
16042	064404	152		DCSO: .BYTE	152
16043	064405	352		.BYTE	352
16044	064406	153		DCS1: .BYTE	153
16045	064407	000		.BYTE	0
16046				.EVEN	
16047					
16048				;*	TABLE VI
16049				;*	
16050	064410			TBL6:	
16051					
16052	064410	003330		ULCDTA: .WORD	3330
16053	064412	003150		.WORD	3150
16054	064414	003375		UMD: .WORD	3375
16055	064416	003271		.WORD	3271
16056	064420	003240		UCNSCTL: .WORD	3240
16057	064422	003224		.WORD	3224
16058	064424	003160		UJAM: .WORD	3160
16059	064426	003161		USERV: .WORD	3161
16060	064430	003170		UPBA: .WORD	3170
16061	064432	003171		UCUA: .WORD	3171
16062	064434	003344		UFLAG: .WORD	3344
16063	064436	003320		.WORD	3320
16064	064440	003345		UDREG: .WORD	3345
16065	064442	003340		UREV: .WORD	3340
16066	064444	003350		USREG: .WORD	3350
16067	064446	003341		UCOUNT: .WORD	3341
16068	064450	003351		UNJA: .WORD	3351
16069	064452	003355		URES: .WORD	3355
16070	064454	003720		UDCSO: .WORD	3720
16071	064456	003724		UINIT: .WORD	3724
16072	064460	003721		UDCS1: .WORD	3721
16073					
16074				;*	TABLE VII

; THIS TABLE CONTAINS THE OPERATION
 ; CODES OF THOSE INTERNAL REGISTERS
 ; WHICH MUST BE TESTED USING THE
 ; MICROBREAK REGISTER. THEIR
 ; ASSOCIATED MICRO-ADDRESSES ARE IN
 ; THE NEXT TABLE

; INIT REG
 ; TABLE TERMINATOR

; THIS TABLE CONTAINS THE MICRO-ADDRESSES
 ; WHICH ARE LOADED INTO THE MICROBREAK
 ; REG. TO TEST THE OPERATION CODES
 ; CONTAINED IN THE PRECEDING TABLE.

16075
16076
16077
16078
16079
16080
16081
16082
16083
16084
16085
16086
16087
16088
16089
16090
16091
16092
16093
16094
16095
16096
16097
16098
16099
16100
16101
16102
16103
16104
16105
16106
16107
16108
16109
16110
16111
16112
16113
16114
16115
16116
16117
16118
16119
16120
16121
16122
16123
16124
16125
16126
16127
16128
16129
16130

064462

064462 000100 077600
064466 000101 000010
064472 000102 020000
064476 000103 000004
064502 000104 050000
064506 000105 054000
064512 000107 024000
064516 000110 177400
064522 000111 177600
064526 000112 100000
064532 000113 000200
064536 000114 000002
064542 000116 000000
064546 000117 000001
064552 000000

064554
064554 000 377 252
064557 125
064560
064560 000 001 120
064563 253

064564
064564
064564 027523 020102 051504
064572 020124
064574 040527 020123 051504
064602 020124
064604 042040 051505 004524
064612 024040 051111 004451
064620 052040 051505 004524
064626 024040 041520 004451
064634 024040 050123 004451
064642 050050 053523 000051
064650 027523 020102 042522
064656 020123 040527 020123
064664 042522 020123 051504
064672 020124 050117 020040
064700 051123 020103 050117
064706 020040 042524 052123
064714 020011 050050 024503
064722 020011 051450 024520
064730 024011 051520 024527
064736 000
064737 123 041057 051440
064744 004520 040527 020123
064752 050123 020011 044450

```

;*
;* THIS TABLE HOLDS THE OPERATION CODES AND THE CONSTANT
;* VALUE EXPECTED FOR CERTAIN INTERNAL REGISTERS.
;BL7:
    
```

```

CLJAM: .WORD 100,77600
CLSERV: .WORD 101,10
CLPBA: .WORD 102,20000
CLCUA: .WORD 103,4
CLFGIN: .WORD 104,50000
CLWHAM: .WORD 105,54000
CLTAG: .WORD 107,24000
CCNSCO: .WORD 110,177400
CCNSC1: .WORD 111,177600
CCNSC2: .WORD 112,100000
CCST200: .WORD 113,200
CCNST2: .WORD 114,2
CCNSTO: .WORD 116,0
CCNST1: .WORD 117,1
        .WORD 0
    
```

```

.EVEN
DBTA:
.BYTE 000,377,252,125
DBTB:
.BYTE 000,001,120,253
    
```

;MESSAGE TABLES

```

EM1:
EM2:
EM4: .ASCII 'S/B DST '
EM7: .ASCII 'WAS DST '
EM6: .ASCII ' DEST'<HT>
EM5: .ASCIZ ' (IR)'<HT>' TEST'<HT>' (PC)'<HT>' (SP)'<HT>' (PSW)'
EM10: .ASCIZ 'S/B RES WAS RES DST OP SRC OP TEST'<HT>' (PC)'<HT>' (SP)'<HT>' (PSW)'
EM3: .ASCIZ 'S/B SP'<HT>'WAS SP'<HT>' (IR)'<HT>' TEST'<HT>' (PC)'<HT>' (PSW)'
    
```

16131	064760	024522	020011	042524	
16132	064766	052123	020011	050050	
16133	064774	024503	024011	051520	
16134	065002	024527	000		
16135	065005	011	020011	051511	DH2: .ASCIZ <HT><HT>' IS R3'
16136	065012	051040	000063		
16137	065016	004411	044440	020123	DH4: .ASCIZ <HT><HT>' IS R5'
16138	065024	032522	000		
16139	065027	015	042412	042116	EOP1: .ASCIZ <15><12>'END PASS # '
16140	065034	050040	051501	020123	
16141	065042	020043	000		
16142	065045	011	051105	047522	EOP2: .ASCIZ <HT>'ERROR COUNT = '
16143	065052	020122	047503	047125	
16144	065060	020124	020075	000	
16145	065065	015	046412	026504	IDENT1: .ASCIZ <15><12>'MD-11-DQKDA-A KD11-K BASIC LOGIC TESTS'<15><12>
16146	065072	030461	042055	045521	
16147	065100	040504	040455	020040	
16148	065106	045440	030504	026461	
16149	065114	020113	040502	044523	
16150	065122	020103	047514	044507	
16151	065130	020103	042524	052123	
16152	065136	006523	000012		
16153	065142	005015	051124	050101	BEMSG: .ASCIZ <CR><LF>'TRAPPED TO 4 PC = '
16154	065150	042520	020104	047524	
16155	065156	032040	050040	020103	
16156	065164	020075	000		
16157	065167	015	052012	040522	RSMSG: .ASCIZ <CR><LF>'TRAPPED TO 10 PC = '
16158	065174	050120	042105	052040	
16159	065202	020117	030061	050040	
16160	065210	020103	020075	000	
16161	065215	124	051505	051524	EM11: .ASCIZ 'TESTS SKIPPED'
16162	065222	051440	044513	050120	
16163	065230	042105	000		
16164	065233	040	050040	004503	DH11: .ASCIZ " PC"<HT>"EXPCTD"<HT>"ACTUAL"<HT>"(TEST #'S)"
16165	065240	054105	041520	042124	
16166	065246	040411	052103	040525	
16167	065254	004514	052050	051505	
16168	065262	020124	023443	024523	
16169	065270	000			
16170	065271	115	042105	042040	EM12: .ASCIZ /MED DID NOT ABORT IN USER MODE/
16171	065276	042111	047040	052117	
16172	065304	040440	047502	052122	
16173	065312	044440	020116	051525	
16174	065320	051105	046440	042117	
16175	065326	000105			
16176	065330	042515	020104	054105	EM13: .ASCIZ /MED EXECUTED IN USER MODE/
16177	065336	041505	052125	042105	
16178	065344	044440	020116	051525	
16179	065352	051105	046440	042117	
16180	065360	000105			
16181	065362	042515	020104	044103	EM14: .ASCIZ /MED CHANGED PSW/
16182	065370	047101	042507	020104	
16183	065376	051520	000127		
16184	065402	044515	051103	041117	EM15: .ASCIZ /MICROBREAK TRAP-TO-4 DID NOT OCCUR/
16185	065410	042522	045501	052040	
16186	065416	040522	026520	047524	

16187	065424	032055	042040	042111	
16188	065432	047040	052117	047440	
16189	065440	041503	051125	000	
16190	065445	114	043517	052503	EM17: .ASCIZ /LOGCUA LOGGED WRONG/
16191	065452	020101	047514	043507	
16192	065460	042105	053440	047522	
16193	065466	043516	000		
16194	065471	103	050123	041440	EM21: .ASCIZ /CSP CONSTANT WRONG/
16195	065476	047117	052123	047101	
16196	065504	020124	051127	047117	
16197	065512	000107			
16198	065514	040502	020104	040504	EM22: .ASCIZ /BAD DATA READ BY A MED/
16199	065522	040524	051040	040505	
16200	065530	020104	054502	040440	
16201	065536	046440	042105	000	
16202	065543	116	020117	042117	EM23: .ASCIZ /NO ODD PC TRAP/
16203	065550	020104	041520	052040	
16204	065556	040522	000120		
16205	065562	042117	020104	042101	EM24: .ASCIZ /ODD ADR. BIT NOT SET IN CPU ERR REG OR LOG JAM/
16206	065570	027122	041040	052111	
16207	065576	047040	052117	051440	
16208	065604	052105	044440	020116	
16209	065612	050103	020125	051105	
16210	065620	020122	042522	020107	
16211	065626	051117	046040	043517	
16212	065634	045040	046501	000	
16213	065641	120	054510	020123	EM26: .ASCIZ /PHYS BA LOGGED WRONG/
16214	065646	040502	046040	043517	
16215	065654	042507	020104	051127	
16216	065662	047117	000107		
16217	065666	040503	044103	020105	EM27: .ASCIZ /CACHE PARITY ERROR LOGGED IN BAKUP MODE/
16218	065674	040520	044522	054524	
16219	065702	042440	051122	051117	
16220	065710	046040	043517	042507	
16221	065716	020104	047111	041040	
16222	065724	045501	050125	046440	
16223	065732	042117	000105		
16224	065736	040503	044103	020105	EM30: .ASCIZ /CACHE PARITY TRAPPED WHEN DISABLED/
16225	065744	040520	044522	054524	
16226	065752	052040	040522	050120	
16227	065760	042105	053440	042510	
16228	065766	020116	044504	040523	
16229	065774	046102	042105	000	
16230	066001	111	051516	051124	EM41: .ASCIZ /INSTR. NOT ABORTED IN CACHE ABORT MODE/
16231	066006	020056	047516	020124	
16232	066014	041101	051117	042524	
16233	066022	020104	047111	041440	
16234	066030	041501	042510	040440	
16235	066036	047502	052122	046440	
16236	066044	042117	000105		
16237	066050	042515	047515	054522	EM32: .ASCIZ /MEMORY ERR REG INCORRECT/
16238	066056	042440	051122	051040	
16239	066064	043505	044440	041516	
16240	066072	051117	042522	052103	
16241	066100	000			
16242	066101	124	046511	047505	EM33: .ASCIZ /TIMEOUT BIT NOT SET IN CPU ERR REG OR LOG JAM/

16243	066106	052125	041040	052111	
16244	066114	047040	052117	051440	
16245	066122	052105	044440	020116	
16246	066130	050103	020125	051105	
16247	066136	020122	042522	020107	
16248	066144	051117	046040	043517	
16249	066152	045040	046501	000	
16250	066157	116	020117	046111	EM34: .ASCIZ /NO ILLEGAL INTERNAL ADR TRAP/
16251	066164	042514	040507	020114	
16252	066172	047111	042524	047122	
16253	066200	046101	040440	051104	
16254	066206	052040	040522	000120	
16255	066214	047111	051124	040516	EM35: .ASCIZ /INTRNAL ADR ERR BIT NOT SET IN CPU ERR REG OR LOG JAM/
16256	066222	020114	042101	020122	
16257	066230	051105	020122	044502	
16258	066236	020124	047516	020124	
16259	066244	042523	020124	047111	
16260	066252	041440	052520	042440	
16261	066260	051122	051040	043505	
16262	066266	047440	020122	047514	
16263	066274	020107	040512	000115	
16264	066302	040514	052123	044440	EM36: .ASCIZ "LAST INTR/TRAP VECTOR NOT LOGGED IN FLAG REG"
16265	066310	052116	027522	051124	
16266	066316	050101	053040	041505	
16267	066324	047524	020122	047516	
16268	066332	020124	047514	043507	
16269	066340	042105	044440	020116	
16270	066346	046106	043501	051040	
16271	066354	043505	000		
16272	066357	114	043517	043040	EM37: .ASCIZ /LOG FIRST MODE DID NOT INHIBIT ERROR LOG AFTER FIRST ERROR/
16273	066364	051111	052123	046440	
16274	066372	042117	020105	044504	
16275	066400	020104	047516	020124	
16276	066406	047111	044510	044502	
16277	066414	020124	051105	047522	
16278	066422	020122	047514	020107	
16279	066430	043101	042524	020122	
16280	066436	044506	051522	020124	
16281	066444	051105	047522	000122	
16282	066452	051105	047522	020122	EM40: .ASCIZ /ERROR LOG WAS NOT REENABLED, ODD ADR BIT CLR IN CPUERR/
16283	066460	047514	020107	040527	
16284	066466	020123	047516	020124	
16285	066474	042522	047105	041101	
16286	066502	042514	026104	047440	
16287	066510	042104	040440	051104	
16288	066516	041040	052111	041440	
16289	066524	051114	044440	020116	
16290	066532	050103	042525	051122	
16291	066540	000			
16292	066541	116	020117	040503	EM31: .ASCIZ /NO CACHE PARITY TRAP/
16293	066546	044103	020105	040520	
16294	066554	044522	054524	052040	
16295	066562	040522	000120		
16296	066566	047514	023040	044040	EM42: .ASCIZ /LO & HI BYTE & TAG PARITY BITS NOT SET IN LOG SERVICE/
16297	066574	020111	054502	042524	
16298	066602	023040	052040	043501	

16299	066610	050040	051101	052111	
16300	066616	020131	044502	051524	
16301	066624	047040	052117	051440	
16302	066632	052105	044440	020116	
16303	066640	047514	020107	042523	
16304	066646	053122	041511	000105	
16305	066654	047514	023040	044040	EM43: .ASCIZ /LO & HI BYTE & TAG PARITY BITS NOT SET IN MEM ERR REG/
16306	066662	020111	054502	042524	
16307	066670	023040	052040	043501	
16308	066676	050040	051101	052111	
16309	066704	020131	044502	051524	
16310	066712	047040	052117	051440	
16311	066720	052105	044440	020116	
16312	066726	042515	020115	051105	
16313	066734	020122	042522	000107	
16314	066742	040503	044103	020105	EM45: .ASCIZ /CACHE TAG LOGGED WRONG/
16315	066750	040524	020107	047514	
16316	066756	043507	042105	053440	
16317	066764	047522	043516	000	
16318	066771	103	041501	042510	EM16: .ASCIZ /CACHE DATA LOGGED WRONG/
16319	066776	042040	052101	020101	
16320	067004	047514	043507	042105	
16321	067012	053440	047522	043516	
16322	067020	000			
16323	067021	105	051511	051440	EMEIS1: .ASCIZ 'EIS SET COND CODES WRONG'
16324	067026	052105	041440	047117	
16325	067034	020104	047503	042504	
16326	067042	020123	051127	047117	
16327	067050	000107			
16328	067052	044505	020123	040507	EMEIS2: .ASCIZ 'EIS GAVE WRONG RESULT'
16329	067060	042526	053440	047522	
16330	067066	043516	051040	051505	
16331	067074	046125	000124		
16332	067100	052501	047524	044455	EM46: .ASCIZ 'AUTO-INCREMENT (DECREMT) DID NOT OCCUR IN EIS'
16333	067106	041516	042522	042515	
16334	067114	052116	024040	042504	
16335	067122	051103	046505	024524	
16336	067130	042040	042111	047040	
16337	067136	052117	047440	041503	
16338	067144	051125	044440	020116	
16339	067152	044505	000123		
16340	067156	050040	053523	051011	DHEIS1: .ASCII 'PSW'<HT>'REG-WAS-REG+1'<HT>'REG-S/B-REG+1'<HT>
16341	067164	043505	053455	051501	
16342	067172	051055	043505	030453	
16343	067200	051011	043505	051455	
16344	067206	041057	051055	043505	
16345	067214	030453	011		
16346	067217	040	050040	004503	DH46: .ASCIZ 'PC'<HT>'(IR)'<HT>'TEST'
16347	067224	024040	051111	004451	
16348	067232	052040	051505	000124	
16349	067240	020040	041520	046411	DH15: .ASCIZ /PC/<HT>/MEDCODE MICROBK REG./
16350	067246	042105	047503	042504	
16351	067254	046440	041511	047522	
16352	067262	045502	051040	043505	
16353	067270	000056			
16354	067272	020040	041520	046411	DH17: .ASCIZ /PC/<HT>/MEDCODE EXPECTD RECEIVD/

16355	067300	042105	047503	042504				
16356	067306	042440	050130	041505				
16357	067314	042124	051040	041505				
16358	067322	044505	042126	000				
16359	067327	040	050040	000103	DH23:	.ASCIZ	/ PC/	
16360	067334	020040	041520	041411	DH24:	.ASCIZ	/ PC/<HT>/CPUERR/<HT>/LOGJAM/	
16361	067342	052520	051105	004522				
16362	067350	047514	045107	046501				
16363	067356	000						
16364	067357	040	050040	004503	DH25:	.ASCIZ	/ PC/<HT>/FLGREG/	
16365	067364	046106	051107	043505				
16366	067372	000						
16367	067373	040	050040	004503	DH26:	.ASCIZ	' PC'<HT>'<17:16>-S/B PA-<15:0> <17:16>-WAS PA-<15:0>'	
16368	067400	030474	035067	033061				
16369	067406	026476	027523	020102				
16370	067414	040520	036055	032461				
16371	067422	030072	020076	036040				
16372	067430	033461	030472	037066				
16373	067436	053455	051501	050040				
16374	067444	026501	030474	035065				
16375	067452	037060	000					
16376	067455	040	050040	004503	DH27:	.ASCIZ	/ PC/<HT>/LOGPBA/<HT>/LOGDATA/<HT>/LOGTAG/	
16377	067462	047514	050107	040502				
16378	067470	046011	043517	040504				
16379	067476	040524	046011	043517				
16380	067504	040524	000107					
16381	067510	020040	041520	046411	DH32:	.ASCIZ	/ PC/<HT>/MEMERR/	
16382	067516	046505	051105	000122				
16383	067524	020040	041520	046011	DH42:	.ASCIZ	/ PC/<HT>/LOGSERVICE/	
16384	067532	043517	042523	053122				
16385	067540	042503	000					
16386	067543	040	050040	004503	DH44:	.ASCIZ	/ PC/<HT>/EXPCT/<HT>/RECVD/	
16387	067550	054105	041520	004524				
16388	067556	042522	053103	000104				
16389								.EVEN
16390	067564	001016	001076	001100	DT15:	.WORD	SERRPC, STMP0, STMP1, 0	
16391	067572	000000						
16392	067574	001016	001100	001102	DT21:	.WORD	SERRPC, STMP1, STMP2, SREG0, 0	
16393	067602	001062	000000					
16394	067606	001016	001100	001102	DT22:	.WORD	SERRPC, STMP1, STMP2, STMP3, 0	
16395	067614	001104	000000					
16396	067620	001016	000000		DT23:	.WORD	SERRPC, 0	
16397	067624	001016	001064	001062	DT24:	.WORD	SERRPC, SREG1, SREG0, 0	
16398	067632	000000						
16399	067634	001016	001062	000000	DT25:	.WORD	SERRPC, SREG0, 0	
16400	067642	001016	001064	001066	DT26:	.WORD	SERRPC, SREG1, SREG2, SREG0, SREG3, 0	
16401	067650	001062	001070	000000				
16402	067656	001016	001070	001064	DT27:	.WORD	SERRPC, SREG3, SREG1, SREG2, 0	
16403	067664	001066	000000					
16404	067670	001060	001066	001070	DTEIS1:	.WORD	SREG0, SREG2, SREG3, SREG1, SREG4	
16405	067676	001064	001072					
16406	067702	001016	001076	001062	DT46:	.WORD	SERRPC, STMP0, SREG0, 0	
16407	067710	000000						
16408								
16409	067712	000	000	000	DF15:	.BYTE	0, 0	
16410	067714	000	000	000	DF17:	.BYTE	0, 0, 0	

16411		067720				.EVEN		
16412	067720					DT1:		
16413	067720					DT2:		
16414	067720					DT4:		
16415	067720	001072				DT10:	.WORD	SREG4
16416	067722	001070				DT7:	.WORD	SREG3
16417	067724	001066				DT6:	.WORD	SREG2
16418	067726	001064	001062	001016		DT5:	.WORD	SREG1, SREG0, SERRPC, SREG5, SREGAD, 0
16419	067734	001074	001060	000000				
16420	067742	001072	001070	001064		DT3:	.WORD	SREG4, SREG3, SREG1, SREG0, SERRPC, SREGAD, 0
16421	067750	001062	001016	001060				
16422	067756	000000						
16423	067760	001016	001124	001062		DT11:	.WORD	SERRPC, \$TESTN, SREG0, 0
16424	067766	000000						
16425		000001				.END		

EM4	064564	293	16108#						
EM40	066452	472	16282#						
EM41	066001	479	16230#						
EM42	066566	486	16296#						
EM43	066654	493	16305#						
EM45	066742	356	16314#						
EM46	067100	514	16332#						
EM5	064612	298	16113#						
EM6	064604	303	16112#						
EM7	064574	308	16110#						
EOP1	065027	14992	16139#						
EOP2	065045	14995	16142#						
ERRA	061764	3718	15351#						
ERRFLG	063164	3720*	3725*	15351*	15669#				
ERRVEC=	000004	122#	14199*	14212*	15241	15242*	15244*	15247*	
EX002	001640	591#							
E001	001632	581#							
E003	001656	601	602	603	606#				
E004	001674	616	617	618	621#				
E005	001712	631	632	633	636#				
E006	001724	650#							
E007	001740	664#							
E010	001756	677	680#						
E011	001776	693	696#						
E012	002016	711#							
E013	002042	728#							
E014	002064	743#							
E016	002136	776	777	778	781#				
E022	002304	862	863	864	867#				
E024	002370	906#							
E025	002410	917	920#						
E026	002430	930	933#						
E027	002450	947#							
E030	002470	961#							
E031	002510	974#							
E035	002670	1053#							
E036	002710	1067#							
E037	002730	1081#							
E040	002756	1098#							
E042	003034	1136#							
E043	003056	1153#							
E044	003104	1169#							
E045	003132	1184#							
E1015A	002102	757#							
E2002	001644	595#							
E2015	002112	764#							
E2017	002176	806#							
E2020	002226	828#							
E2021	002260	851#							
E2023	002344	891#							
E2032	002550	996#							
E2033	002610	1018#							
E2034	002650	1039#							
E2041	003014	1120#							
E2046	003216	1218#							
FACO.0	064250	15962#							

H10

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 309
 DQKDA.P11 08-FEB-77 16:17 CROSS REFERENCE TABLE -- USER SYMBOLS

PWRVEC=	000024	128#	15019*	15020*	15029*	15035*	15050*	15051*
RCSR =	177560	564#	3787	3873				
RDBR =	177562	565#						
RDFLAG=	000144	540#	14381	14408				
RDLCUR=	000103	548#	14395					
RDLDAT=	000106	554#	14550	14820				
RDLFGI=	000104	550#	14675	14834	14947	14962		
RDLJAM=	000100	542#	14462	14628	14658	14706	14881	14918
RDLPBA=	000102	546#	14477	14544	14644	14795		
RDLSER=	000101	544#	14484	14774				
RDLTAG=	000107	556#	14556	14806				
RDLWHA=	000105	552#						
RDWHAM=	000022	538#	14376	14427				
RES	064403	16041#						
RESTAR	061122	15130	15139	15141#	15144	15198	15205	
RESVEC=	000010	123#	14200*	14213*				
REV	064377	16037#						
RSBERT	061074	15133#	15201					
RSERR	061046	3751	13133	14213	15126#			
RSMSG	065167	15132	16157#					
RSVFLG	063172	3739#	3745*	15123*	15672#			
RSVTST	061040	3736	15123#					
RT1A	064316	15983#						
RT1B	064322	15985#						
RT2A	064320	15984#						
RT2B	064324	15986#						
RVECT	064246	15961#						
RZERO	064244	15960#						
RDA	064346	16014#						
ROB	064352	16016#						
R1A	064112	15913#						
R1B	064202	15943#						
R2A	064114	15914#						
R2B	064204	15944#						
R3A	064116	15915#						
R3B	064206	15945#						
R4A	064120	15916#						
R4B	064210	15946#						
R5A	064122	15917#						
R5B	064212	15947#						
R6A	064124	15918#						
R6B	064214	15948#						
R7A	064350	14304	16015#					
R7B	064354	14316	16017#					
SCOFLG	063170	3614#	3621*	15285*	15671#			
SCOPEA	061536	3615	15285#					
SELTST	063166	15231#	15232*	15233	15670#			
SERV	064371	16031#						
S0BERR	042752	11270	11279#					
S0B1	042724	11265#	11348					
S0B2	042740	11260	11263	11273#				
S0B3	043116	11265	11346#					
S0B4	043120	11344	11348#					
S0B5	042732	11269#	11273					
SREG	064400	16038#						
STACK =	001000	27#	735	1215	1231	14198	15064	

TST102	007342	2599	2608#
TST103	007422	2626	2633#
TST104	007502	2651	2658#
TST105	007562	2676	2683#
TST106	007642	2701	2709#
TST107	007720	2727	2735#
TST11	003540	1363	1371#
TST110	007776	2753	2761#
TST111	010056	2779	2787#
TST112	010136	2805	2813#
TST113	010200	2828	2835#
TST114	010242	2850	2858#
TST115	010332	2894#	
TST116	010422	2930#	
TST117	010464	2941	2948#
TST12	003576	1385	1393#
TST120	010526	2959	2966#
TST121	010570	2977	2984#
TST122	010644	2999	3007#
TST123	010670	3015	3023#
TST124	010726	3043	3051#
TST125	010744	3057	3065#
TST126	010760	3071	3079#
TST127	010774	3085	3093#
TST13	003632	1407	1415#
TST130	011014	3100	3108#
TST131	011034	3115	3123#
TST132	011052	3130	3138#
TST133	011070	3144	3152#
TST134	011106	3159	3167#
TST135	011124	3174	3182#
TST136	011144	3189	3197#
TST137	011164	3204	3212#
TST14	003664	1425	1433#
TST140	011204	3219	3227#
TST141	011224	3234	3242#
TST142	011240	3248	3256#
TST143	011260	3263	3271#
TST144	011300	3278	3286#
TST145	011320	3293	3301#
TST146	011336	3308	3316#
TST147	011352	3322	3330#
TST15	003720	1443	1451#
TST150	011372	3337	3345#
TST151	011412	3352	3360#
TST152	011432	3367	3375#
TST153	011500	3393	3401#
TST154	011542	3411	3419#
TST155	011576	3428	3436#
TST156	011664	3464	3472#
TST157	011720	3485	3493#
TST16	003764	1462	1470#
TST160	011766	3509	3517#
TST161	012054	3540	3548#
TST162	012126	3561	3570#
TST163	012250	3601	3610#

TST164	012316	3631#	
TST165	012440	3662	3671#
TST166	012514	3694#	
TST167	012560	3706	3714#
TST17	004030	1481	1489#
TST170	012624	3726	3733#
TST171	012706	3757#	
TST172	012756	3770	3778#
TST173	013056	3806#	
TST174	013116	3816	3827#
TST175	013150	3836	3845#
TST176	013204	3854	3871#
TST177	013516	3943#	
TST2	003324	1258	1266#
TST20	004064	1499	1506#
TST200	013540	3950	3957#
TST201	013560	3964	3971#
TST202	013620	3984	3991#
TST203	013662	4006	4013#
TST204	013720	4026	4033#
TST205	013760	4048	4055#
TST206	014002	4062	4069#
TST207	014024	4077	4084#
TST21	004130	1521	1529#
TST210	014046	4092	4099#
TST211	014070	4107	4114#
TST212	014114	4122	4129#
TST213	014136	4136	4143#
TST214	014160	4151	4158#
TST215	014202	4166	4173#
TST216	014224	4181	4188#
TST217	014272	4207	4214#
TST22	004170	1544	1552#
TST220	014334	4228	4235#
TST221	014402	4254	4261#
TST222	014434	4271	4278#
TST223	014562	4323	4330#
TST224	014616	4340	4347#
TST225	014666	4366	4373#
TST226	014724	4384	4391#
TST227	014774	4410	4417#
TST23	004224	1561	1569#
TST230	015044	4436	4443#
TST231	015174	4488	4495#
TST232	015250	4515	4522#
TST233	015314	4541	4548#
TST234	015364	4567	4574#
TST235	015432	4593	4600#
TST236	015502	4620	4627#
TST237	015556	4647	4654#
TST24	004264	1583	1591#
TST240	015632	4674	4681#
TST241	015704	4701	4708#
TST242	015754	4727	4734#
TST243	016022	4753	4760#
TST244	016072	4779	4786#

TST245	016140	4805	4812#
TST246	016210	4831	4838#
TST247	016260	4857	4864#
TST25	004334	1609	1617#
TST250	016336	4884	4891#
TST251	016412	4911	4919#
TST252	016470	4939	4947#
TST253	016544	4967	4975#
TST254	016622	4995	5003#
TST255	016700	5023	5031#
TST256	016760	5053	5061#
TST257	017032	5079	5087#
TST26	004410	1638	1647#
TST260	017130	5114	5122#
TST261	017204	5140	5148#
TST262	017270	5171	5179#
TST263	017344	5197	5205#
TST264	017424	5227	5235#
TST265	017474	5252	5260#
TST266	017540	5279	5286#
TST267	017606	5306	5313#
TST27	004444	1658	1666#
TST270	017654	5332	5339#
TST271	017720	5357	5364#
TST272	017770	5383	5390#
TST273	020036	5409	5416#
TST274	020104	5435	5442#
TST275	020154	5461	5468#
TST276	020222	5487	5494#
TST277	020272	5513	5520#
TST3	003340	1271	1278#
TST30	004502	1676	1684#
TST300	020336	5538	5545#
TST301	020404	5564	5571#
TST302	020454	5590	5597#
TST303	020520	5616	5623#
TST304	020570	5642	5649#
TST305	020640	5667	5674#
TST306	020704	5693	5700#
TST307	020754	5719	5726#
TST31	004526	1692	1700#
TST310	021022	5745	5752#
TST311	021072	5771	5778#
TST312	021140	5797	5804#
TST313	021210	5823	5830#
TST314	021260	5849	5856#
TST315	021326	5875	5882#
TST316	021400	5902	5910#
TST317	021456	5931	5939#
TST32	004562	1710	1718#
TST320	021544	5963	5971#
TST321	021626	5995	6003#
TST322	021704	6022	6030#
TST323	021760	6049	6057#
TST324	022034	6076	6084#
TST325	022112	6103	6111#

TST326	022166	6130	6138#
TST327	022244	6157	6165#
TST33	004622	1729	1737#
TST330	022316	6183	6191#
TST331	022372	6210	6218#
TST332	022450	6237	6245#
TST333	022522	6264	6272#
TST334	022600	6291	6299#
TST335	022656	6318	6326#
TST336	022730	6345	6353#
TST337	023006	6372	6380#
TST34	004656	1750	1758#
TST340	023062	6399	6407#
TST341	023140	6426	6434#
TST342	023214	6453	6461#
TST343	023272	6480	6488#
TST344	023350	6507	6515#
TST345	023424	6534	6542#
TST346	023474	6561	6568#
TST347	023544	6587	6594#
TST35	004712	1771	1778#
TST350	023614	6613	6620#
TST351	023664	6639	6646#
TST352	023734	6665	6672#
TST353	024014	6691	6699#
TST354	024064	6713	6721#
TST355	024140	6739	6747#
TST356	024204	6760	6768#
TST357	024264	6787	6795#
TST36	004746	1791	1799#
TST360	024346	6813	6821#
TST361	024422	6839	6847#
TST362	024466	6860	6868#
TST363	024540	6888	6895#
TST364	024614	6915	6922#
TST365	024662	6939	6946#
TST366	024730	6963	6970#
TST367	025004	6988	6995#
TST37	005002	1812	1820#
TST370	025052	7012	7019#
TST371	025126	7037	7044#
TST372	025166	7056	7063#
TST373	025232	7076	7083#
TST374	025276	7096	7104#
TST375	025342	7117	7125#
TST376	025430	7147	7154#
TST377	025516	7176	7183#
TST4	003356	1284	1292#
TST40	005042	1834	1842#
TST400	025606	7205	7212#
TST401	025676	7234	7241#
TST402	025764	7263	7270#
TST403	026052	7292	7299#
TST404	026142	7321	7328#
TST405	026232	7350	7357#
TST406	026304	7371	7379#

TST407	026356	7393	7401#
TST41	005106	1857	1866#
TST410	026426	7415	7423#
TST411	026476	7437	7445#
TST412	026542	7464	7471#
TST413	026612	7491	7498#
TST414	026666	7518	7525#
TST415	026742	7545	7552#
TST416	027016	7573	7581#
TST417	027076	7602	7610#
TST42	005150	1880	1889#
TST420	027172	7635	7643#
TST421	027254	7664	7672#
TST422	027324	7692	7699#
TST423	027374	7719	7726#
TST424	027450	7746	7753#
TST425	027524	7773	7780#
TST426	027600	7800	7807#
TST427	027660	7828	7836#
TST43	005212	1903	1912#
TST430	027740	7857	7865#
TST431	030020	7885	7893#
TST432	030072	7913	7920#
TST433	030142	7940	7947#
TST434	030220	7967	7974#
TST435	030304	7996	8004#
TST436	030372	8027	8035#
TST437	030446	8054	8061#
TST44	005250	1933#	
TST440	030524	8080	8087#
TST441	030600	8106	8113#
TST442	030656	8132	8139#
TST443	030726	8152	8160#
TST444	030776	8173	8181#
TST445	031062	8204	8213#
TST446	031146	8236	8245#
TST447	031232	8268	8277#
TST45	005276	1941	1949#
TST450	031316	8300	8309#
TST451	031412	8337	8346#
TST452	031506	8374	8383#
TST453	031604	8411	8420#
TST454	031702	8448	8457#
TST455	031776	8485	8494#
TST456	032072	8522	8531#
TST457	032202	8563	8572#
TST46	005340	1960	1968#
TST460	032300	8600	8609#
TST461	032370	8632	8641#
TST462	032460	8664	8673#
TST463	032550	8696	8705#
TST464	032640	8728	8737#
TST465	032700	8750	8758#
TST466	032740	8771	8779#
TST467	033002	8792	8800#
TST47	005374	1978	1986#

TST470	033044	8813	8821#
TST471	033106	8834	8843#
TST472	033152	8856	8864#
TST473	033224	8884	8891#
TST474	033276	8912	8919#
TST475	033334	8932	8939#
TST476	033374	8951	8958#
TST477	033442	8975	8982#
TST5	003400	1299	1307#
TST50	005436	2001	2009#
TST500	033510	8999	9006#
TST501	033554	9019	9027#
TST502	033632	9047	9054#
TST503	033712	9073	9081#
TST504	033770	9100	9108#
TST505	034050	9127	9135#
TST506	034122	9149	9157#
TST507	034174	9171	9179#
TST51	005474	2018	2026#
TST510	034242	9193	9201#
TST511	034310	9215	9223#
TST512	034360	9237	9245#
TST513	034430	9259	9267#
TST514	034502	9281	9289#
TST515	034556	9309	9316#
TST516	034636	9340	9347#
TST517	034712	9367	9374#
TST52	005520	2041#	
TST520	034760	9394	9401#
TST521	035036	9421	9429#
TST522	035110	9449	9456#
TST523	035162	9476	9483#
TST524	035234	9503	9510#
TST525	035306	9529	9536#
TST526	035370	9557	9565#
TST527	035444	9586	9594#
TST53	005554	2051	2059#
TST530	035526	9615	9623#
TST531	035602	9644	9652#
TST532	035664	9673	9681#
TST533	035746	9702	9709#
TST534	036030	9730	9738#
TST535	036110	9759	9767#
TST536	036170	9787	9795#
TST537	036244	9815	9822#
TST54	005606	2068	2076#
TST540	036314	9842	9849#
TST541	036374	9870	9877#
TST542	036444	9897	9904#
TST543	036516	9924	9931#
TST544	036570	9951	9957#
TST545	036642	9977	9984#
TST546	036714	10004	10011#
TST547	036772	10031	10038#
TST55	005654	2093	2101#
TST550	037054	10059	10067#

TST551	037132	10088	10096#			
TST552	037220	10118	10126#			
TST553	037300	10148	10156#			
TST554	037366	10178	10186#			
TST555	037466	10212	10220#			
TST556	037552	10242	10250#			
TST557	037636	10272	10280#			
TST56	005724	2118	2126#			
TST560	037722	10301	10309#			
TST561	037760	10321	10328#			
TST562	040024	10341	10349#			
TST563	040072	10363	10371#			
TST564	040142	10385	10393#			
TST565	040212	10407	10415#			
TST566	040262	10429	10437#			
TST567	040334	10451	10459#			
TST57	005764	2137	2145#			
TST570	040406	10473	10481#			
TST571	040450	10494	10502#			
TST572	040520	10516	10524#			
TST573	040562	10537	10545#			
TST574	040632	10559	10567#			
TST575	040672	10577	10582#	10589#		
TST576	040732	10599	10604#	10611#		
TST577	041002	10621	10631#	10638#		
TST6	003420	1313	1321#			
TST60	006036	2162	2169#			
TST600	041042	10648	10653#	10660#		
TST601	041120	10670	10680#	10683#	10692#	
TST602	041166	10702	10707#	10710#	10718#	
TST603	041244	10728	10732#	10742#	10749#	
TST604	041304	10759	10764#	10771#		
TST605	041362	10781	10791#	10794#	10801#	
TST606	041430	10811	10816#	10819#	10827#	
TST607	041476	10837	10842#	10845#	10853#	
TST61	006100	2184	2192#			
TST610	041544	10863	10868#	10871#	10879#	
TST611	041620	10889	10892#	10897#	10900#	10909#
TST612	041674	10919	10922#	10927#	10930#	10939#
TST613	041752	10955	10967#			
TST614	042040	10998#				
TST615	042122	11027#				
TST616	042172	11052#				
TST617	042250	11068	11081#			
TST62	006132	2201	2208#			
TST620	042334	11097	11114#			
TST621	042416	11133	11146#			
TST622	042502	11162	11179#			
TST623	042562	11195	11207#			
TST624	042650	11224	11241#			
TST625	042700	11249	11257#			
TST626	042754	11275	11278#	11284#		
TST627	043010	11296	11303#			
TST63	006172	2218	2226#			
TST630	043044	11315	11322#			
TST631	043100	11334	11341#			

TST632	043134	11353	11360#			
TST633	043236	11390	11401#			
TST634	043342	11427	11438#			
TST635	043450	11464	11475#			
TST636	043626	11491	11499	11511	11520	11535#
TST637	044004	11551	11559	11571	11580	11595#
TST64	006250	2244	2252#			
TST640	044066	11619#				
TST641	044122	11629	11637#			
TST642	044160	11647	11655#			
TST643	044272	11685#				
TST644	044410	11717#				
TST645	044526	11750#				
TST646	044642	11783#				
TST647	045046	11840#				
TST65	006326	2272	2280#			
TST650	045250	11897#				
TST651	045366	11928#				
TST652	045504	11959#				
TST653	045622	11990#				
TST654	045740	12021#				
TST655	046056	12052#				
TST656	046210	12088#				
TST657	046330	12123#				
TST66	006362	2290	2298#			
TST660	046470	12168#				
TST661	046770	12251#				
TST662	047036	12271#				
TST663	047172	12311#				
TST664	047306	12349#				
TST665	047370	12374#				
TST666	047452	12399#				
TST667	047534	12424#				
TST67	006420	2308	2316#			
TST670	047620	12448#				
TST671	047726	12479	12487#			
TST672	050034	12517	12525#			
TST673	050174	12556	12577#			
TST674	050356	12625	12637#			
TST675	050540	12685	12697#			
TST676	050634	12727#				
TST677	050740	12759#				
TST7	003452	1331	1338#			
TST70	006460	2326	2334#			
TST700	051044	12791#				
TST701	051124	12814#				
TST702	051204	12837#				
TST703	051264	12859#				
TST704	051364	12892#				
TST705	051470	12910	12927#			
TST706	051576	12945	12962#			
TST707	051704	12980	12997#			
TST71	006510	2343	2351#			
TST710	051772	13023#				
TST711	052060	13050#				
TST712	052146	13077#				

G11

SCM4 = 000005	223#	224#	225#	226#	227#	228#													
SCPUOP 001146	252#																		
SCRLF 001115	231#	14998	15136	15327	15350	15362	15381	15386	15390	15454	15489								
SDEVCT 001130	243#																		
SDOAGN 060600	14988	15000	15006#																
SENDAD 060570	174	15002#																	
SENDCT 060524	14990#																		
SENULL 060604	15009#																		
SENV 001140	248#	15329	15433	15580	15604														
SEVM 001141	249#	3924	15435	15440	15582														
SEOP 060474	14981#																		
SEOPCT 060516	14987#	14991																	
SERFLG 001003	188#	15214	15250	15252	15258*	15284	15317*	15350											
SERMAX 001015	194#	15252	15275*	15284															
SERROR 061544	3918	14941	15304#																
SERRPC 001016	195#	15321*	15322*	15323	15350	15368	16390	16392	16394	16396	16397	16399	16400						
	16402	16406	16418	16420	16423														
SERRTB 001150	275#	15376																	
SERTY 061772	15326	15361#																	
SERTL 001012	192#	1225#	14996	15137*	15320*	15350													
SESCAP 001112	229#	15225*	15227*	15274*	15342	15344	15350												
SETABL 001140	247#																		
SETEND 001150	167	259#																	
SFATAL 001122	240#	15608#																	
SFFLG 063110	15571*	15574*	15602	15611*	15619#														
SFILLC 001056	213#	15458	15489																
SFILLS 001055	212#	15489																	
SGADR 001020	196#																		
SGOAT 001024	198#																		
SGET42 060560	14999#																		
SGTSMR= ***** U	15662																		
SHD = 000000	12																		
SHIBTS 000700	162#																		
SICNT 001004	189#	15265*	15266	15268*	15283														
SILLUP 060762	15019	15035	15057#																
SINTAG 001035	203#																		
SITEMB 001014	193#	15323*	15331	15350	15365														
SLF 001116	232#	15350	15489																
SLFLG 063107	15612*	15618#																	
SLPADR 001006	190#	3935*	15256*	15272*	15277	15283													
SLPERR 001010	191#	10944*	10972*	11007*	11032*	11057*	11086*	11119*	11151*	11184*	11212*	11365*	11407*						
	11444*	11601*	11662*	11691*	11788*	11845*	11902*	11933*	11964*	11995*	12026*	12061*	12093*						
	12129*	12173*	12256*	12429*	12796*	12819*	12842*	12864*	13004*	13019*	13030*	13045*	13057*						
	13072*	13084*	13102*	13114*	13132*	13156*	13172*	13201*	13255*	13303*	13325*	13363*	13385*						
	13423*	13445*	13483*	13505*	13531*	13550*	13575*	13594*	14402*	14433*	15256	15273*	15283						
	15341																		
SMAIL 001120	163	167	238#	15271	15329	15433													
SMBADR 000702	163#																		
SMLG 063106	15572*	15578	15613*	15617#															
SMSGAD 001134	245#	15588*	15591																
SMSGLC 001136	246#	15593*																	
SMSGTY 001120	239#	15586	15594*	15606	15610*														
SIXCNT 061534	15269	15283#																	
SNULL 001054	211#	15460	15489																
SNWTST= 000001	1236#	1249#	1263#	1275#	1289#	1304#	1318#	1335#	1351#	1368#	1390#	1412#	1430#						
	1448#	1467#	1486#	1503#	1526#	1549#	1566#	1588#	1614#	1644#	1663#	1681#	1697#						

H11

1715#	1734#	1755#	1775#	1796#	1817#	1839#	1863#	1886#	1909#	1930#	1946#	1965#
1983#	2006#	2023#	2038#	2056#	2073#	2098#	2123#	2142#	2166#	2189#	2205#	2223#
2249#	2277#	2295#	2313#	2331#	2348#	2372#	2388#	2406#	2425#	2457#	2494#	2531#
2568#	2605#	2630#	2655#	2680#	2706#	2732#	2758#	2784#	2810#	2832#	2855#	2891#
2927#	2945#	2963#	2981#	3004#	3020#	3048#	3062#	3076#	3090#	3105#	3120#	3135#
3149#	3164#	3179#	3194#	3209#	3224#	3239#	3253#	3268#	3283#	3298#	3313#	3327#
3342#	3357#	3372#	3398#	3416#	3433#	3469#	3490#	3514#	3545#	3567#	3607#	3628#
3668#	3691#	3711#	3730#	3754#	3775#	3803#	3824#	3842#	3860#	3862	3940#	3954#
3968#	3988#	4010#	4030#	4052#	4066#	4081#	4096#	4111#	4126#	4140#	4155#	4170#
4185#	4211#	4232#	4258#	4275#	4327#	4344#	4370#	4388#	4414#	4440#	4492#	4519#
4545#	4571#	4597#	4624#	4651#	4678#	4705#	4731#	4757#	4783#	4809#	4835#	4861#
4888#	4916#	4944#	4972#	5000#	5028#	5058#	5084#	5119#	5145#	5176#	5202#	5232#
5257#	5283#	5310#	5336#	5361#	5387#	5413#	5439#	5465#	5491#	5517#	5542#	5568#
5594#	5620#	5646#	5671#	5697#	5723#	5749#	5775#	5801#	5827#	5853#	5879#	5907#
5936#	5968#	6000#	6027#	6054#	6081#	6108#	6135#	6162#	6188#	6215#	6242#	6269#
6296#	6323#	6350#	6377#	6404#	6431#	6458#	6485#	6512#	6539#	6565#	6591#	6617#
6643#	6669#	6696#	6718#	6744#	6765#	6792#	6818#	6844#	6865#	6892#	6919#	6943#
6967#	6992#	7016#	7041#	7060#	7080#	7101#	7122#	7151#	7180#	7209#	7238#	7267#
7296#	7325#	7354#	7376#	7398#	7420#	7442#	7468#	7495#	7522#	7549#	7578#	7607#
7640#	7669#	7696#	7723#	7750#	7777#	7804#	7833#	7862#	7890#	7917#	7944#	7971#
8001#	8032#	8058#	8084#	8110#	8136#	8157#	8178#	8210#	8242#	8274#	8306#	8343#
8380#	8417#	8454#	8491#	8528#	8569#	8606#	8638#	8670#	8702#	8734#	8755#	8776#
8797#	8818#	8840#	8861#	8888#	8916#	8936#	8955#	8979#	9003#	9024#	9051#	9078#
9105#	9132#	9154#	9176#	9198#	9220#	9242#	9264#	9286#	9313#	9344#	9371#	9398#
9426#	9453#	9480#	9507#	9533#	9562#	9591#	9620#	9649#	9678#	9706#	9735#	9764#
9792#	9819#	9846#	9874#	9901#	9928#	9954#	9981#	10008#	10035#	10064#	10093#	10123#
10153#	10183#	10217#	10247#	10277#	10306#	10325#	10346#	10368#	10390#	10412#	10434#	10456#
10478#	10499#	10521#	10542#	10564#	10586#	10608#	10635#	10657#	10689#	10715#	10746#	10768#
10798#	10824#	10850#	10876#	10906#	10936#	10964#	10995#	11024#	11049#	11078#	11111#	11143#
11176#	11204#	11238#	11254#	11281#	11300#	11319#	11338#	11357#	11398#	11435#	11472#	11532#
11592#	11616#	11634#	11652#	11682#	11714#	11747#	11780#	11837#	11894#	11925#	11956#	11987#
12018#	12049#	12085#	12117#	12119	12161#	12163	12248#	12268#	12308#	12346#	12371#	12396#
12421#	12445#	12484#	12522#	12574#	12634#	12694#	12724#	12756#	12788#	12811#	12834#	12856#
12887#	12889	12924#	12959#	12994#	13020#	13047#	13074#	13104#	13144#	13146	13174#	13189#
13214#	13230#	13247#	13267#	13269	13327#	13329	13387#	13389	13447#	13449	13507#	13509
13552#	13554	13596#	13626#	13657#	13693#	13729#	13765#	13801#	13832#	13856#	13885#	13914#
13943#	13965#	13967	14011#	14035#	14060#	14084#	14112#	14141#	14186#	14188	14224#	14226
14275#	14277	14321#	14323	14351#	14353	14438#	14440	14499#	14501	14569#	14571	14606#
14608	14683#	14685	14722#	14724	14844#	14846	14931#					
15522#	15551#	15564#										
15517#	15521#	15526	15529#	15540#	15566#							
15234	15237	15257	15267	15276#								
242#	1226#	14984#	14985#	14993	15009	15263	15284					
165#												
15053	15060#											
15055#												
3922	15019#	15050										
15053#												
15029	15035#											
230#	15350	15489										
15663												
15663												
15663												
15663												
215#	15306	16404	16418	16420								
217#	14739#	14750#	14767	14837#	16392	16397	16399	16400	16406	16418	16420	16423

SOCNT 062640
SOMODE 062642
SOVER 061504
SPASS 001126
SPASTH 000706
SPOWER 060770
SPWRAD 060756
SPWRDN 060610
SPWRMG 060752
SPWRUP 060662
SQUES 001114
SR0CHR= ***** U
SR0DEC= ***** U
SR0LIN= ***** U
SR0OCT= ***** U
SREGAD 001060
SREGO 001062

	15259	15260	15261	15273	15276	15283	15297	15298	15299	15300	15320	15324	15336
SSWREG	001142												
SSWRNK=	000000												
STESTN	001124												
STINES	001110												
STKB	001046												
STKS	001044												
STNPO	001076												
STMP1	001100												
STMP2	001102												
STMP3	001104												
STMP4	001106												
STN =	000770												
	15339	15350											
	250#	3926											
	15219												
	241#	3937*	15223	15271*	15281*	16423							
	228#	3936*	11721*	11754*	14983*	15259*	15266	15269*	15283				
	208#												
	207#												
	223#	13602*	13632*	13663*	13699*	13735*	13771*	13807*	13838*	13862*	13891*	13920*	13949*
	14090*	14118*	14151*	14249*	14258	14388*	14418*	16390	16406				
	224#	14250*	14263*	14269	14310*	14345*	14389*	14419*	16390	16392	16394		
	225#	14241*	14251	14261	14268*	14270	14311*	14346*	16392	16394			
	226#	14257*	14261	14312*	16394								
	227#	14246*	14264	14267*									
	1#	12	1236	1240	1241#	1244	1249	1253	1254#	1258	1263	1267	1268#
	1271	1275	1279	1280#	1284	1289	1293	1294#	1299	1304	1308	1309#	1313
	1318	1322	1323#	1331	1335	1339	1340#	1347	1351	1355	1356#	1363	1368
	1372	1373#	1385	1390	1394	1395#	1407	1412	1416	1417#	1425	1430	1434
	1435#	1443	1448	1452	1453#	1462	1467	1471	1472#	1481	1486	1490	1491#
	1499	1503	1507	1508#	1521	1526	1530	1531#	1544	1549	1553	1554#	1561
	1566	1570	1571#	1583	1588	1592	1593#	1609	1614	1618	1619#	1638	1644
	1648	1649#	1658	1663	1667	1668#	1676	1681	1685	1686#	1692	1697	1701
	1702#	1710	1715	1719	1720#	1729	1734	1738	1739#	1750	1755	1759	1760#
	1771	1775	1779	1780#	1791	1796	1800	1801#	1812	1817	1821	1822#	1834
	1839	1843	1844#	1857	1863	1867	1868#	1880	1886	1890	1891#	1903	1909
	1913	1914#	1930	1934	1935#	1941	1946	1950	1951#	1960	1965	1969	1970#
	1978	1983	1987	1988#	2001	2006	2010	2011#	2018	2023	2027	2028#	2038
	2042	2043#	2051	2056	2060	2061#	2068	2073	2077	2078#	2093	2098	2102
	2103#	2118	2123	2127	2128#	2137	2142	2146	2147#	2162	2166	2170	2171#
	2184	2189	2193	2194#	2201	2205	2209	2210#	2218	2223	2227	2228#	2244
	2249	2253	2254#	2272	2277	2281	2282#	2290	2295	2299	2300#	2308	2313
	2317	2318#	2326	2331	2335	2336#	2343	2348	2352	2353#	2372	2376	2377#
	2383	2388	2392	2393#	2401	2406	2410	2411#	2420	2425	2429	2430#	2452
	2457	2461	2462#	2488	2494	2498	2499#	2525	2531	2535	2536#	2562	2568
	2572	2573#	2599	2605	2609	2610#	2626	2630	2634	2635#	2651	2655	2659
	2660#	2676	2680	2684	2685#	2701	2706	2710	2711#	2727	2732	2736	2737#
	2753	2758	2762	2763#	2779	2784	2788	2789#	2805	2810	2814	2815#	2828
	2832	2836	2837#	2850	2855	2859	2860#	2891	2895	2896#	2927	2931	2932#
	2941	2945	2949	2950#	2959	2963	2967	2968#	2977	2981	2985	2986#	2999
	3004	3008	3009#	3015	3020	3024	3025#	3043	3048	3052	3053#	3057	3062
	3066	3067#	3071	3076	3080	3081#	3085	3090	3094	3095#	3100	3105	3109
	3110#	3115	3120	3124	3125#	3130	3135	3139	3140#	3144	3149	3153	3154#
	3159	3164	3168	3169#	3174	3179	3183	3184#	3189	3194	3198	3199#	3204
	3209	3213	3214#	3219	3224	3228	3229#	3234	3239	3243	3244#	3248	3253
	3257	3258#	3263	3268	3272	3273#	3278	3283	3287	3288#	3293	3298	3302
	3303#	3308	3313	3317	3318#	3322	3327	3331	3332#	3337	3342	3346	3347#
	3352	3357	3361	3362#	3367	3372	3376	3377#	3393	3398	3402	3403#	3411
	3416	3420	3421#	3428	3433	3437	3438#	3464	3469	3473	3474#	3485	3490
	3494	3495#	3509	3514	3518	3519#	3540	3545	3549	3550#	3561	3567	3571
	3572#	3601	3607	3611	3612#	3628	3632	3633#	3662	3668	3672	3673#	3691
	3695	3696#	3706	3711	3715	3716#	3726	3730	3734	3735#	3754	3758	3759#
	3770	3775	3779	3780#	3803	3807	3808#	3816	3824	3828	3829#	3836	3842
	3846	3847#	3854	3860	3872	3873#	3940	3945	3947#	3950	3954	3959	3961#
	3964	3968	3973	3975#	3984	3988	3993	3995#	4006	4010	4015	4017#	4026
	4030	4035	4037#	4048	4052	4057	4059#	4062	4066	4071	4073#	4077	4081
	4086	4088#	4092	4096	4101	4103#	4107	4111	4116	4118#	4122	4126	4131

4133	4136	4140	4145	4147	4151	4155	4160	4162	4166	4170	4175	4177
4181	4185	4190	4192	4207	4211	4216	4218	4228	4232	4237	4239	4254
4258	4263	4265	4271	4275	4280	4282	4323	4327	4332	4334	4340	4344
4349	4351	4366	4370	4375	4377	4384	4388	4393	4395	4410	4414	4419
4421	4436	4440	4445	4447	4488	4492	4497	4499	4515	4519	4524	4526
4541	4545	4550	4552	4567	4571	4576	4578	4593	4597	4602	4604	4620
4624	4629	4631	4647	4651	4656	4658	4674	4678	4683	4685	4701	4705
4710	4712	4727	4731	4736	4738	4753	4757	4762	4764	4779	4783	4788
4790	4805	4809	4814	4816	4831	4835	4840	4842	4857	4861	4866	4868
4884	4888	4893	4895	4911	4916	4921	4923	4939	4944	4949	4951	4967
4972	4977	4979	4995	5000	5005	5007	5023	5028	5033	5035	5053	5058
5063	5065	5079	5084	5089	5091	5114	5119	5124	5126	5140	5145	5150
5152	5171	5176	5181	5183	5197	5202	5207	5209	5227	5232	5237	5239
5252	5257	5262	5264	5279	5283	5288	5290	5306	5310	5315	5317	5332
5336	5341	5343	5357	5361	5366	5368	5383	5387	5392	5394	5409	5413
5418	5420	5435	5439	5444	5446	5461	5465	5470	5472	5487	5491	5496
5498	5513	5517	5522	5524	5538	5542	5547	5549	5564	5568	5573	5575
5590	5594	5599	5601	5616	5620	5625	5627	5642	5646	5651	5653	5667
5671	5676	5678	5693	5697	5702	5704	5719	5723	5728	5730	5745	5749
5754	5756	5771	5775	5780	5782	5797	5801	5806	5808	5823	5827	5832
5834	5849	5853	5858	5860	5875	5879	5884	5886	5902	5907	5912	5914
5931	5936	5941	5943	5963	5968	5973	5975	5995	6000	6005	6007	6022
6027	6032	6034	6049	6054	6059	6061	6076	6081	6086	6088	6103	6108
6113	6115	6130	6135	6140	6142	6157	6162	6167	6169	6183	6188	6193
6195	6210	6215	6220	6222	6237	6242	6247	6249	6264	6269	6274	6276
6291	6296	6301	6303	6318	6323	6328	6330	6345	6350	6355	6357	6372
6377	6382	6384	6399	6404	6409	6411	6426	6431	6436	6438	6453	6458
6463	6465	6480	6485	6490	6492	6507	6512	6517	6519	6534	6539	6544
6546	6561	6565	6570	6572	6587	6591	6596	6598	6613	6617	6622	6624
6639	6643	6648	6650	6665	6669	6674	6676	6691	6696	6701	6703	6713
6718	6723	6725	6739	6744	6749	6751	6760	6765	6770	6772	6787	6792
6797	6799	6813	6818	6823	6825	6839	6844	6849	6851	6860	6865	6870
6872	6888	6892	6897	6899	6915	6919	6924	6926	6939	6943	6948	6950
6963	6967	6972	6974	6988	6992	6997	6999	7012	7016	7021	7023	7037
7041	7046	7048	7056	7060	7065	7067	7076	7080	7085	7087	7096	7101
7106	7108	7117	7122	7127	7129	7147	7151	7156	7158	7176	7180	7185
7187	7205	7209	7214	7216	7234	7238	7243	7245	7263	7267	7272	7274
7292	7296	7301	7303	7321	7325	7330	7332	7350	7354	7359	7361	7371
7376	7381	7383	7393	7398	7403	7405	7415	7420	7425	7427	7437	7442
7447	7449	7464	7468	7473	7475	7491	7495	7500	7502	7518	7522	7527
7529	7545	7549	7554	7556	7573	7578	7583	7585	7602	7607	7612	7614
7635	7640	7645	7647	7664	7669	7674	7676	7692	7696	7701	7703	7719
7723	7728	7730	7746	7750	7755	7757	7773	7777	7782	7784	7800	7804
7809	7811	7828	7833	7838	7840	7857	7862	7867	7869	7885	7890	7895
7897	7913	7917	7922	7924	7940	7944	7949	7951	7967	7971	7976	7978
7996	8001	8006	8008	8027	8032	8037	8039	8054	8058	8063	8065	8080
8084	8089	8091	8106	8110	8115	8117	8132	8136	8141	8143	8152	8157
8162	8164	8173	8178	8183	8185	8204	8210	8215	8217	8236	8242	8247
8249	8268	8274	8279	8281	8300	8306	8311	8313	8337	8343	8348	8350
8374	8380	8385	8387	8411	8417	8422	8424	8448	8454	8459	8461	8485
8491	8496	8498	8522	8528	8533	8535	8563	8569	8574	8576	8600	8606
8611	8613	8632	8638	8643	8645	8664	8670	8675	8677	8696	8702	8707
8709	8728	8734	8739	8741	8750	8755	8760	8762	8771	8776	8781	8783
8792	8797	8802	8804	8813	8818	8823	8825	8834	8840	8845	8847	8856
8861	8866	8868	8884	8888	8893	8895	8912	8916	8921	8923	8932	8936
8941	8943	8951	8955	8960	8962	8975	8979	8984	8986	8999	9003	9008

9010#	9019	9024	9029	9031#	9047	9051	9056	9058#	9073	9078	9083	9085#
9100	9105	9110	9112#	9127	9132	9137	9139#	9149	9154	9159	9161#	9171
9176	9181	9183#	9193	9198	9203	9205#	9215	9220	9225	9227#	9237	9242
9247	9249#	9259	9264	9269	9271#	9281	9286	9291	9293#	9309	9313	9318
9320#	9340	9344	9349	9351#	9367	9371	9376	9378#	9394	9398	9403	9405#
9421	9426	9431	9433#	9449	9453	9458	9460#	9476	9480	9485	9487#	9503
9507	9512	9514#	9529	9533	9538	9540#	9557	9562	9567	9569#	9586	9591
9596	9598#	9615	9620	9625	9627#	9644	9649	9654	9656#	9673	9678	9683
9685#	9702	9706	9711	9713#	9730	9735	9740	9742#	9759	9764	9769	9771#
9787	9792	9797	9799#	9815	9819	9824	9826#	9842	9846	9851	9853#	9870
9874	9879	9881#	9897	9901	9906	9908#	9924	9928	9933	9935#	9951	9954
9959	9961#	9977	9981	9986	9988#	10004	10008	10013	10015#	10031	10035	10040
10042#	10059	10064	10069	10071#	10088	10093	10098	10100#	10118	10123	10128	10130#
10148	10153	10158	10160#	10178	10183	10188	10190#	10212	10217	10222	10224#	10242
10247	10252	10254#	10272	10277	10282	10284#	10301	10306	10311	10313#	10321	10325
10330	10332#	10341	10346	10351	10353#	10363	10368	10373	10375#	10385	10390	10395
10397#	10407	10412	10417	10419#	10429	10434	10439	10441#	10451	10456	10461	10463#
10473	10478	10483	10485#	10494	10499	10504	10506#	10516	10521	10526	10528#	10537
10542	10547	10549#	10559	10564	10569	10571#	10577	10582	10586	10591	10593#	10599
10604	10608	10613	10615#	10621	10631	10635	10640	10642#	10648	10653	10657	10662
10664#	10670	10680	10683	10689	10694	10696#	10702	10707	10710	10715	10720	10722#
10728	10732	10742	10746	10751	10753#	10759	10764	10768	10773	10775#	10781	10791
10794	10798	10803	10805#	10811	10816	10819	10824	10829	10831#	10837	10842	10845
10850	10855	10857#	10863	10868	10871	10876	10881	10883#	10889	10892	10897	10900
10906	10911	10913#	10919	10922	10927	10930	10936	10941	10943#	10955	10964	10969
10971#	10995	11000	11002#	11024	11029	11031#	11049	11054	11056#	11068	11078	11083
11085#	11097	11111	11116	11118#	11133	11143	11148	11150#	11162	11176	11181	11183#
11195	11204	11209	11211#	11224	11238	11243	11245#	11249	11254	11259	11260#	11275
11278	11281	11286	11288#	11296	11300	11305	11307#	11315	11319	11324	11326#	11334
11338	11343	11344#	11353	11357	11362	11364#	11390	11398	11403	11405#	11427	11435
11440	11442#	11464	11472	11477	11479#	11491	11499	11511	11520	11532	11537	11539#
11551	11559	11571	11580	11592	11597	11599#	11616	11621	11623#	11629	11634	11639
11641#	11647	11652	11657	11659#	11682	11687	11689#	11714	11719	11721#	11747	11752
11754#	11780	11785	11787#	11837	11842	11844#	11894	11899	11901#	11925	11930	11932#
11956	11961	11963#	11987	11992	11994#	12018	12023	12025#	12049	12054	12056#	12085
12090	12092#	12117	12125	12127#	12161	12170	12172#	12248	12253	12255#	12268	12273
12275#	12308	12313	12315#	12346	12351	12353#	12371	12376	12378#	12396	12401	12403#
12421	12426	12428#	12445	12450	12452#	12479	12484	12489	12491#	12517	12522	12527
12529#	12556	12574	12579	12581#	12625	12634	12639	12641#	12685	12694	12699	12701#
12724	12729	12731#	12756	12761	12763#	12788	12793	12795#	12811	12816	12818#	12834
12839	12841#	12856	12861	12863#	12887	12894	12896#	12910	12924	12929	12931#	12945
12959	12964	12966#	12980	12994	12999	13000#	13020	13025	13026#	13047	13052	13053#
13074	13079	13080#	13104	13109	13110#	13144	13151	13152#	13174	13179	13181#	13189
13194	13196#	13214	13219	13221#	13230	13235	13237#	13247	13252	13254#	13267	13301
13302#	13327	13361	13362#	13387	13421	13422#	13447	13481	13482#	13507	13527	13528#
13543	13552	13571	13572#	13587	13596	13601	13602#	13622	13626	13631	13632#	13653
13657	13662	13663#	13689	13693	13698	13699#	13725	13729	13734	13735#	13761	13765
13770	13771#	13797	13801	13806	13807#	13828	13832	13837	13838#	13852	13856	13861
13862#	13881	13885	13890	13891#	13910	13914	13919	13920#	13939	13943	13948	13949#
13961	13965	13974	13976#	14004	14011	14016	14018#	14032	14035	14040	14042#	14057
14060	14065	14067#	14081	14084	14089	14090#	14109	14112	14117	14118#	14138	14141
14146	14147#	14170	14186	14194	14196#	14224	14237	14239#	14275	14288	14290#	14321
14332	14334#	14351	14369	14371#	14438	14447	14449#	14499	14510	14512#	14569	14577
14579#	14606	14618	14620#	14683	14694	14696#	14722	14735	14737#	14844	14860	14862#
14931	14935	14937#	14968									
210#	15478#	15489										

POP	134#	15043	15044	15614	15615										
PREERR	15289#	15305													
PRENEW	570#	14193	14236	14287	14331	14368	14446	14509	14576	14617	14693	14734	14859	14934	
PRESCO	15207#	15223													
PUSH	134#	15021	15027	15575	15577	15598									
REPORT	134#														
SCOPE	29#	3619	3944	3958	3972	3992	4014	4034	4056	4070	4085	4100	4115	4130	4144
	4159	4174	4189	4215	4236	4262	4279	4331	4348	4374	4392	4418	4444	4496	4523
	4549	4575	4601	4628	4655	4682	4709	4735	4761	4787	4813	4839	4865	4892	4920
	4948	4976	5004	5032	5062	5088	5123	5149	5180	5206	5236	5261	5287	5314	5340
	5365	5391	5417	5443	5469	5495	5521	5546	5572	5598	5624	5650	5675	5701	5727
	5753	5779	5805	5831	5857	5883	5911	5940	5972	6004	6031	6058	6085	6112	6139
	6166	6192	6219	6246	6273	6300	6327	6354	6381	6408	6435	6462	6489	6516	6543
	6569	6595	6621	6647	6673	6700	6722	6748	6769	6796	6822	6848	6869	6896	6923
	6947	6971	6996	7020	7045	7064	7084	7105	7126	7155	7184	7213	7242	7271	7300
	7329	7358	7380	7402	7424	7446	7472	7499	7526	7553	7582	7611	7644	7673	7700
	7727	7754	7781	7808	7837	7866	7894	7921	7948	7975	8005	8036	8062	8088	8114
	8140	8161	8182	8214	8246	8278	8310	8347	8384	8421	8458	8495	8532	8573	8610
	8642	8674	8706	8738	8759	8780	8801	8822	8844	8865	8892	8920	8940	8959	8983
	9007	9028	9055	9082	9109	9136	9158	9180	9202	9224	9246	9268	9290	9317	9348
	9375	9402	9430	9457	9484	9511	9537	9566	9595	9624	9653	9682	9710	9739	9768
	9796	9823	9850	9878	9905	9932	9958	9985	10012	10039	10068	10097	10127	10157	10187
	10221	10251	10281	10310	10329	10350	10372	10394	10416	10438	10460	10482	10503	10525	10546
	10568	10590	10612	10639	10661	10693	10719	10750	10772	10802	10828	10854	10880	10910	10940
	10968	10999	11028	11053	11082	11115	11147	11180	11208	11242	11258	11285	11304	11323	11342
	11361	11402	11439	11476	11536	11596	11620	11638	11656	11686	11718	11751	11784	11841	11898
	11929	11960	11991	12022	12053	12089	12124	12169	12252	12272	12312	12350	12375	12400	12425
	12449	12488	12526	12578	12638	12698	12728	12760	12792	12815	12838	12860	12893	12928	12963
	12998	13024	13051	13078	13108	13135	13178	13193	13218	13234	13251	13300	13360	13420	13480
	13526	13570	13600	13630	13661	13697	13733	13769	13805	13836	13860	13889	13918	13947	13973
	14015	14039	14064	14088	14116	14145	14195	14238	14289	14333	14370	14448	14511	14578	14619
	14695	14736	14861	14936	14982										
SETPRI	134#														
SETTRA	15649#	15658	15659	15660											
SETUP	134#														
SKIP	134#	1243	1258	1270	1284	1299	1312	1331	1347	1363	1385	1407	1425	1443	1462
	1481	1499	1521	1544	1561	1583	1609	1638	1658	1676	1692	1710	1729	1750	1771
	1791	1812	1834	1857	1880	1903	1941	1960	1978	2001	2018	2051	2068	2093	2118
	2137	2162	2184	2201	2218	2244	2272	2290	2308	2326	2343	2383	2401	2420	2452
	2488	2525	2562	2599	2626	2651	2676	2701	2727	2753	2779	2805	2828	2850	2941
	2959	2977	2999	3015	3042	3057	3070	3084	3100	3115	3129	3144	3158	3173	3189
	3204	3219	3234	3247	3263	3278	3293	3307	3321	3337	3352	3367	3393	3411	3428
	3464	3485	3509	3540	3561	3601	3662	3706	3726	3770	3816	3836	3854	3950	3963
	3984	4006	4026	4048	4062	4076	4091	4106	4122	4136	4150	4165	4180	4207	4228
	4254	4271	4323	4340	4366	4384	4410	4436	4488	4515	4541	4567	4593	4620	4647
	4674	4701	4727	4753	4779	4805	4831	4857	4884	4911	4939	4967	4995	5023	5053
	5079	5114	5140	5171	5197	5227	5252	5279	5306	5332	5357	5383	5409	5435	5461
	5487	5513	5538	5564	5590	5616	5642	5667	5693	5719	5745	5771	5797	5823	5849
	5875	5902	5931	5963	5995	6022	6049	6076	6103	6130	6157	6183	6210	6237	6264
	6291	6318	6345	6372	6399	6426	6453	6480	6507	6534	6561	6587	6613	6639	6665
	6691	6713	6739	6760	6787	6813	6839	6860	6888	6915	6939	6963	6988	7012	7037
	7056	7076	7096	7117	7147	7176	7205	7234	7263	7292	7321	7350	7371	7393	7415
	7437	7464	7491	7518	7545	7573	7602	7635	7664	7692	7719	7746	7773	7800	7828
	7857	7885	7913	7940	7967	7996	8027	8054	8080	8106	8132	8152	8173	8204	8236
	8268	8300	8337	8374	8411	8448	8485	8522	8563	8600	8632	8664	8696	8728	8750
	8771	8792	8813	8834	8856	8884	8912	8932	8951	8975	8999	9019	9047	9073	9100

9127	9149	9171	9193	9215	9237	9259	9281	9309	9340	9367	9394	9421	9449	9476
9503	9529	9557	9586	9615	9644	9673	9702	9730	9759	9787	9815	9842	9870	9897
9924	9951	9977	10004	10031	10059	10088	10118	10148	10178	10212	10242	10272	10301	10321
10341	10363	10385	10407	10429	10451	10473	10494	10516	10537	10559	10577	10582	10599	10604
10621	10631	10648	10653	10670	10680	10683	10702	10707	10710	10728	10732	10742	10759	10764
10781	10791	10794	10811	10816	10819	10837	10842	10845	10863	10868	10871	10889	10892	10897
10900	10919	10922	10927	10930	10955	11068	11097	11133	11162	11195	11224	11249	11275	11278
11296	11315	11334	11353	11390	11427	11464	11491	11499	11511	11520	11551	11559	11571	11580
11629	11647	12479	12517	12556	12625	12685	12910	12945	12980	13543	13587	13622	13653	13689
13725	13761	13797	13828	13852	13881	13910	13939	13961	14004	14032	14057	14081	14109	14138
14170														
SLASH	134#													
SPACE	134#													
STARS	134#													
1277	1289	1291	1304	1306	1318	1320	1335	1337	1351	1353	1368	1370	1390	1392
1412	1414	1430	1432	1448	1450	1467	1469	1486	1488	1503	1505	1526	1528	1549
1551	1566	1568	1588	1590	1614	1616	1644	1646	1663	1665	1681	1683	1697	1699
1715	1717	1734	1736	1755	1757	1775	1777	1796	1798	1817	1819	1839	1841	1863
1865	1886	1888	1909	1911	1930	1932	1946	1948	1965	1967	1983	1985	2006	2008
2023	2025	2038	2040	2056	2058	2073	2075	2098	2100	2123	2125	2142	2144	2166
2168	2189	2191	2205	2207	2223	2225	2249	2251	2277	2279	2295	2297	2313	2315
2331	2333	2348	2350	2372	2374	2388	2390	2406	2408	2425	2427	2457	2459	2494
2496	2531	2533	2568	2570	2605	2607	2630	2632	2655	2657	2680	2682	2706	2708
2732	2734	2758	2760	2784	2786	2810	2812	2832	2834	2855	2857	2891	2893	2927
2929	2945	2947	2963	2965	2981	2983	3004	3006	3020	3022	3048	3050	3062	3064
3076	3078	3090	3092	3105	3107	3120	3122	3135	3137	3149	3151	3164	3166	3179
3181	3194	3196	3209	3211	3224	3226	3239	3241	3253	3255	3268	3270	3283	3285
3298	3300	3313	3315	3327	3329	3342	3344	3357	3359	3372	3374	3398	3400	3416
3418	3433	3435	3469	3471	3490	3492	3514	3516	3545	3547	3567	3569	3607	3609
3628	3630	3668	3670	3691	3693	3711	3713	3730	3732	3754	3756	3775	3777	3803
3805	3824	3826	3842	3844	3860	3870	3940	3942	3954	3956	3968	3970	3988	3990
4010	4012	4030	4032	4052	4054	4066	4068	4081	4083	4096	4098	4111	4113	4126
4128	4140	4142	4155	4157	4170	4172	4185	4187	4211	4213	4232	4234	4258	4260
4275	4277	4327	4329	4344	4346	4370	4372	4388	4390	4414	4416	4440	4442	4492
4494	4519	4521	4545	4547	4571	4573	4597	4599	4624	4626	4651	4653	4678	4680
4705	4707	4731	4733	4757	4759	4783	4785	4809	4811	4835	4837	4861	4863	4888
4890	4916	4918	4944	4946	4972	4974	5000	5002	5028	5030	5058	5060	5084	5086
5119	5121	5145	5147	5176	5178	5202	5204	5232	5234	5257	5259	5283	5285	5310
5312	5336	5338	5361	5363	5387	5389	5413	5415	5439	5441	5465	5467	5491	5493
5517	5519	5542	5544	5568	5570	5594	5596	5620	5622	5646	5648	5671	5673	5697
5699	5723	5725	5749	5751	5775	5777	5801	5803	5827	5829	5853	5855	5879	5881
5907	5909	5936	5938	5968	5970	6000	6002	6027	6029	6054	6056	6081	6083	6108
6110	6135	6137	6162	6164	6188	6190	6215	6217	6242	6244	6269	6271	6296	6298
6323	6325	6350	6352	6377	6379	6404	6406	6431	6433	6458	6460	6485	6487	6512
6514	6539	6541	6565	6567	6591	6593	6617	6619	6643	6645	6669	6671	6696	6698
6718	6720	6744	6746	6765	6767	6792	6794	6818	6820	6844	6846	6865	6867	6892
6894	6919	6921	6943	6945	6967	6969	6992	6994	7016	7018	7041	7043	7060	7062
7080	7082	7101	7103	7122	7124	7151	7153	7180	7182	7209	7211	7238	7240	7267
7269	7296	7298	7325	7327	7354	7356	7376	7378	7398	7400	7420	7422	7442	7444
7468	7470	7495	7497	7522	7524	7549	7551	7578	7580	7607	7609	7640	7642	7669
7671	7696	7698	7723	7725	7750	7752	7777	7779	7804	7806	7833	7835	7862	7864
7890	7892	7917	7919	7944	7946	7971	7973	8001	8003	8032	8034	8058	8060	8084
8086	8110	8112	8136	8138	8157	8159	8178	8180	8210	8212	8242	8244	8274	8276
8306	8308	8343	8345	8380	8382	8417	8419	8454	8456	8491	8493	8528	8530	8569
8571	8606	8608	8638	8640	8670	8672	8702	8704	8734	8736	8755	8757	8776	8778
8797	8799	8818	8820	8840	8842	8861	8863	8888	8890	8916	8918	8936	8938	8955

8957	8979	8981	9003	9005	9024	9026	9051	9053	9078	9080	9105	9107	9132	9134
9154	9156	9176	9178	9198	9200	9220	9222	9242	9244	9264	9266	9286	9288	9313
9315	9344	9346	9371	9373	9398	9400	9426	9428	9453	9455	9480	9482	9507	9509
9533	9535	9562	9564	9591	9593	9620	9622	9649	9651	9678	9680	9706	9708	9735
9737	9764	9766	9792	9794	9819	9821	9846	9848	9874	9876	9901	9903	9928	9930
9954	9956	9981	9983	10008	10010	10035	10037	10064	10066	10093	10095	10123	10125	10153
10155	10183	10185	10217	10219	10247	10249	10277	10279	10306	10308	10325	10327	10346	10348
10368	10370	10390	10392	10412	10414	10434	10436	10456	10458	10478	10480	10499	10501	10521
10523	10542	10544	10564	10566	10586	10588	10608	10610	10635	10637	10657	10659	10689	10691
10715	10717	10746	10748	10768	10770	10798	10800	10824	10826	10850	10852	10876	10878	10906
10908	10936	10938	10964	10966	10995	10997	11024	11026	11049	11051	11078	11080	11111	11113
11143	11145	11176	11178	11204	11206	11238	11240	11254	11256	11281	11283	11300	11302	11319
11321	11338	11340	11357	11359	11398	11400	11435	11437	11472	11474	11532	11534	11592	11594
11616	11618	11634	11636	11652	11654	11682	11684	11714	11716	11747	11749	11780	11782	11837
11839	11894	11896	11925	11927	11956	11958	11987	11989	12018	12020	12049	12051	12085	12087
12117	12122	12161	12167	12248	12250	12268	12270	12308	12310	12346	12348	12371	12373	12396
12398	12421	12423	12445	12447	12484	12486	12522	12524	12574	12576	12634	12636	12694	12696
12724	12726	12756	12758	12788	12790	12811	12813	12834	12836	12856	12858	12887	12891	12924
12926	12959	12961	12994	12996	13020	13022	13047	13049	13074	13076	13104	13106	13144	13149
13174	13176	13189	13191	13214	13216	13230	13232	13247	13249	13267	13298	13327	13358	13387
13418	13447	13478	13507	13524	13552	13568	13596	13598	13626	13628	13657	13659	13693	13695
13729	13731	13765	13767	13801	13803	13832	13834	13856	13858	13885	13887	13914	13916	13943
13945	13965	13971	14011	14013	14035	14037	14060	14062	14084	14086	14112	14114	14141	14143
14186	14192	14224	14235	14275	14286	14321	14330	14351	14367	14438	14445	14499	14508	14569
14575	14606	14616	14683	14692	14722	14733	14844	14858	14931	14933	14976	15017	15033	15077
15211	15293	15356	15412	15492	15570	15628								

SWRSU TESTNO	134# 570#	1240	1253	1267	1279	1293	1308	1322	1339	1355	1372	1394	1416	1434	1452
	1471	1490	1507	1530	1553	1570	1592	1618	1648	1667	1685	1701	1719	1738	1759
	1779	1800	1821	1843	1867	1890	1913	1934	1950	1969	1987	2010	2027	2042	2060
	2077	2102	2127	2146	2170	2193	2209	2227	2253	2281	2299	2317	2335	2352	2376
	2392	2410	2429	2461	2498	2535	2572	2609	2634	2659	2684	2710	2736	2762	2788
	2814	2836	2859	2895	2931	2949	2967	2985	3008	3024	3052	3066	3080	3094	3109
	3124	3139	3153	3168	3183	3198	3213	3228	3243	3257	3272	3287	3302	3317	3331
	3346	3361	3376	3402	3420	3437	3473	3494	3518	3549	3571	3611	3632	3672	3695
	3715	3734	3758	3779	3807	3828	3846	3872	3945	3959	3973	3993	4015	4035	4057
	4071	4086	4101	4116	4131	4145	4160	4175	4190	4216	4237	4263	4280	4332	4349
	4375	4393	4419	4445	4497	4524	4550	4576	4602	4629	4656	4683	4710	4736	4762
	4788	4814	4840	4866	4893	4921	4949	4977	5005	5033	5063	5089	5124	5150	5181
	5207	5237	5262	5288	5315	5341	5366	5392	5418	5444	5470	5496	5522	5547	5573
	5599	5625	5651	5676	5702	5728	5754	5780	5806	5832	5858	5884	5912	5941	5973
	6005	6032	6059	6086	6113	6140	6167	6193	6220	6247	6274	6301	6328	6355	6382
	6409	6436	6463	6490	6517	6544	6570	6596	6622	6648	6674	6701	6723	6749	6770
	6797	6823	6849	6870	6897	6924	6948	6972	6997	7021	7046	7065	7085	7106	7127
	7156	7185	7214	7243	7272	7301	7330	7359	7381	7403	7425	7447	7473	7500	7527
	7554	7583	7612	7645	7674	7701	7728	7755	7782	7809	7838	7867	7895	7922	7949
	7976	8006	8037	8063	8089	8115	8141	8162	8183	8215	8247	8279	8311	8348	8385
	8422	8459	8496	8533	8574	8611	8643	8675	8707	8739	8760	8781	8802	8823	8845
	8866	8893	8921	8941	8960	8984	9008	9029	9056	9083	9110	9137	9159	9181	9203
	9225	9247	9269	9291	9318	9349	9376	9403	9431	9458	9485	9512	9538	9567	9596
	9625	9654	9683	9711	9740	9769	9797	9824	9851	9879	9906	9933	9959	9986	10013
	10040	10069	10098	10128	10158	10188	10222	10252	10282	10311	10330	10351	10373	10395	10417
	10439	10461	10483	10504	10526	10547	10569	10591	10613	10640	10662	10694	10720	10751	10773
	10803	10829	10855	10881	10911	10941	10969	11000	11029	11054	11083	11116	11148	11181	11209
	11243	11259	11286	11305	11324	11343	11362	11403	11440	11477	11537	11597	11621	11639	11657
	11687	11719	11752	11785	11842	11899	11930	11961	11992	12023	12054	12090	12125	12170	12253

	12273	12313	12351	12376	12401	12426	12450	12489	12527	12579	12639	12699	12729	12761	12793
	12816	12839	12861	12894	12929	12964	12999	13025	13052	13079	13109	13151	13179	13194	13219
	13235	13252	13301	13361	13421	13481	13527	13571	13601	13631	13662	13698	13734	13770	13806
	13837	13861	13890	13919	13948	13974	14016	14040	14065	14089	14117	14146	14194	14237	14288
	14332	14369	14447	14510	14577	14618	14694	14735	14860	14935					
TRMTRP	15649#														
TYPBIN	134#														
TYPDEC	134#														
TYPNAM	134#														
TYPNUM	134#														
TYPOCS	134#														
TYPOCT	134#	15368	15392												
TYPTXT	134#														
UPCODE	15015#	15040													
YESCOP	570#	3943	3957	3971	3991	4013	4033	4055	4069	4084	4099	4114	4129	4143	4158
	4173	4188	4214	4235	4261	4278	4330	4347	4373	4391	4417	4443	4495	4522	4548
	4574	4600	4627	4654	4681	4708	4734	4760	4786	4812	4838	4864	4891	4919	4947
	4975	5003	5031	5061	5087	5122	5148	5179	5205	5235	5260	5286	5313	5339	5364
	5390	5416	5442	5468	5494	5520	5545	5571	5597	5623	5649	5674	5700	5726	5752
	5778	5804	5830	5856	5882	5910	5939	5971	6003	6030	6057	6084	6111	6138	6165
	6191	6218	6245	6272	6299	6326	6353	6380	6407	6434	6461	6488	6515	6542	6568
	6594	6620	6646	6672	6699	6721	6747	6768	6795	6821	6847	6868	6895	6922	6946
	6970	6995	7019	7044	7063	7083	7104	7125	7154	7183	7212	7241	7270	7299	7328
	7357	7379	7401	7423	7445	7471	7498	7525	7552	7581	7610	7643	7672	7699	7726
	7753	7780	7807	7836	7865	7893	7920	7947	7974	8004	8035	8061	8087	8113	8139
	8160	8181	8213	8245	8277	8309	8346	8383	8420	8457	8494	8531	8572	8609	8641
	8673	8705	8737	8758	8779	8800	8821	8843	8864	8891	8919	8939	8958	8982	9006
	9027	9054	9081	9108	9135	9157	9179	9201	9223	9245	9267	9289	9316	9347	9374
	9401	9429	9456	9483	9510	9536	9565	9594	9623	9652	9681	9709	9738	9767	9795
	9822	9849	9877	9904	9931	9957	9984	10011	10038	10067	10096	10126	10156	10186	10220
	10250	10280	10309	10328	10349	10371	10393	10415	10437	10459	10481	10502	10524	10545	10567
	10589	10611	10638	10660	10692	10718	10749	10771	10801	10827	10853	10879	10909	10939	10967
	10998	11027	11052	11081	11114	11146	11179	11207	11241	11284	11303	11322	11360	11401	11438
	11475	11535	11595	11619	11637	11655	11685	11717	11750	11783	11840	11897	11928	11959	11990
	12021	12052	12088	12123	12168	12251	12271	12311	12349	12374	12399	12424	12448	12487	12525
	12577	12637	12697	12727	12759	12791	12814	12837	12859	12892	12927	12962	13177	13192	13217
	13233	13250	13972	14014	14038	14063									
SSCMRE	178#	217	218	219	220	221	222								
SSCHTM	178#	223	224	225	226	227									
SSESCA	134#														
SSNEWT	134#	1236	1249	1263	1275	1289	1304	1318	1335	1351	1368	1390	1412	1430	1448
	1467	1486	1503	1526	1549	1566	1588	1614	1644	1663	1681	1697	1715	1734	1755
	1775	1796	1817	1839	1863	1886	1909	1930	1946	1965	1983	2006	2023	2038	2056
	2073	2098	2123	2142	2166	2189	2205	2223	2249	2277	2295	2313	2331	2348	2372
	2388	2406	2425	2457	2494	2531	2568	2605	2630	2655	2680	2706	2732	2758	2784
	2810	2832	2855	2891	2927	2945	2963	2981	3004	3020	3048	3062	3076	3090	3105
	3120	3135	3149	3164	3179	3194	3209	3224	3239	3253	3268	3283	3298	3313	3327
	3342	3357	3372	3398	3416	3433	3469	3490	3514	3545	3567	3607	3628	3668	3691
	3711	3730	3754	3775	3803	3824	3842	3860	3940	3954	3968	3988	4010	4030	4052
	4066	4081	4096	4111	4126	4140	4155	4170	4185	4211	4232	4258	4275	4327	4344
	4370	4388	4414	4440	4492	4519	4545	4571	4597	4624	4651	4678	4705	4731	4757
	4783	4809	4835	4861	4888	4916	4944	4972	5000	5028	5058	5084	5119	5145	5176
	5202	5232	5257	5283	5310	5336	5361	5387	5413	5439	5465	5491	5517	5542	5568
	5594	5620	5646	5671	5697	5723	5749	5775	5801	5827	5853	5879	5907	5936	5968
	6000	6027	6054	6081	6108	6135	6162	6188	6215	6242	6269	6296	6323	6350	6377
	6404	6431	6458	6485	6512	6539	6565	6591	6617	6643	6669	6696	6718	6744	6765

H12

MAINDEC-11-DQKDA-A KD11-K BASIC LOGIC TESTS MACY11 27(1006) 08-FEB-77 16:23 PAGE 336
DQKDAA.P11 08-FEB-77 16:17 CROSS REFERENCE TABLE -- MACRO NAMES

.SAPT8	18	2348
.SAPTH	18	146
.SAPTY	18	15568
.SCATC	18	134
.SCHTA	18	178
.SEOP	18	14974
.SERRO	18	15291
.SERRT	18	15354
.SPOWE	18	15015
.SSCOP	18	15209
.STRAP	18	15626
.STYPD	18	
.STYPE	18	15410
.STYPO	18	15490

. ABS. 067770 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DQKDA A DSKZ:DQKDA A/CRF/SOL/DS:ERFZ=DQKDA A
RUN-TIME: 61 64 5 SECONDS
RUN-TIME RATIO: 349/132=2.6
CORE USED: 28K (56 PAGES)