PDP-11/70

CPU DIAGNOSTIC PART 1

MD-11-DEKBA-B

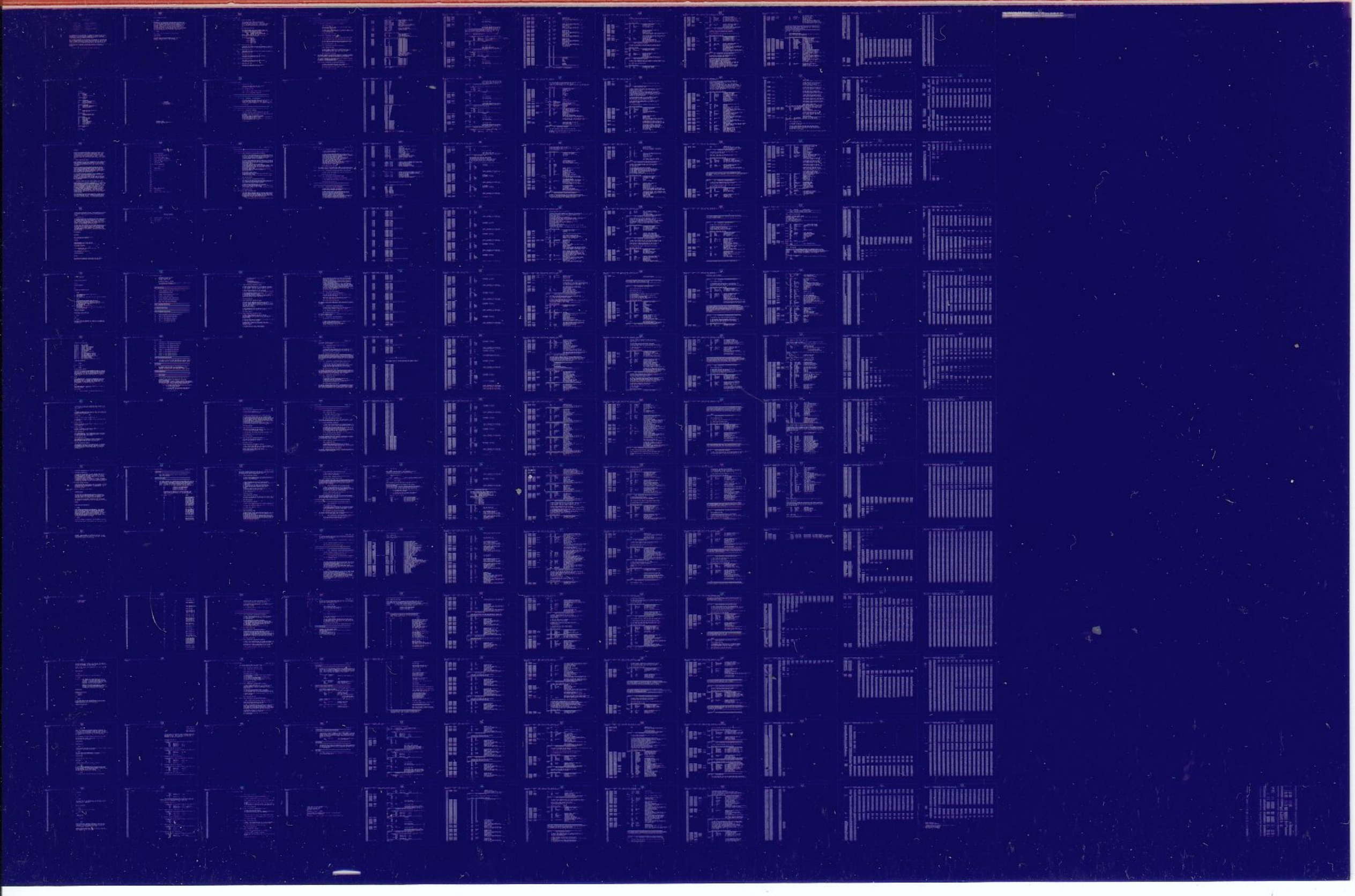
EP-DEKBA-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



```
                                .REM    !
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DEKBA-B-D
PRODUCT NAME:	PDP-11/70 CPU DIAGNOSTIC PART 1
DATE CREATED:	21-AUG-75
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHORS:	DON MONROE

CONTENTS

1. ABSTRACT

DEKBA/B ARE PROGRAMS DESIGNED TO DETECT AND REPORT LOGIC FAULTS IN THE PDP 11/70 CENTRAL PROCESSING UNIT. THEY CONSISTS OF 210(8) 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 TWO STAND-ALONE PROGRAMS AS DESCRIBED BELOW.

A. BASIC INSTRUCTION TESTS

DEKBA CONSISTS OF A LOGICALLY SEQUENCED SET OF INSTRUCTION TESTS DESIGNED TO VERIFY THE INTEGRITY OF THOSE INSTRUCTIONS AND LOGIC OPERATIONS USED BY THE UTILITY ROUTINES THAT PROVIDE ERROR REPORTING AND SCOPE LOOPING FACILITIES FOR DEKBB.

ANY FAULT DETECTED IN THIS PROGRAM CAUSES THE PROGRAM TO "HALT" WITH THE CONSOLE ADDRESS LIGHTS INDICATING THE ERROR PROGRAM COUNTER AND THE CONSOLE DATA LIGHTS SHOWING THE TEST NUMBER (FOR TESTS 24 AND ABOVE). ADDITIONAL FAULT IDENTIFICATION INFORMATION IS AVAILABLE IN THE PROGRAM ANNOTATION FOR THE FAILING TEST.

IF THE PROGRAM HALTS AT LOCATION 6 OR 12 (ADDRESS LIGHTS OF 10 OR 14) THE PROGRAM ANNOTATION FOR THE INDICATED TEST NUMBER, SHOULD GIVE A CLUE TO THE PROBLEM. TO LOOP ON THE ERROR THE HALT MUST BE REPLACED BY THE OCTAL CODE SHOWN IN THE COMMENT FIELD OF THE HALT AND THE PROGRAM RESTARTED AT 200, OR THE START ADDRESS OF THAT PARTICULAR TEST.

B. ADVANCED INSTRUCTION AND MISCELLANEOUS LOGIC TESTS

DEKBB CONSISTS OF A LOGICALLY SEQUENCED SET OF INSTRUCTION TESTS FOLLOWED BY A SET OF MISCELLANEOUS LOGIC TESTS. THE INSTRUCTION TESTS COMPLETE THE TEST OF THE PDP 11/70 INSTRUCTION REPERTOIRE. THE LOGIC TESTS VERIFY SUCH THINGS AS: 1) THE INTERNAL REGISTERS; 2) REGISTERS SET 1; 3) INTERNAL INTERRUPTS; 4) BUS REQUEST LEVELS 4, 5, AND 6; 5) INTERNAL TRAPS, AND ABORTS; 6) OUTER MODE SELECTION; AND 7) EXTERNAL TRAPS AND ABORTS. EACH TEST IN THIS PROGRAM 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" THAT 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 ITS NORMAL SEQUENCE UNLESS MOD-
IFIED BY THE USER ACTIVATING THE "HALT ON ERROR" SWITCH
OPTION.

C. IMPORTANT NOTE

THE PROGRAM ANNOTATION IN DEKBA AND THE TYPED ERROR REPORTS
IN DEKBB ARE BASED UPON THE KNOWLEDGE THAT ALL PREVIOUS
TESTS WERE FAULTLESS AND THAT THERE IS ONLY ONE SINGLE
POINT FAILURE IN THE PROCESSOR. THIS MEANS THAT IF EITHER
PROGRAM, OR THE PROGRAMS THEMSELVES, ARE NOT RUN IN SE-
QUENCE, THE ERROR MESSAGE MAY NOT BE VALID.

ALTHOUGH EACH ERROR ANNOTATION AND TYPED MESSAGE CONCLUSION
HAS BEEN PROVEN BY PHYSICAL FAULT INSERTION (ONE SIGNAL
STUCK LOW), IT IS HUMANLY IMPOSSIBLE TO GUARANTEE THAT
THE ERROR REPORT IS 100% CORRECT. THE SOLE FUNCTION OF
THE ERROR REPORT IS TO DIRECT THE USER TO THE MOST PROBABLE
AREA OF FAILURE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP 11/70 CPU WITH OPERATORS CONSOLE
LA30 OR EQUIVALENT TERMINAL

2.2 STORAGE

DEKBA REQUIRES 16K TO LOAD AND RUN
DEKBB REQUIRES 16K TO LOAD AND 32K TO RUN

2.3 PRELIMINARY PROGRAMS

DEKBA REQUIRES THAT TWO INSTRUCTIONS WORK:
"BR" AND "HALT"

DEKBB REQUIRES THAT DEKBA RUN

3. LOADING PROCEDURE

3.1 METHOD

BOTH DEKBA AND DEKBB ARE LOADED FROM THE XXDP MEDIA.
REFER TO THE XXDP MANUAL FOR FURTHER INFORMATION.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1

4.2 STARTING ADDRESS

200

4.3 PROGRAM AND OPERATOR ACTION

A. DEKBA

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3)
2. LOAD ADDRESS 200
3. PRESS START
5. THE PROGRAM WILL LOOP UNTIL THE HALT SWITCH IS PRESSED

B. DEKBB

1. LOAD PROGRAM INTO MEMORY (SEE SECTION 3)
2. ENSURE RH CONTROLLER IS ENABLED, IF SW<5>=0
3. IF AN RK05 IS AVAILABLE (AND THERE WAS NO RH) ENSURE AT LEAST ONE DRIVE IS ENABLED; IF SW<5>=0
4. LOAD ADDRESS 200
5. SET SWITCHES (SEE SECTION 5.1)
6. PRESS START
7. THE PROGRAM WILL LOOP AND AN END OF PASS MESSAGE WILL BE TYPED EVERY PASS.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

A. DEKBA

NONE

B. DEKBB

WITH SW<15:0>=0 THE PROGRAM WILL PRINT OUT ON ERRORS AND CONTINUE. "END OF PASS" WILL BE TYPED AT THE COMPLETION OF EACH PASS.

THE SWITCH SETTINGS ARE:

SW<15>=1 ... HALT ON ERROR
SW<14>=1 ... LOOP ON TEST
SW<13>=1 ... INHIBIT ERROR TYPEOUTS
SW<12>=1 ... INHIBIT T BIT TRAPPING
SW<11>=1 ... INHIBIT ITERATIONS
SW<10>=1 ... RING BELL ON ERROR
SW<9> =1 ... LOOP ON ERROR
SW<8> =1 ... LOOP ON TEST IN SW<7:0>
SW<7> =1 ... NO ACTION
SW<6> =1 ... SKIP BUS REQUEST 6 TESTING
SW<5> =1 ... SKIP BUS REQUEST 5 TESTING
SW<4> =1 ... SKIP BUS REQUEST 4 TESTING
SW<3> =1 ... SKIP MEMORY MANAGEMENT TESTING
SW<2> =1 ... SKIP CACHE TESTING
SW<1> =1 ... SKIP MAP BOX TESTING
SW<0> =1 ... SKIP OPERATOR INTERVENTION TESTING

5.2 SUBROUTINE ABSTRACTS

A. DEKBA

SEE 5.2.4 AND 5.2.5

B. DEKBB

5.2.1 SPURIOUS ERROR HANDLER

THIS ROUTINE IS CALLED BY AN UNEXPECTED TRAP TO LOCATION 4 OR 114. IT PRINTS A SHORT MESSAGE FOLLOWED BY THE PROGRAM COUNTER AT THE TIME OF THE TRAP AND THE APPROPRIATE ERROR REGISTER (I.E. THE CPU ERROR REGISTER IN CASES OF A TRAP TO 4 AND THE MEMORY ERROR REGISTER IN CASES OF A TRAP TO 114).

5.2.2 SCOPE

THIS SUBROUTINE CALL (VIA AN IOT INSTRUCTION) IS PLACED BETWEEN EACH TEST. IT RECORDS THE STARTING ADDRESS OF EACH TEST IN LOCATION "$LPADR" AND "$LPERR" AS IT IS BEING ENTERED. IT ALSO CONTROLS TEST ITTERATION, LOOPING, AND SEQUENTIAL FLOW CHECKS (SEE 5.2.8).

5.2.3 ERROR

THIS SUBROUTINE CALL (VIA A EMT INSTRUCTION) IS USED TO REPORT ALL ERRORS. (REFER TO 6)

5.2.4 TRAP CATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM LOCATION 0 TO

LOCATION 776 TO CATCH ANY UNEXPECTED TRAPS (EXCEPT 4 AND 114). THUS, ANY UNEXPECTED TRAPS WILL HALT AT THE DEVICE TRAP VECTOR +2.

5.2.5 TRAP

A NUMBER OF SUBROUTINES ARE CALLED BY THE TRAP INSTRUCTION. FOLLOWING ARE THE CALLS USED AND THE LABEL OF THE STARTING ADDRESS OF THE SUBROUTINES.

5.2.5.1 TYPE ($TYPE)

ROUTINE TO TYPE AN ASCII STRING ON THE TTY.

THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.

5.2.5.2 TYPEOC ($TYPOC)

ROUTINE TO CONVERT A 16-BIT BINARY NUMBER TO A 6-DIGIT OCTAL NUMBER AND TYPE IT.

5.2.5.3 TYPEDS ($TYPDS)

ROUTINE TO CONVERT A 16-BIT BINARY NUMBER TO A 5-DIGIT SIGNED DECIMAL NUMBER AND TYPE IT.

5.2.6 POWER DOWN AND UP

THIS SUBROUTINE CALL (VIA A POWER DOWN) SAVES THE RETURN PC UPON A POWER DOWN. WHEN POWER IS RESTORED A MESSAGE IS TYPED AND THE TEST WILL RESTART.

5.2.7 MONITOR RESTORE (QUIT)

THIS SUBROUTINE IS ENTERED BY TYPING A "CONTROL C" OR WHEN THE END OF PASS IS REACHED AND THE PROGRAM IS RUNNING UNDER A MONITOR. SEE 7.2 FOR DETAILS.

5.2.8 CHECK TEST SEQUENCE (SEQENC)

THIS ROUTINE IS CALLED IN THE SCOPE ROUTINE IT COMPARES THE ADDRESS OF THE SCOPE CALL WITH THE ADDRESS POINTED TO BY $TSTNM IN THE "TEST ADDRESS TABLE". IF THEY DON'T COMPARE, A MESSAGE IS TYPED, INDICATIONG THAT A TEST WAS SKIPPED.

5.2.9 PERIPHERAL DETERMINATOR AND INTERRUPT ENABLE ROUTINES

5.2.9.1 PERIPHERAL DETERMINATOR

THIS ROUTINE IS EXECUTED, IN LINE, BETWEEN TESTS 32 AND 33
OF DEKBB. IT CHECKS THE SYSTEM TO DETERMINE IF ONE OF
FOUR PERIPHERALS IS AVAILABLE (RS04, RP04, TM, OR RK05)
FOR BUS REQUEST LEVEL 5 TESTING AND IF A LINE CLOCK IS
AVAILABLE FOR LEVEL 6 TESTING.
IF A DEVICE IS FOUND, THE ADDRESS OF "INT5SU" (INTERRUPT
5 SUBROUTINE) AND $KW11L/P (LINE CLOCK INTERRUPT SUBROUTINE)
IS PLACED IN LOCATIONS "INTER5" AND "INTER6" RESPECTIVELY.

5.2.9.2 INTERRUPT ENABLE ROUTINES

THESE ROUTINES ARE CALLED VIA A "JSR PC,@INTERX" (X=5 OR 6).
THE ROUTINE SETS UP AND RESPONDS TO A BUS REQUEST. IF THE
BR DOES NOT WORK THE RETURN PC IS INCREMENTED BY 2 AND
THE RETURN IS MADE.

.MAIN. MACY11 27(657) 13-MAR-75 10:08 PAGE 8
DEKBA.TMP

5.3 OPERATOR ACTION

THE LAST TEST OF DEKBB REQUIRES OPERATOR INTERVENTION.
THIS TEST IS ONLY EXECUTED ON PASS 1, IF SW<0>=0. QUES-
TIONS ARE TYPED ON THE TELETYPE AND THE OPERATOR MUST
RESPOND EITHER ON THE CONSOLE OR ON THE TELETYPE.

IF LOCATION 42 IS NON-ZERO, INDICATING THAT THE PROGRAM
WAS LOADED BY A MONITOR, THIS TEST IS SKIPPED ON ALL PASSES.

6. ERRORS

6.1 ERROR HALTS AND DESCRIPTION

A. DEKBA

EVERY ERROR IN DEKBA HALTS THE PROCESSOR. THE COMMENT
FIELD OF THE HALT INSTRUCTION CONTAINS THE NAME OF THE
SIGNAL THAT WAS MOST LIKELY TO HAVE CAUSED THE ERROR.
ALSO, IN THE COMMENT FIELD, IS THE OCTAL CODE THAT SHOULD
REPLACE THE HALT IF LOOP ON ERROR IS DESIRED. IF THE
PROGRAM HALTS AT LOCATION 6 OR 12 THE USER SHOULD LOOK
IN THE TEST DESCRIPTION, OF THE TEST THAT FAILED, TO FIND
THE MOST LIKELY CAUSE OF THE ERROR.

B. DEKBB

NONE OF THE ERRORS IN DEKBB HALT THE PROCESSOR IF SW<15>=0.

THERE ARE OVER 450(8) UNIQUE ERRORS THAT CAN OCCUR IN THIS

```
                                PROGRAM.  WHEN AN ERROR IS ENCOUNTERED THE CALL TO THE
                                ERROR ROUTINE IS MADE AND IF SW<13> IS NOT SET, AN ERROR
                                MESSAGE PERTAINING TO THE ERROR WILL BE TYPED.  EACH ERROR
                                TYPE OUT WILL CONTAIN THE FOLLOWING:
```

```
                                                    1.  AN ERROR MESSAGE
                                                    2.  A DATA HEADER
                                                    3.  A DATA STRING
```

THE DATA STRING WILL CONTAIN, AT A MINIMUM, THE ERROR PC
AND THE TEST NUMBER. IN SOME CASES THE EXPECTED AND ACTUAL
VALUES OF A REGISTER ARE ALSO INCLUDED.

REFER TO THE LISTING UNDER $ERRTB FOR THE TYPES OF ERRORS
THAT CAN OCCUR.

6.2 ERROR RECOVERY

A. DEKBA

ERROR RECOVERY IS STRICTLY BY USER INTERVENTION.

B. DEKBB

SW<15:9>=0 - MOST ERRORS WILL CAUSE EXECUTION TO GO TO
THE START OF THE NEXT TEST AFTER THE MESSAGE
IS TYPED. A FEW TESTS ARE DIVIDED INTO
SECTIONS. IN THESE TESTS AN ERROR WILL CAUSE
EXECUTION TO GO TO THE NEXT SECTION.

SW<15>=1 - PRESSING THE CONSOLE CONTINUE WILL CAUSE THE
PROGRAM TO TYPE AN ERROR MESSAGE AND HALT.
PRESSING THE CONSOLE CONTINUE AGAIN WILL CAUSE
THE PROGRAM TO CONTINUE AS IF SW<15>=0.

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

A. DEKBA

NONE

B. DEKBB

IF THE USER WANTS TO RUN THE BUS REQUEST 5 TEST HE MUST
ENSURE THAT EITHER AN RH CONTROLLER IS ACTIVE OR THAT A
UNIBUS DEVICE (RK, RS, RP, TM) IS ACTIVE.

7.2 OPERATING RESTRICTIONS

A. DEKBA

NONE

B. DEKBB

SINCE THE PROGRAM COULD POSSIBLY DESTROY A MONITOR, IN
PAGE 6, ALL LOCATIONS BETWEEN 152000 AND 157776 ARE SAVED
AT THE BOTTOM OF THE PROGRAM. TO RESTORE THESE LOCTIONS
A "CONTROL C" SHOULD BE TYPED ON THE TERMINAL. THE LOCATIONS
WILL BE RESTORED, A MESSAGE TYPED, AND THE PROCESSOR
WILL HALT.

IF THE PROGRAM IS RUNNING UNDER A MONITOR THE LOCATIONS

ARE RESTORED AND CONTROL IS RETURNED TO THE MONITOR
THRU THE END OF PASS LINKAGE.

8. MISCELLANEOUS

8.1 EXECUTION TIME

A. DEKBA

FIVE(5) SECONDS PER END OF PASS MESSAGE IF RUNNING UNDER A MONITOR.
4 MINUTES IF THE PROGRAM WAS DUMPED.

B. DEKBB

THE FIRST PASS TAKES APPROXIMATELY 8 SECONDS. ALL SUB-
SEQUENT PASSES TAKE APPROXIMATELY 3 MINUTES.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 1100.

8.3 PASS COUNT

A PROGRAM PASS THRU COUNT IS KEPT IN "$PASS".

A. DEKBA

IF THE PROGRAM IS RUNNING UNDER A MONITOR OR WAS LOADED
BY ACT 11 THE PROGRAM MAKES 144(8) PASSES FOR EACH END
OF PASS MESSAGE. IF THE PROGRAM WAS DUMPED, 10000(8) PASSES
ARE MADE FOR EACH END OF PASS MESSAGE.
THE PASS COUNT IS DISPLAYED IN THE DATA LIGHTS.

B. DEKBB

THE PROGRAM MAKES 1 PASS FOR EACH END OF PASS MESSAGE.

8.4 ITERATIONS

A. DEKBA

NONE

B. DEKBB

THE FIRST PASS OF THE PROGRAM WILL AUTOMATICALLY INHIBIT ITERATIONS. ALL SUBSEQUENT PASSES WILL PERFORM FULL, (2000 DECIMAL) ITERATIONS.

8.5 SPECIAL REGISTERS

A. DEKBA

R0 IS RESERVED FOR THE TEST NUMBER.

B. DEKBB

NONE

8.6 T BIT TRAPPING

A. DEKBA

NONE

B. DEKBB

EVERY OTHER PASS, STARTING WITH PASS 2, RUNS WITH THE T BIT ON. THIS CAUSES EVERY INSTRUCTION TO T BIT TRAP THEREFORE, IT IS NOT POSSIBLE TO "SINGLE INSTRUCTION" THE TEST WITHOUT TURNING THE T BIT OFF.

CERTAIN TESTS AUTOMATTICALLY TURN IT OFF IF IT WAS ON. THESE TESTS WILL ALSO TURN IT BACK ON UNLESS THE FOLLOWING TEST REQUIRES THAT IT ALSO BE OFF.

8.7 OSCILLOSCOPE SYNC POINTS

A. DEKBA

BEGINNING WITH TEST 24 EACH TEST HAS AN OSCILLOSCOPE SYNC INSTRUCTION. THE ADDRESS OF THE CONDITION CODE ROM STATE (44) IS IN THE PROCESSOR MICROBREAK REGISTER (ADDRESS 17777770). THIS WILL CAUSE PIN AE1 (SLOT 10) ON THE BACKPLANE TO GO HIGH EACH TIME A CONDITION CODE (OR NOP) INSTRUCTION IS EXECUTED. THEREFORE, IF THE OSCILLOSCOPE EXTERNAL SYNC IS CONNECTED TO THIS PIN AND THE SYNC SELECT PUT ON EXTERNAL THE OSCILLOSCOPE WILL BE SYNCHRONIZED WITH THE INSTRUCTION IMMEDIATELY PRECEEDING THE INSTRUCTION UNDER TEST (IUT).

B. DEKBB

ONLY TESTS 1 THRU 20 CONTAIN SYNC INSTRUCTIONS.

8.8 CACHE CONTROL

THE FIRST PASS OF BOTH PROGRAMS RUN WITH THE CACHE DISABLED (FORCING MISSES IN BOTH GROUPS). ALL SUBSEQUENT PASSES RUN WITH THE CACH ENABLED.

DOCUMENT
****
PDP 11/70 CPU DIAGNOSTIC PART 1
****

TABLE OF CONTENTS
****

TABLE OF CONTENTS
****

```
                                PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0
                                
                                        17      COPYRIGHT (C) JULY 21,1975
                                                DIGITAL EQUIPMENT CORP.
                                                MAYNARD, MASS. 01754

                                                PROGRAM BY DONALD W. MONROE

                                                THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                                PACKAGE (MAINDEC-11-DZQAC-A5).


                                        ;*****************************************************************
                                28      ;BASIC DEFINITIONS
                                        ;*****************************************************************

                                        30      INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

                                        44      MISCELLANEOUS DEFINITIONS

                                        50      GENERAL PURPOSE REGISTER DEFINITIONS

                                        71      PRIORITY LEVEL DEFINITIONS

                                        81      "SWITCH REGISTER" SWITCH DEFINITIONS

                                        109     DATA BIT DEFINITIONS (BIT00 TO BIT15)

                                        137     BASIC "CPU" TRAP VECTOR ADDRESSES

                                        ;*****************************************************************
                                153     ;CACHE   REGISTER DEFINITIONS
                                        ;*****************************************************************

                                        ;*****************************************************************
                                164     ;CPU REGISTER DEFINITIONS
                                        ;*****************************************************************

                                        ;*****************************************************************
                                178     ;MEMORY MANAGEMENT DEFINITIONS
                                        ;*****************************************************************

                                        181     MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

                                        192     USER "I" PAGE DESCRIPTOR REGISTERS

                                        203     USER "D" PAGE DESCRIPTOR REGISTORS

                                        214     USER "I" PAGE ADDRESS REGISTERS

                                        225     USER "D" PAGE ADDRESS REGISTERS
```

```
                ;*****************************************************************
327             ;UNIBUS MAP REGISTER DEFINITIONS
                ;*****************************************************************

330             ;THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
                ;THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

                ;*****************************************************************
419             ;TRAP CATCHER
                ;*****************************************************************

422             ;ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                ;SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                ;LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

                ;*****************************************************************
426             ;STARTING ADDRESS(ES)
                ;*****************************************************************

                ;*****************************************************************
432             ;ACT11 HOOKS
                ;*****************************************************************

434             ;THE FOLLOWING LOCATIONS ARE SETUP TO BE USED WITH ACT11

                ;LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOCICAL
                ;END OF THE PROGRAM.
                ;LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
                ;AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BIT
                ;TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:

                ;       BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
                ;            =0 NO POWER FAIL DESIRED

                ;       BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
                ;            =0 RUN TIME IS NOT MEMORY SIZE DEPENDENT

                ;       BITS 13-0  MUST BE ZERO'S
```

```
800
```

```
                                PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0

                        458     COMMON TAGS

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

                        506     ERROR POINTER TABLE

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

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

                        522             FOLLOWING IS AN INDEX OF THE POSSIBLE FAILURES THAT
                                        CAUSE A TRAP TO LOCATIONS 4, 10, 14, OR 24 OR THAT
                                        CAUSE THE PROCESSOR TO HANG (H). NGT=NOT GETTING THRU

                TEST NUMBER  EFFECT                             CAUSE

                     25                 10                      RACH NEG.B*DMD N
                     25                 24                      RACH A2 RAB00 NO
                     25                 H                       RACH E59(6) NOT
                                                                OR NGT RACL RADR
                     26                 H                       RACL RADR00 NOT
                     26                 4                       EITHER IRCC SM35
                                                                OR RACL E70 BAD
                     26                 14                      IRCC C0 RAB03 NO
                                                                RACL RADR03 INPU
                     26                 4                       SRC CONST ADDED

                     27                 H                       RACK RADR01 NOT

                     31                 10                      RACK A0 RAB01 NO
                                                                NGT RACL RADR01
                     31                 4                       RACK BRCAB05 NOT
                                                                NGT PACL RADR05
                                                                1$ ON TOP OF STA
                     31                 4                       DST CONST ADDED
                     32                 10                      RACE A0 RAB00 D0

                     33                 10                      RACE A0 RAB02 D0

                     34                 10                      RACE E44 IS BAD
                     34                 10                      IRCB K/CLASS STU
                     34                 H                       GRAB OBD(1) STUC
```

857

```
858                          PDP 11/70 CPU DIAGNOSTIC PART 1                    DECDOC VER  00.0
859
860                                                  34          H              GRAD DRMX00 STUC
861
862                                                  35          10             RACE BIN*SM0 H D
863                                                  35          10             IR DECODE ROM W0
864
865                                          558
866                                                  36          10             RACE BIN*SM0 FAI
867                                                  36          10             IR DECODE ROM W0
868
869                                                  37          10             RACE E45 BAD
870
871                                                  40          10             RACE A0 RAB00 D0
872                                                  40          10             IRCB(JMP+JSR) IS
873                                                  40          H              IRCB FJ CLASS IS
874
875                                                  41          10             RACE E45 IS BAD
876
877                                                  42          10             RACE E33 IS BAD
878
879                                                  43          10             RACH U/CLASS NOT
880                                                                             5$-2 ON TOP OF S
881                                                  43          10             EITHER RACE E10
882                                                                             RACE BIN NOT GOI
883                                                                             5$ ON TOP OF STA
884
885                                                  44          10             RACJ AFIR 14(1)
886                                                  44          10             IRCB E38(6) STUC
887
888                                                  45          H              GRAB OBD(0) STUC
889
890                                                  46          10             RACE E33 BAD
891
892                                                  54          10             RACF E3 DOES NOT
893
894                                                  56          10             PART PCLASS FIEL
895
896                                                  60          10             PART PCLASS FIEL
897
898                                                  62          10             IRCC C0 RAB03 ST
899                                                                             OR FORK C MUX IN
900
901                                                  65          10             B FORK MUX SELEC
902                                                  65          H              IRCB B0 RAB04 NG
903                                                  65          H              B FORK MUX INPUT
904                                                                             IRCC B0 RAB00 ST
905                                                  65          4              B FORK MUX INPUT
906                                                                             IRCB B0 RAB00 ST
907                                                  65          H              IRCB E46(10) STU
908
909                                                  67          10             RACE E35(1) BAD
910
911                                                  70          10             B FORK MUX STROB
912
913                                                  75          10             RACE JMP+JSR+SWA
```

914

```
                                PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0

                                                        75              10              IRCB E63 BAD-R5

                                                        105             4               RACF (HALT:OP CO
                                                        105             4               RACE E7 BAD-ODD


                                613                     THE FOLLOWING FIVE CONDITION CODE AND BRANCH TESTS ARE A
                                                        FUNCTION OF RACK F TRUE 1. SECTION 1 OF EACH TEST IS
                                                        DEPENDENT ON TRUE 1 NOT GOING HIGH, WHILE SECTION 2 IS
                                                        DEPENDENT ON TRUE ONE GOING HIGH.


                                620     TEST 1  CCC*BRANCH THRU FET.13

                                                        THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                                           SECTION 1
                                                                BCS     E58(13,12)      L,H
                                                                BMI     E59(10,11,9)    H,L,H
                                                                BVS     E48(5,3,4)      H,L,H
                                                                BLOS    E59(4,3,5)      H,L,H

                                642     TEST 2  SEC*BRANCH THRU FET.13 AND FET.11

                                                        THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                                           SECTION 1
                                                                BMI     E58(13,12)      H,L
                                                                BVS     E58(13,12)      H,L
                                                           SECTION 2
                                                                BCC     E58(13,12)      H,H

                                667     TEST 3  SEV*BRANCH THRU FET.13 AND FET.11

                                                        THE FOLLOWING IS A LIST OF PATTERNS PUT ON THE GATES OF
                                                           SECTION 1
                                                                BCS     E48(5,3,4)      L,H,H
                                                                BMI     E48(5,3,4)      H,H,L
                                                           SECTION 2
                                                                BVC     E48(5,3,4)      H,H,H

                                692     TEST 4  SEZ*BRANCH THRU FET.13 AND FET.11

                                                        THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                                           SECTION 1
                                                                BCS     E59(4,3,5)      H,H,L
                                                                BMI     E59(4,3,5)      L,H,H
                                                           SECTION 2
                                                                BHI     E59(4,3,5)      H,H,H

                                717     TEST 5  SEN*BRANCH THRU FET.13 AND FET.11

                                                        THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
```

```
                720             SECTION 1
                                    BLOS    E59(10,11,9)    H,H,L
                                    BVS     E59(10,11,9)    L,H,H
                                SECTION 2
                                    BPL     E59(10,11,9)    H,H,H

                742         THE FOLLOWING SEVEN TESTS ARE A FUNCTION OF RACF TRUE 2.
                        SECTION 1 OF EACH TEST IS DEPENDENT ON TRUE 2 NOT GOING HIGH
                        WHILE SECTION 2 IS DEPENDENT ON TRUE 2 GOING HIGH.

                        TEST 6  BRANCHES THRU FET.13

                                THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                   SECTION 1
                                        BLE     E58(2,1)H,L     E59(13,1,2)L,H,H
                                        BLT     E59(13,1,2)L,H,H        E48(1,13,2)H,H,L
                                        BEQ     E58(2,1)H,L     E58(10,9)H,L

                764     TEST 7  BRANCH THRU FET.13 AND FET.12

                                THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                   SECTION 1
                                        BEQ     E48(1,13,2)     L,H,H
                                   SECTION 2
                                        BGT     E48(1,13,2)     H,H,H

                788     TEST 10 BRANCH THRU FET.13 AND FET.12

                                THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                   SECTION 1
                                        BLT     E58(2,1)        L,H
                                   SECTION 2
                                        BNE     E58(2,1)        H,H

                811     TEST 11 BRANCH THRU FET.13 AND FET.12

                                THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                   SECTION 1
                                        BEQ     E59(13,1,2)     H,H,L
                                   SECTION 2
                                        BGE     E59(13,1,2)     H,H,H

                834     TEST 12 BRANCHES THRU FET.13 AND FET.12

                                THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TR
                                   SECTION 1
                                        BEQ     E58(2,1)H,L     E58(10,9)H,L
                                        BLT     E59(13,1,2)H,L,H        E48(1,13,2)H,L,H
```

```
                        850     TEST 13 UNIARY AND BINARY (SMO)

                                THE FOLLOWING TEST TESTS ALL THE E/CLASS
                                INSTRUCTIONS WITH A DESTINATION MODE OF 0
                                AND DESTINATION FIELD OF NOT 7. THIS CLASS CONSISTS
                                OF ALL THE UNIARY (EXCEPT NEG) INSTRUCTIONS AND
                                ALL THE BINARY INSTRUCTIONS WITH A SOURCE MODE
                                OF 0.

                        1143    TEST 14 REGISTER SELECTION TEST

                                THIS TEST ENSURES THAT THE 6 ADDRESS LINES INTO
                                THE GENERAL PURPOSE REGISTERS (GPR) ARE NOT STUCK.
                                THE LABELS OF THE ADDRESS LINES ARE:
                                        GSAX    GENERAL SOURCE ADDRESS LINE
                                        GDAX    GENERAL DESTINATION ADDRESS LINE
                                WHERE X STANDS FOR LINE 0,1, OR 2.
                                THE CLASSES OF ERRORS DESCRIBED IN THIS TEST

                        1152            ARE DEFINED AS FOLLOWS:
                                        CLASS A=GDAX OK
                                               GSAX STUCK
                                        CLASS B=GSAX OK
                                               GDAX STUCK
                                        CLASS C=GSAX STUCK
                                               GDAX STUCK

                        1248    TEST 15 GPR1 STUCK BIT TEST

                                LOADS GPR1 WITH ZEROS AND ONES AND COMPARES R1 SOURCE AN
                                DESTINATIONS WITH R0. IF THE COMPARISON FAILS A BIT IS S

                        1274    TEST 16 GPR2 STUCK BIT TEST

                                LOADS GPR2 WITH ZEROS AND ONES AND COMPARES R2 SOURCE AN
                                DESTINATION WITH R0.

                        1300    TEST 17 GPR3 STUCK BIT TEST

                                LOADS GPR3 WITH ZEROS AND ONES AND COMPARES
                                R3 SOURCE AND DESTINATION WITH R0.

                        1326    TEST 20 GPR4 STUCK BIT TEST

                                LOADS GPR4 WITH ZEROS AND ONES AND COMPARES
                                R4 SOURCE AND DESTINATION WITH R0.

                        1352    TEST 21 GPR5 STUCK BIT TEST

                                LOADS R5 WITH ZEROS AND ONES AND COMPARES
                                R5 SOURCE AND DESTINATION WITH R0.
```

1378 TEST 22 GPR6 STUCK BIT TEST

LOADS R6 WITH ZEROS AND ONES AND COMPARES
R6 SOURCE AND DESTINATION WITH R0.

1404 TEST 23 GPR SHORTED BIT TEST

TEST IF GPR'S 1 THRU 6 HAVE TWO BITS TIED TOGETHER

NOTE: R0 IS CONSIDERED "HARDCORE"

****

1509

****

1511 TEST 24 ONE MICROSTATE (E/CLASS*DM0*DF7)

THIS TEST EXECUTES AN ADD INSTRUCTION WITH SM0,DM0, AND
IF THE TEST FAIL THE SAME FLOW (EXC. 90) IS
TRIED WITH A PENDING BRQ (T BIT TRAP) INSTEAD OF A DF7.
A FORK A FAILURE IS REPORTED. IF IT PASSES, A TEST 1 FAI

ROM FLOW-30

****

1559

****

1561 TEST 25 TWO MICROSTATES (NEG*DM0)

THIS TEST EXECUTES A NEGATE INSTRUCTION WITH DM0.

IF FORK A FAILS EXECUTION WOULD GO TO EITHER RSD.00, ZAP
OR FOP.00.
FOP.00 WILL CAUSE THE PROCESSOR TO HANG.
RSD.00 WOULD CAUSE A TRAP TO LOCATION 10. THIS WOULD ONL
IF RACH NEG.B*DM0 DID NOT GO HIGH.
ZAP.00 WOULD CAUSE A TRAP TO LOCATION 24. THIS WOULD ONL
IF RACH A2 RAB00 DID NOT GO LOW.

ROM FLOW-301,210

```
                         PDP 11/70 CPU DIAGNOSTIC PART 1                             DECDOC VER  00.0

                                        ****************************************************************
                                1616
                                        ****************************************************************

                                        1618     TEST 26 THREE MICROSTATES (BIN*SM1*DM0*-DF7*SR0(0)

                                                         IF FORK A FAILS EXECUTION WILL GO TO EITHER EXEC.80 OR D
                                                         EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MIC
                                                         THIS WILL ONLY HAPPEN IF RACL RADR00 IS NOT GOING LOW DU
                                                         TO RACF A1 RAB00 (AFIR59(1)*[-BIN+SM01]*U/CLASS).
                                                         D12.00 WOULD MOV THE PC TO LOCATION 0.

                                                         IF FORK C FAILS EXECUTION WOULD GO FROM S13.10 TO D00.80
                                                         D45.01 OR S13.20 OR D12.00 OR JSR.10 OR ASC.80 OR RTI.50
                                                         D00.80 WOULD SWAP THE BYTES OF THE SOURCE OPERAND BEFORE
                                                         PUTTING THEM IN RS.
                                                         D45.01 WOULD MOVE THE PC TO LOCATION 0.
                                                         S13.20 WILL EXECUTE A SM3 (AND NO AUTO INC) INSTR.  THIS
                                                         AN ODD ADDRESS TRAP SINCE LOCATION POSERR CONTAINS AN OD
                                                         JSR.10 WOULD PUSH THE ADDR OF POSERR ONTO THE STACK.
                                                         ASC.80 WILL HALT AT 8$.
                                                         RTI.50 WILL CAUSE 1004XX TO BE PLACED IN THE PS WORD
                                                         AND THE PROCESSOR WILL TRAP TO LOCATION 14.
                                                         FOP.50 WILL ?????.

                                        1638             ASH.20 WOULD CAUSE A BAD CC.
                                                         IF THE SRC CONST FAILS IN STATE S13.00 AND ADDS 1 OR 3,
                                                         ADDRESS TRAP WILL OCCUR.
                                                         IF THE SRC CONSTANT ADDS 2, THE ERROR AT 6$ WILL REPORT

                                                         ROM FLOW-21,27,205

                                        1700     TEST 27 THREE MICROSTATES (BIN*SM2*DM0*-DF7*SR0(0)

                                                         THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR TH
                                                         SM. A WORD AND BYTE INSTRUCTION IS EXECUTED TO VERIFY TH
                                                         S13.01 ADDS THE CORRECT SOURCE CONSTANT.

                                                         IF FORK A FAILS EXECUTION WILL GO TO EXC.80.
                                                         EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MIC
                                                         THIS WILL ONLY HAPPEN IF RACL RADR01
                                                         IS NOT GOING LOW DUE TO RACF A1 RAB01 (AFIR10(1)*U/CLASS

                                                         ROM FLOW-22,27,205

                                        1752     TEST 30 ALU CARRY FUNCTIONAL TEST

                                                         THIS TEST DOES A COMPLETE CHECK OF THE ALU CARRY FUNCTIO
                                                         THE FIRST SECTION ENSURES THAT ALL THE INPUT AND OUTPUT
                                                         OK AND THE REST OF THE TEST ENSURES THAT THE CARRY LOGIC
                                                         FOLLOWING ARE THE LOGIC EQUATIONS FOR A 74S181 AND 74S18
                                                         DICTATED THE PATTERNS USED IN EACH SECTION:
                                                            74S181
                                                                  G=A3*B3+A2*B2*(A3+B3)+A1*B1*(A2+B2)*(A3+B3)+A0*B
```

1174

```
                                        P=(A3+B3)*(A2+B2)*(A1+B1)*(A0+B0)
                                        COUT=G+P*CIN
                                    74S182
                                        CX=G0+P0*CIN
                                        CY=G1+P1*G0+P1*P0*CIN
                                        CZ=G2+P2*G1+P2*P1*G0+P2*P1*P0*CIN

                1882    TEST 31 THREE MICROSTATES (DAC*DM2*0/CLASS)

                                IF FORK A FAILS EXECUTION WILL GO TO RSD.00. THIS WILL O
                                IF RACE A0 RAB01 DOES NOT GO LOW OR DOES NOT GET THRU
                                TO RACL RADR01.  THIS WILL CAUSE A TRAP TO LOCATION 10.

                                IF BEN15 FAILS EXECUTION WILL GO TO D45.80.
                                THIS WILL CAUSE A TRAP TO 4 WITH THE ADDRESS OF 1$ ON TH

                                IF THE DESTINATION CONSTANT FAILS(ADDS 1 OR 3) IN STATE
                                AN ODD ADDRESS TRAP WILL OCCUR.
                                IF THE DST CONST ADDS 0, THE ERROR AT 3$ WILL REPORT THE
                                IF THE DESTINATION IS NOT LOADED WITH THE SOURCE, THE
                                ERROR AFTER 5$ WILL REPORT THE FAILURE.

                                ROM FLOW-2,155,312

                1943    TEST 32 THREE MICROSTATES (DAC*DM1*0/CLASS)

                                THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR TH
                                IF FORK A FAILS, EXECUTION WILL GO TO RSD.00.
                                THIS WILL CAUSE A TRAP TO LOCATION 10 WITH AN ODD ADDRES
                                THIS WILL ONLY HAPPEN IF RACE A0 RAB00 DOES NOT GO LOW O
                                DOES NOT GET TO RACL.
                                EITHER E44 OR E6 IS BAD.

                                IF THE INSTRUCTION FAILS TO MOVE R5 TO THE PC INDIRECT,
                                THE ERROR AFTER 1$ WILL REPORT THE FAILURE.

                                ROM FLOW-1,155,312

                1976    TEST 33 THREE MICROSTATES (DAC*DM4*0/CLASS)

                                IF FORK A FAILS EXECUTION WILL GO TO RSD.00.
                                THIS WILL CAUSE A TRAP TO LOCATION 10. THIS WILL ONLY HA
                                RACE A0 RAB02 IS NOT GOING LOW. EITHER RACE E33 OR
                                E6(9&8) IS BAD.

                                IF THE DST CONST FAILS TO SUBTRACT 2, THE ERROR AT EITHE
                                OR 1$-2 WILL REPORT THE FAILURE.

                                IF BEN01 FAILS (CAUSED BY IRCD DM357 STUCK HIGH)
                                EXECUTION WILL GO TO D10.00 WHICH WILL EXECUTE A MODE 5
                                INSTEAD OF MODE 4.

                                IF THE DESTINATION IS NOT LOADED PROPERLY,
                                THE ERROR AT 3$ WILL REPORT THE FAILURE.
```

```
1231
```

ROM FLOW-4,122,157

2037 TEST 34 THREE MICROSTATES (DAC*DM1*TST.B*DR0(0))

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY HAPPEN IF RA SAME?

2055

IF BEN15 FAILS EXECUTION WILL GO TO STATE D12.60.

IF B FORK FAILS (AFTER STATE D12.10) CAUSED BY IRCB
K/CLASS STUCK LOW EXECUTION WILL GO TO STATE RSD.00 CAUS
IF IRCB B1 RAB00 IS STUCK HIGH EXECUTION WILL GO FROM
D12.10 TO JSR.40. THIS WILL CAUSE THE PROCESSOR TO HANG.
IF EITHER GRAB 0BD(1) IS STUCK HIGH OR NOT GETTING THRU
EXECUTION WILL GO TO STATE D12.30.
IF GRAD DRMX00 IS STUCK HIGH OR GRAB E50 IS BAD, EXECUTI
WILL GO TO D10.60 WHICH WILL HANG THE PROCESSOR.

ROM FLOW-1,175,33

2093 TEST 35 THREE MICROSTATES (DAC*DM1*BIT.B*DR0(0))

THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT A BIT
IF FORK A FAILS RACE BIN*SM0 H FAILED.

IF FORK B FAILS THE INSTRUCTION DECODE ROM WORD IS BAD.

IF THE RESULTANT DATA IS BAD STATE TST.10 FAILED.

ROM FLOW-1,175,33

2116 TEST 36 THREE MICROSTATES (DAC*DM1*CMP.B*DR0(0))

THIS TEST IS THE SAME AS THE PREVIOUS TWO TESTS
EXCEPT A CMP INSTRUCTION IS USED.

ROM FLOW-1,175,33

2135 TEST 37 THREE MICROSTATES (DAC*DM2*TST.B*DR0(0))

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY HAPPEN IF RACE E45 IS BAD (AFIR04(1)*R/CL

BEN15 & FORK B HAVE ALREADY BEEN TESTED
IF THE AUTO INC FAILS IT WILL BE DUE TO A BAD
FIELD IN ROM STATE D12.10.

ROM FLOW-2,175,33

```
1285
1286
1287                                2165    THE LOGICAL SEQUENCE WOULD NEXT TEST THE BIT.B AND CMP.B INSTRUC
1288                                        THESE WILL NOT BE TESTED WITH INDIVIDUAL TESTS SINCE THEY DO NOT
1289                                        ANY HARDWARE THAT HAS NOT ALREADY BEEN TESTED.
1290
1291                                2172    TEST 40 THREE MICROSTATES (JMP*DM1)
1292
1293                                            IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
1294                                        THIS WILL ONLY HAPPEN IF RACE A0 RAB00 DOES NOT GO LOW
1295                                        (AFIR03(1)*[JMP+JSR+SWAB]).
1296
1297
1298                                2178        A FORK B FAILURE WOULD BE ONE OF THE FOLLOWING:
1299                                        IF IRCB (JMP+JSR) IS STUCK LOW EXECUTION WILL GO TO RSD.
1300                                        A TRAP TO 10.
1301                                        IF IRCB IR(14:9) 04 IS STUCK LOW EXECUTION WILL
1302                                        GO TO JSR.00 WHICH WILL EXECUTE A JSR INSTEAD OF A JMP.
1303                                        IF IRCB B FORK MUX FAILS EXECUTION WILL GO TO
1304                                        FOP.00.
1305                                        IF IRCB FJ CLASS IS STUCK HIGH EXECUTION WILL GO TO
1306                                        STATE D12.00 AND THE JMP WON'T JUMP.
1307
1308                                        IF THE INSTRUCTION FAILS TO JUMP, STATE JMP.00 IS REPORT
1309
1310                                        ROM FLOW-1,135,35
1311
1312                                2212    TEST 41 THREE MICROSTATES (JMP*DM2)
1313
1314                                        THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT THE DM
1315                                        IF FORK A FAILS RAC E45 IS BAD (AFIR04(1)*[JMP+JSR+SWAB]
1316
1317                                        ROM FLOW-2,135,35
1318
1319                                2229    TEST 42 THREE MICROSTATES (JMP*DM4)
1320
1321                                            IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
1322
1323                                2232        THIS WILL ONLY HAPPEN IF RACE E33(AFIR05(1)*[JMP+JSR+SWA
1324
1325                                        ALL OTHER LOGIC HAS BEEN TESTED.
1326
1327                                        ROM FLOW-4,122,35
1328
1329                                2246    TEST 43 THREE MICROSTATES (SOB)
1330
1331                                            IF RACH U/CLASS IS NOT GOING HIGH EXECUTION WILL GO TO R
1332                                        CAUSING A TRAP TO LOCATION 10 WITH THE ADDRESS OF 5$-2
1333                                        ON THE TOP OF THE STACK.
1334                                        IF EITHER RACE E10 IS BAD OR RACE BIN DOES NOT GO HIGH
1335                                        EXECUTION WILL GO TO D67.01. EXECUTION WILL EVENTUALL GO
1336                                        TO RSD.00 AFTER STATE D10.60 AND THE STACKED PC WILL BE
1337                                        IF RACF E8 FAILS EXECUTION WILL GO TO ASC.10 WHICH WILL
1338                                        PERFORM AN ASHC*DM0 OPERATION.
1339
1340                                        IF THE SOB BRANCHES WHEN IT IS NOT SUPPOSE TO EITHER
```

```
1341
```

GRAE SR EQ ONE IS STUCK HIGH OR RACK E63 IS BAD
OR STATE SOB.10 DOES NOT RESTORE THE OLD PC.

IF THE SOB DOES NOT BRANCH WHEN IT IS SUPPOSE TO EITHER
STATE SOB.00 IS BAD OR GRAE SR EQ ONE STUCK LOW OR
RACK E63(C1) IS BAD.

IF THE REGISTER DOES NOT DECREMENT STATE SOB.20 IS BAD.

ROM FLOW-57,242/262.262

```
                ;*****************************************************************
2311            ;*****************************************************************
```

2313 TEST 44 FOUR MICROSTATES (DAC*DM12*P/CLASS*DR0(0)

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY HAPPEN IF RACJ AFIR 14(1) DOES NOT
GET THRU RAC E42.

THE FOLLOWING COULD BE FORK B FAILURES:
IF IRCB B0 RAB00 IS STUCK HIGH OR NOT GETTING THRU
TO RACL RADR00 EXECUTION WILL GO TO EXC.90 WHICH WOULD
PUT THE RESULT INTO A REGISTER RATHER THAN MEMORY.
IF IRCB E38(6) IS STUCK HIGH EXECUTION WILL GO TO
RSD.00 CAUSING A TRAP TO 10.
IF THE BIC DOES NOT HAPPEN THEN EXC.00 IS BAD.

ROM FLOW-2,175,31,132

2358 TEST 45 FOUR MICROSTATES (DAC*DM12*TST.B*DR0(1))

AFTER STATE D12.10 IF EITHER GRAB OBD(1) DOES NOT GO HIG
OR DOES NOT GET THRU RACL E71 EXECUTION WILL GO
TO TST.10. IF EITHER GRAB OBD(0) IS STUCK HIGH OR NOT GE
THRU RACK E41, EXECUTION WILL GO TO D10.60. THIS WILL CA
THE PROCESSOR TO HANG UP IN THE PAUSE STATE AT MICRO
ADDRESS 177. IF THE TEST FAILS THEN STATE D12.30 FAILED

ROM FLOW-1,175,137,33

2382 TEST 46 FOUR MICROSTATES (DAC*DM4*TST.B*DR0(0))

IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
THIS WILL ONLY OCCUR IF RACE E33(AFIR05(1)*R/CLASS) IS B

AFTER D10.30 IF IRCB FJ/CLASS DOES NOT GET TO RACL E71
OR IF RACL E71 IS BAD EXECUTION WILL GO TO SVC.50.

IF THE INSTRUCTION DOESN'T WORK THEN D10.60 IS BAD.

ROM FLOW-4,122,177,33

2414 THE LOGICAL SEQUENCE HERE WOULD BE TO TEST THE BIT & CMP INSTRUC
BUT NO ADDITIONAL LOGIC IS TESTED BY THEM.

2420 TEST 47 FOUR MICROSTATES (DAC*DM6*0/CLASS)

FORK A SHOULD NOT FAIL ON THIS TEST.

BEN01 SHOULD NOT FAIL.

BEN15*FEN2 SHOULD NOT FAIL.

IF D67.00 FAILS TO INCREMENT THE PC AN RTI INSTRUCTION
WILL BE EXECUTED.
IF D67.00 FAILS TO CLOCK THE BR THE INSTRUCTION
WILL BE ADDED AS THE INDEX WORD.
IF D67.10 FAILS TO ADD THE INDEX NUMBER THE SOURCE
WILL BE PUT IN THE WRONG LOCATION.

ROM FLOW-6,251,122,157

2461 TEST 50 FOUR MICROSTATES (BIN*SM12*DM0*-DF7*SR0(1))

IF FORK A FAILS EXECTUTION WILL GO TO STATE D12.00.
THIS WILL HAPPEN IF RACE BR 1=7 DOES NOT GO HIGH.
STATE D12.00 WOULD BITB R5 & THE CONTENTS OF LOCATION 20
WHICH IS 137.

THE FOLLOWING COULD BE BEN14*FORK C FAILURES:
IF IRCC CO RAB00 DOES NOT GO HIGH EXECUTION WILL GO TO
D00.90 WHICH WILL TEST THE LOW BYTE INSTEAD OF THE HIGH

IF STATE D00.80 FAILS TO SWAP THE BYTES IT WILL LOOK LIK
A FORK C FAILURE.

ROM FLOW-21,27,204,205

2499 TEST 51 FOUR MICROSTATES (BIN*SM12*DM0*DF7*SR0(0))

IF FORK A FAILS EXECUTION WILL EITHER GO TO STATE D12.00

2502 STATE D12.00 WOULD ADD R5 TO THE CONTENTS OF THE PC.
STATE EXC.80 WOULD NOT CHANGE THE PC.

IF FORK C FAILS EXECUTION WILL EITHER GO TO JSR.10 OR AS
JSR.10 WILL CAUSE R5 TO BE STACKED, THE PC TO BE PUT
IN R5, AND THE PC REPLACED BY R5 WHICH WILL CAUSE AND RT
THE CONTENTS OF THE LOCATION POINTED TO BY R5 IS 000002.
ASC.80 WILL CAUSE THE ADD TO LOOK LIKE IT FAILED.

IF STATE D07.10 FAILS TO LOAD THE SHFTR THE PC WILL
DOUBLE AND THE PROGRAM WILL BLOW UP.

IF THE SHFTR FAILS TO BE PUT IN THE SR THE PC
WILL NOT CHANGE.

```
1454
```

```
                        ROM FLOW-21,27,203,30

2567    TEST 52 FOUR  MICROSTATES (BIN*SM12*DM0*DF7*SR0(1))

                        IF FORK A FAILS EXECUTION WILL GO TO D12.00.

                        FORK C SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN

                        IF STATE D07.00 FAILS TO SWAP THE BYTES THE CC'S WILL
                        BE BAD.
                        IF D07.00 FAILS TO LOAD THE SHIFTER, THE THIRD CMPB WILL
                        IF D07.00 FAILS TO LOAD THE SR THE SECOND CMPB WILL FAIL

                        ROM FLOW-21,27,202,30

2604    TEST 53 FOUR  MICROSTATES (BIN*SM4*DM0*-DF7*SR0(0))

                        IF FORK A FAILS EXECUTION WILL
                        GO TO D45.00 WHICH WILL EXECUTE A SM0*DM4 INSTRUCTION EX
                        THE DESTINATION REGISTER WILL NOT DECREMENT.

2610                    FORK C WILL NOT FAIL SINCE IT HAS ALREADY BEEN TESTED.

                        IF THE SRC FAILS TO AUTO DECREMENT STATE S45.00 IS BAD.

                        ROM FLOW-24,23,27,205

2639    TEST 54 FOUR  MICROSTATES (RTS)

                        IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
                        THIS WILL ONLY HAPPEN IF RACF E3 DOES NOT GO HIGH.

                        IF THE PC OR R5 FAILS THE TEST WILL HALT.

                        ROM FLOW-40,223,224,342

2661    TEST 55 FOUR  MICROSTATES (JMP*DM6)

                        IF FORK A FAILS EXECUTION WILL GO TO STATE D12.01.

2664                    THIS WOULD CAUSE A JMP*DM1 TO EXECUTE.

                        NEITHER BEN01 NOR BEN15*FEN2 SHOULD FAIL SINCE THEY HAVE
                        TESTED.

                        ROM FLOW-6,251,122,35
```

```
                        **********************************************************************
                2681
                        **********************************************************************

                2683    TEST 56 FIVE  MICROSTATES (DAC*DM12*P/CLASS*DR0(1))

                        FORK A SHOULDN'T FAIL.

                        BEN15 SHOULDN'T FAIL SINCE THIS LOGIC HAS BEEN TESTED.
                        NEITHER SHOULD BEN05*FEN2.

                        IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
                        THIS FAILURE WOULD BE CAUSED BY A BAD FIELD (PART PCLASS
                        IN THE IR DECODE ROM.

                        ROM FLOW-1,175,137,31,132

                2708    TEST 57 FIVE  MICROSTATES (DAC*DM3*0/CLASS)

                        FORK A SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN

                        IF THE DR DOES NOT AUTO INC THEN STATE D30.10 IS BAD.

                        IF THE CONDITION CODES ARE BAD THEN EITHER STATE D10.50
                        DID NOT LOAD THE BR OR THE DOUBLE DEFERED DIDN'T WORK.

                        ROM FLOW-3,221,233,311,157

                2741    TEST 60 FIVE  MICROSTATES (DAC*DM4*P/CLASS*DR0(0))

                        FORK A SHOULD NOT FAIL.

                        IF FORK B FAILS, AFTER D10.60, EXECUTION WILL GO TO
                        RSD.00 CAUSING A TRAP TO 10. THIS WILL ONLY HAPPEN IF
                        THE IR DECODE ROM HAS A BAD FIELD (PART PCLASS).

                        ROM FLOW-4,122,177,31,132

                2765    THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A DAC*DM4
                BIT.B+CMP.B]*DR0(1) INSTRUCTION FOLLOWED BY A DAC*DM6*[TST.B+BIT
                DR0(0) INSTRUCTION TEST BUT NO ADDITIONAL LOGIC IS TESTED.

                2774    THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A BIN*SM4
                FOLLOWED BY A BIN*SM4*DM0*DF7*SR0(0)
                FOLLOWED BY A BIN*SM4*DM0*DF7*SR0(1) INSTRUCTION BUT NO ADDITION

                2781    TEST 61 FIVE  MICROSTATES (BIN*SM6*DM0*-DF7*SR0(0))

                        IF FORK A FAILS EXECUTION WILL GO TO STATE D67.00
                        WHICH WOULD EXECUTE A SM0*DM6 INSTRUCTION.

                        BEN14*FEN4 SHOULD NOT FAIL SINCE IT HAS ALREADY BEEN TES

                        ROM FLOW-26,54,141,142,205
```

1562

```
                                PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0

                        2807    TEST 62 FIVE  MICROSTATES (BIN*SM12*DM12*0/CLASS)

                                IF FORK A FAILS EXECUTION WILL GO TO D12.01.THIS WOULD C
                                TO BE WRITTEN INTO $TMP2.

                                IF FORK C FAILS EXECUTION WOULD GO TO ONE OF THE
                                FOLLOWING STATES: D45.80,D30.80,FOP.00,WAT.00, D00.90, A
                                STATE D45.80 WOULD EXECUTE A SM2*DM4 INSTEAD OF SM2*DM1.
                                STATE D30.80 WOULD EXECUTE A SM1*DM3.
                                STATE FOP.00 WOULD CAUSE A TRAP TO LOCATION 10.
                                THIS WILL ONLY HAPPEN IF EITHER IRCC CO RAB03 IS STUCK
                                OR IT IS NOT GETTING THRU RACL RADR03 OR
                                IRCC FORK C MUX INPUT B0 IS HIGH.
                                THE LA30 PRINTER BUFFER WILL BE SET UP TO GENERATE AN IN
                                CASE THE TEST FAILS TO THE WAT.00 STATE.
                                STATE D00.90 WILL EXECUTE A DM0 INSTEAD OF A DM1.
                                STATE ASH.20 WILL CLEAR THE C BIT.

                                ROM FLOW-22,27,111,155,312

                        2900    THE LOGICAL FLOW AT THIS POINT WOULD TEST A BIN*SM12*DM12*SR0(0)
                                *[TST.B+BIT.B+CMP.B] INSTRUCTION BUT NO ADDITIONAL LOGIC WOULD B

                        2906    TEST 63 FIVE  MICROSTATES (BIN*SM12*DM12*SR0(1)*DR0(0)*CMPB)

                                THE ONLY THING THAT SHOULD FAIL WOULD BE STATE D12.90(11

                                ROM FLOW-21,27,110,175,33

                        2925    TEST 64 FIVE  MICROSTATES (BIN*SM12*DM4*0/CLASS)

                                WHICH WILL EXECUTE A SM1*DM2 TYPE INSTRUCTION.

                                IF THE INSTRUCTION FAILS EITHER D45.80 OR D40.20 FAILED.

                                ROM FLOW-21,27,115,121,157

                        ;*******************************************************************
                2950
                        ;*******************************************************************

                        2952    TEST 65 SIX   MICROSTATES (DAC*DM12*ASRB*DR0(1))

                                NEITHER FORK A NOR BEN15 NOR BEN05*FEN2 SHOULD FAIL
                                SINCE THEY HAVE ALREADY BEEN TESTED.

                                IF FORK B FAILS AFTER D12.30 EXECUTION WILL GO TO
                                ONE OF THE FOLLOWING: RSD.00,D45.00,EXC.00,S45.00,
                                CCP.00,MUL.00,SVC.10,MFP.00 OR DEP.00.
                                RSD.00 WILL CAUSE A TRAP TO LOCATION 10.
                                THIS WILL HAPPEN IF THE B FORK MUX SELECT IS STUCK LOW.
                                IF STATE D45.00 IS ENTERED THE PROCESSOR WILL HANG UP
                                IN A LOOP BETWEEN STATES D45.00 AND D10.30.
                                IF STATE S45.00 IS ENTERED EXECUTION WILL GO TO STATE
```

```
1619
```

```
                                        D12.80 AFTER S13.10 WHICH WILL HANG UP THE PROCESSOR.
                                        STATE CCP.00 WOULD SET OR CLEAR THE CONDITION CODES
                                        ACCORDING TO IR(4:0).
                                        STATE MUL.00 WOULD CAUSE THE DESTINATION OPERAND TO
                                        BE MULTIPLIED BY REGISTER 2 AND THE RESULT WOULD BE
                                        STORED IN REGISTER 2 AND 3.
                                        IF SVC.10 IS ENTERED A TRAP TO 4 WILL OCCUR BECAUSE
                                        THE DESTINATION REGISTER IS ODD. THIS WILL ONLY HAPPEN
                                        IF EITHER B FORK MUX INPUT B3 OR IRCB B0 RAB00 IS STUCK
                                        IF MFP.00 IS ENTERED AN MFPI INSTRUCTION WILL BE EXECUTE
                                        THIS WILL PUSH THE ADDRESS OF 1$ ONTO THE STACK.
                                        DEP.00 WILL CAUSE THE PROCESSOR TO HANG IN MICRO ADDRESS
                                        170 WITH THE RUN LIGHT ON.

                                        ROM FLOW-1,175,137,64,123,132

                                3028    TEST 66 SIX   MICROSTATES (DAC*DM12*RORB*DRO(1))

                                        THIS TEST IS THE SAME AS THE LAST ONE EXCEPT A RORB
                                        IS USED INSTEAD OF AN ASRB.

                                        FORK B WILL ONLY FAIL IF IRCB E36(13) IS STUCK HIGH
                                        WHICH WILL CAUSE EXECUTION TO GO TO EXC.00.

                                        ROM FLOW-2,175,137,64,123,132

                                3053    THE LOGICAL FLOW AT THIS POINT WOULD TEST A DAC*DM3*[TST.B+BIT.B
                                        FOLLOWED BY A DAC*DM4*P/CLASS*DRO(0) INSTRUCTION BUT NO ADDITION
                                        IS TESTED

                                3061    TEST 67 SIX   MICROSTATES (DAC*DM6*XOR*DRO(0))

                                                IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
                                                THIS SHOULD ONLY HAPPEN IF RACE E35(1) IS BAD.

                                                IF THE INSTRUCTION DOESN'T WORK IT WILL HALT IN THIS TES

                                                ROM FLOW-6,251,122,177,31,132

                                3086    THE LOGICAL SEQUENCE WOULD TEST A DAC*DM6*[TST.B+BIT.B+CMP.B]*DR
                                        THE LOGIC HAS BEEN TESTED.

                                3093    TEST 70 SIX   MICROSTATES (NEG.B*DM12*DRO(0))

                                                NEITHER FORK A NOR BGN15 SHOULD FAIL.

                                3096
                                                IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING
                                                A TRAP TO LOCATION 10 OR EXC.00.
                                                RSD.00 SHOULD ONLY OCCUR IF THE B FORK MUX
                                                STROBE IS BEING HELD LOW(CHIP FAILURE).

                                                ROM FLOW-1,175,67,271,163,132
```

3135 THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A JMP*DM3 BUT NO ADDITIONAL LOGIC IS TESTED.

3141 TEST 71 SIX MICROSTATES (BIN*SM3*DM0*-DF7*SR0(0))

FORK A SHOULD NOT FAIL.

IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90. THIS WILL CAUSE A SM2 TO BE EXECUTED. THIS SHOULD ONLY HAPPEN IF IRCC SM357 IS STUCK LOW OR NOT GETTING THRU RA

ROM FLOW-22,27,317,143,146,205

3170 THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM6*DM0*DF7*SR0(1), T DM12*SR0(0)*DR0(1)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM12*DM12*SR0(DR0(0)*P/CLASS THEN A BIN*SM12*DM12*SR0(1)*DR0(1)*[TST.B+BIT.B+C THEN A BIN*SM12*DM12*SR0(1)*DR0(0)*P/CLASS AND THEN A BIN*SM12*D SR0(0)*DR0(0)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT NO ADDITIONAL IS TESTED.

3180 TEST 72 SIX MICROSTATES (BIN*SM12*DM4*SR0(1)*DR0(0)*CMPB)

A FAILURE WILL ONLY OCCUR IF STATE D45.90 FAILS.

IF THE DST REG. DOES NOT DECREMENT STATE D45.90 IS BAD.
IF THE CONDITION CODES ARE BAD THEN STATE D40.30 PROBABLY DID NOT SWAP THE BYTES OF THE SRC OPERAND.

ROM FLOW-1,27,114,131,177,33

3212 THE LOGICAL FLOW WOULD NEXT TEST A BIN*SM4*DM12*SR0(0)*DR0(0)*0/ THEN A BIN*SM4*DM12*SR0(0)*DR0(0)*[TST.B+BIT.B+CMP.B] THEN A BIN DM12*SR0(1)*DR0(0)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM4*DM4*SR0(0) 0/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.

3220 TEST 73 SIX MICROSTATES (DAC*DM5*DR0(0)*0/CLASS)

FORK A SHOULD NOT FAIL.

IF IRCD DM357 IS STUCK LOW OR NOT GETTING THRU
TO RACK E51 A DM4 WILL BE EXECUTED.
IF STATE D10.00 OR D10.10 FAIL TO FETCH THE DEFERED ADDR
THE SOURCE WILL BE STORED IN THE DESTINATION.

ROM FLOW-5,162,231,233,311,157

3252 THE LOGICAL SEQUENCE WOULD NEXT TEST A DAC*DM3*DR0(0)*P/CLASS TH DAC*DM3*DR0(1)*[TST.B+BIT.B+CMP.B] THEN A DAC*DM4*DR0(1)*[ASRB+ RORB] THEN A DAC*DM5*DR0(0)*[TST.B+BIT.B+CMP.B] THEN A DAC+DM6* DR0(1)*P/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.

```
1729                                PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0
1730
1731                                ************************************************************************
1732                        3259
1733                                ************************************************************************
1734
1735                        3261    TEST 74 SEVEN MICROSTATES (DAC*DM7*0/CLASS)
1736
1737                                FORK A SHOULD NOT FAIL.
1738
1739                                IF IRCD DM357 DOES NOT GO HIGH A DM6 WILL BE EXECUTED.
1740
1741                                ALL OTHER LOGIC HAS BEEN TESTED.
1742
1743                                ROM FLOW-7,251,162,231,233,311,157
1744
1745                        3292    THE LOGICAL SEQUENCE WOULD NEXT TEST A NEG.B*DM12*DR0(1) THEN A
1746                                THEN A JMP*DM5 INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.
1747
1748                        3298    TEST 75 SEVEN MICROSTATES (JSR*DM12)
1749
1750                                IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TR
1751                                THIS WILL ONLY OCCUR IF RACE JMP+JSR+SWAB DOES NOT GO HI
1752
1753                                IF EITHER IRCB IR(14:9)04 DOES NOT GO LOW OR E63 IS BAD
1754                                EXECUTION WILL GO FROM D12.10 TO EXC.00.
1755                                IF IRCB E63 IS BAD (PIN 10 OR 485 FLOATING) EXECUTION WI
1756                                GO TO RSD.00 CAUSING A TRAP TO LOCATION 10. THIS FAILURE
1757                                WOULD INCREMENT THE DST REG. BEFORE THE TRAP.
1758
1759                                IF THE INSTRUCTION FAILS THEN ONE OF THE JSR STATES FAIL
1760
1761                                ROM FLOW-2,135,34,201,274,275,32
1762
1763                        3342    THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM3*DM0*-DF7*SR0(1
1764                                A BIN*SM3*DM0*DF7*SR0(0) THEN A BIN*SM3*DM0*DF7*SR0(1) INSTRUCTI
1765                                BUT NO ADDITIONAL LOGIC IS TESTED.
1766
1767                        3349    TEST 76 SEVEN MICROSTATES (BIN*SM5*DM0*-DF7*SR0(0))
1768
1769                                FORK A SHOULD NOT FAIL.
1770
1771                                IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90
1772                                CAUSING A SM4 INSTRUCTION TO BE EXECUTED. THIS WILL ONLY
1773                                OCCUR IF EITHER IRCC SRCM5 DOES NOT GO LOW OR IF IRCC E2
1774
1775                                ROM FLOW-24,23,27,317,143,146,205
1776
1777                        3378    THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM12*DM12*SR0(0)*D
1778                                P/CLASS THEN A BIN*SM12*DM12*SR0(1)*DR0(1)P/CLASS INSTRUCTION, B
1779                                NO ADDITIONAL LOGIC IS TESTED.
1780
```

```
                3385    TEST 77 SEVEN MICROSTATES (BIN*+SM12*DM3*0/CLASS)

                        IF IRCC C FORK MUX INPUT B2 IS NOT GOING LOW OR IRCC E40
                        IS BAD A DM2 WILL BE EXECUTED.

                        THE ONLY OTHER  POSSIBLE FAILURE IS STATE D30.80.

                        ROM FLOW-21,27,113,221,233,311,157

                3414    THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM4*SRO(0)*DRO(0
                        P/CLASS FOLLOWED BY A BIN*SM12*DM4*SRO(0)*DRO(1)*[TST.B+BIT.B+CM
                        FOLLOWED BY A BIN*SM12*DM4*SRO(1)*DRO(0)*P/CLASS FOLLOWED BY A
                        BIN*SM12*DM4*SRO(1)*DRO(1)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT
                        ADDITIONAL LOGIC IS TESTED.

                3423    TEST 100        SEVEN MICROSTATES (BIN*SM12*DM6*0/CLASS)

                        IF FORK C FAILS EXECUTION WILL GO TO D45.90 AND A
                        DM4 WILL BE EXECUTED. THIS WILL ONLY HAPPEN IF IRCC E39
                        IS NOT GOING LOW. THIS WILL CAUSE AN RTI SINCE THE LOCAT
                        FOLLOWING THE INSTRUCTION CONTAINS 000002.

                        THE ONLY OTHER FAILURE WOULD BE CAUSED BY STATE D67.80 B

                        ROM FLOW-21,27,117,6,251,122,157

                3454    THE LOGICAL SEQUENCE WOULD NEXT TEST THE INSTRUCTIONS BETWEEN
                        BIN*SM4*DM12*SRO(0)*DRO(1)*[TST.B+BIT.B+CMP.B] AND BIN*SM5*DM0*D
                        BUT NOT ADDITIONAL LOGIC IS TESTED.

                ********************************************************************
        3460
                ********************************************************************

                3462    TEST 101        EIGHT MICROSTATES (BIN*SM7*DM0*-DF7*SRO(0))

                        FORK A SHOULD NOT FAIL.

                        IF FEN4*BEN14 FAILS EXECUTION WILL GO TO D00.90 CAUSING
                        A SM6 TO BE EXECUTED. THIS WILL ONLY HAPPEN IF EITHER
                        IRCC SRCM7 DOES NOT GO LOW OR IF IRCC E28(1) IS BAD.

                        ROM FLOW-26,54,141,142,317,143,146,205

                3492    THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM3*SRO(0)*DRO(0
                        BIT.B+CMP.B] BUT NO ADDITIONAL LOGIC IS TESTED.

                3498    TEST 102        EIGHT MICROSTATES (BIN*SM12*DM3*SRO(1)*DRO(0)*CM

                        THE ONLY POSSIBLE FAILURE WOULD BE IN STATE D30.90
                        SINCE ALL THE OTHER LOGIC HAS BEEN TESTED.

                        ROM FLOW-21,27,112,221,233,311,177,33
```

```
                3520    THE LOGICAL SEQUENCE WOULD NEXT TEST INSTRUCTIONS BIN*SM12*DM4*S
                        P/CLASS THRU BIN*SM12*DM6*SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] BUT
                        ADDITIONAL LOGIC IS TESTED.

                3527    TEST 103        EIGHT MICROSTATES (BIN*SM12*DM6*SRO(1)*DRO(0)*CM

                3528
                            THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D67.90.

                            ROM FLOW-21,27,116,6,251,122,177,33

                ************************************************************************
        3543
                ************************************************************************

                3545    TEST 104        NINE  MICROSTATES (BIN*SM12*DM5*SRO(0)*DRO(0)*CM

                            THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.20.

                            ROM FLOW-21,27,115,161,231,233,311,177,33

                3564    TEST 105        EIGHT MICROSTATES (BIN*SM12*DM5*SRO(1)*DRO(0)*CM

                            THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.30.

                3580    TEST 106        WRITE/READ PSW

                3582            THIS TEST VERIFIES THAT THE PSW CAN BE READ THRU THE DAT
                                IF THE TEST FAILS ONE OF MANY THINGS COULD BE BAD WHICH
                                DETERMINED IN THIS DIAGNOSTIC.
                                THIS TEST REQUIRES THAT SCCE PS ADRS GETS TO TMC,
                                THAT THE TMC DMUX SELECT LINES GET TO PDR, THAT THE PSW
                                BITS GET TO THE DMUX, THAT SCCE INTERNAL ADDRESS GETS TO
                                AND SCCA VADD GETS TO UBC.

                3615    TEST 107        RTI

                            IF FORK A FAILS EXECUTION WILL GO TO ONE OF THREE STATES
                            RSD.00 WILL CAUSE A TRAP TO LOCATION 4. THIS WOULD HAPPE
                            IF RACF (HALT:OP CD 7) DOES NOT GO HIGH.
                            STATE D12.01 WOULD CAUSE AN ODD ADDRESS TRAP SINCE R0
                            WILL CONTAIN A 1. THIS WILL HAPPEN IF RACE E7 IS BAD.
                            HLT.00 WILL CAUSE THE PROCESSOR TO HALT ON THE INSTRUCTI
                            UNDER TEST AND WILL  OCCUR IF RACF E17 IS BAD.
                            IF THE INSTRUCTION DOESN'T WORK THEN ONE OF THE RTI MACH
                            STATES IS BAD.

                            ROM FLOW-12,156,212,213,214,215,172
```

```
                3652    THE RTT WILL NOT BE TESTED HERE SINCE THE ONLY POSSIBLE FORK A
                        FAILURE WOULD CAUSE AN RTI TO BE EXECUTED. THE T BIT FUNCTIONS O
                        THE RTI & RTT ARE TESTED IN PART 2.

                3659    TEST 110        EMT AND TRAP

                                FORK A SHOULD NOT FAIL.
                                THE INSTRUCTIONS ARE EXECUTED AND THE STACK IS CHECKED T
                                VERIFY THAT EVERYTHING WORKED OK.

                                ROM FLOW-0,345,354,SVC.00-SVC.90

                3741    TEST 111        IOT

                                FORK A SHOULD NOT FAIL.

                3744
                                IF THE TRAP VECTOR LOGIC FAILS THE TRAP VECTOR COULD COM
                                OUT TO BE 0, 4, OR 24.

                                THE ONLY OTHER POSSIBLE FAILURE WOULD BE STATE TRP.01.
                                IF THIS STATE FAILS TO LOAD THE DR THE TRAP VECTOR WILL
                                BE WHATEVER IS IN R4.  IF IT FAILS TO LOAD THE BR THE OL
                                PS WILL FAIL TO BE STACKED.

                                ROM FLOW-14,354,(SVC.00-SVC.90) 355,65,357,360,367,37,25

                ;*****************************************************************
        3815    END OF PASS ROUTINE
                ;*****************************************************************

                3817    INCREMENT THE PASS NUMBER ($PASS)
                        INDICATE END-OF-PROGRAM AFTER 144 PASSES THRU THE PROGRAM
                        TYPE "END PASS"
                        IF THERES A MONITOR GO TO IT
                        IF THERE ISN'T JUMP TO TST1
                        IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCAT
                        $ENDMG CAN BE CHANGED TO 7.
```

```
                        PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0

                ;*****************************************************************
3851            ;TYPE ROUTINE
                ;*****************************************************************

3853            ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 B
                ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
                ;NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
                ;NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
                ;NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                ;
                ;CALL:
                ;1) USING A TRAP INSTRUCTION
                ;       TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN
                ;OR
                ;       TYPE
                ;       MESADR
                ;
                ;2) USING A JSR INSTRUCTION
                ;       MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON
                ;       JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
                ;       MESADDR                 ;;FIRST ADRESS OF MESSAGE

                ;*****************************************************************
3924            ;BINARY TO OCTAL (ASCII) AND TYPE
                ;*****************************************************************

3926            ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIG
                ;OCTAL (ASCII) NUMBER AND TYPE IT.
                ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS
                ;CALL:
                ;       MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;       TYPOS                   ;;CALL FOR TYPEOUT
                ;       .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS
                ;       .BYTE   M               ;;M=1 OR 0
                ;                               ;;1=TYPE LEADING ZEROS
                ;                               ;;0=SUPPRESS LEADING ZER
                ;
                ;$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE
                ;$TYPOS OR $TYPOC
                ;CALL:
                ;       MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;       TYPON                   ;;CALL FOR TYPEOUT
                ;
                ;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                ;CALL:
                ;       MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                ;       TYPOC                   ;;CALL FOR TYPEOUT
```

```
                                PDP 11/70 CPU DIAGNOSTIC PART 1                                 DECDOC VER  00.0

                        ;*****************************************************************
                4002    ;CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                        ;*****************************************************************

                        ;4004   THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIG
                        ;       SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER
                        ;       NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE T
                        ;       BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS
                        ;       REPLACED WITH SPACES.
                        ;       CALL:
                        ;               MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE S
                        ;               TYPDS                   ;;GO TO THE ROUTINE

                        ;*****************************************************************
                4070    ;TRAP DECODER
                        ;*****************************************************************

                        ;4072   THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTIO
                        ;       AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDR
                        ;       OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                        ;       GO TO THAT ROUTINE.

                        ;*****************************************************************
                4085    ;TRAP TABLE
                        ;*****************************************************************

                        ;4087   THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLE
                        ;       BY THE "TRAP" INSTRUCTION.
```

```
                                        !
                                .TITLE  PDP 11/70 CPU DIAGNOSTIC PART 1
                                ;*COPYRIGHT (C) JULY 21,1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY DONALD W. MONROE
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A5).
                                ;*
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
```

```
                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100            ;;FIRST ADDRESS OF THE STACK
001100                          KERSTK= STACK           ;;KERNEL STACK
000700                          SUPSTK= STACK-200       ;;SUPERVISOR STACK
000600                          USESTK= STACK-300       ;;USER STACK
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          SWR=    177570          ;;SWITCH REGISTER
177570                          DISPLAY=SWR

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE LINE FEED
000015                          CR=     15              ;;CODE CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R0,R10          ;;GENERAL REGISTER
                                .EQUIV  R1,R11          ;;GENERAL REGISTER
                                .EQUIV  R2,R12          ;;GENERAL REGISTER
                                .EQUIV  R3,R13          ;;GENERAL REGISTER
                                .EQUIV  R4,R14          ;;GENERAL REGISTER
                                .EQUIV  R5,R15          ;;GENERAL REGISTER
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                                .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                                .EQUIV  SP,USP          ;;USER STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
```

```
                        040000          SW14=   40000
                        020000          SW13=   20000
                        010000          SW12=   10000
                        004000          SW11=   4000
                        002000          SW10=   2000
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
```

```
                        RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                        TBITVEC=14              ;;"T" BIT
                        TRTVEC= 14              ;;TRACE TRAP
                        BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                        IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                        PWRVEC= 24              ;;POWER FAIL
                        EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                        TRAPVEC=34              ;;"TRAP" TRAP
                        TKVEC=  60              ;;TTY KEYBOARD VECTOR
                        TPVEC=  64              ;;TTY PRINTER VECTOR
                        CACHVEC=114             ;;CACHE ERROR INTERRUPT VECTOR
                        PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                        MMVEC=  250             ;;MEMORY MANAGEMENT VECTOR

                        .SBTTL  CACHE   REGISTER DEFINITIONS

                        LOADRS = 177740         ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
                        HIADRS = 177742         ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
                        MEMERR = 177744         ;;CACHE ERROR REGISTER
                        CONTRL = 177746         ;;MEMORY CONTROL REGISTER
                        MAINT  = 177750         ;;MEMORY MAINTENENCE REGISTER
                        HITMIS = 177752         ;;HIT MISS REGISTER "1" IMPLIES HIT IN CACHE

                        .SBTTL  CPU REGISTER DEFINITIONS

                        SIZELO = 177760         ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
                                                ;;TO GET TO THE LAST 32 WORDS OF MEMORY
                        SIZEHI = 177762         ;;HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
                                                ;;CURRENTLY ALL ZERO
                        SYSTID = 177764         ;;SYSTEM ID REGISTER
                        CPUERR = 177766         ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
                                                ;;THE TRAP TO ERRVEC (000004)



                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                        ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES

                        MMR0=   177572
                        MMR1=   177574
                        MMR2=   177576
                        MMR3=   172516
                        .EQUIV  MMR0,SR0
                        .EQUIV  MMR1,SR1
                        .EQUIV  MMR2,SR2
                        .EQUIV  MMR3,SR3

                        ;*USER "I" PAGE DESCRIPTOR REGISTERS

                        UIPDR0= 177600
```

```
                        177602          UIPDR1= 177602
                        177604          UIPDR2= 177604
                        177606          UIPDR3= 177606
                        177610          UIPDR4= 177610
                        177612          UIPDR5= 177612
                        177614          UIPDR6= 177614
                        177616          UIPDR7= 177616

                                        ;*USER "D" PAGE DESCRIPTOR REGISTORS

                        177620          UDPDR0= 177620
                        177622          UDPDR1= 177622
                        177624          UDPDR2= 177624
                        177626          UDPDR3= 177626
                        177630          UDPDR4= 177630
                        177632          UDPDR5= 177632
                        177634          UDPDR6= 177634
                        177636          UDPDR7= 177636

                                        ;*USER "I" PAGE ADDRESS REGISTERS

                        177640          UIPAR0= 177640
                        177642          UIPAR1= 177642
                        177644          UIPAR2= 177644
                        177646          UIPAR3= 177646
                        177650          UIPAR4= 177650
                        177652          UIPAR5= 177652
                        177654          UIPAR6= 177654
                        177656          UIPAR7= 177656

                                        ;*USER "D" PAGE ADDRESS REGISTERS

                        177660          UDPAR0= 177660
                        177662          UDPAR1= 177662
                        177664          UDPAR2= 177664
                        177666          UDPAR3= 177666
                        177670          UDPAR4= 177670
                        177672          UDPAR5= 177672
                        177674          UDPAR6= 177674
                        177676          UDPAR7= 177676

                                        ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS

                        172200          SIPDR0= 172200
                        172202          SIPDR1= 172202
                        172204          SIPDR2= 172204
                        172206          SIPDR3= 172206
                        172210          SIPDR4= 172210
                        172212          SIPDR5= 172212
                        172214          SIPDR6= 172214
                        172216          SIPDR7= 172216

                                        ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS

                        172220          SDPDR0= 172220
                        172222          SDPDR1= 172222
```

```
2261            172224                  SDPDR2= 172224
2262            172226                  SDPDR3= 172226
2263            172230                  SDPDR4= 172230
2264            172232                  SDPDR5= 172232
2265            172234                  SDPDR6= 172234
2266            172236                  SDPDR7= 172236
2267
2268                                    ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS
2269
2270            172240                  SIPAR0= 172240
2271            172242                  SIPAR1= 172242
2272            172244                  SIPAR2= 172244
2273            172246                  SIPAR3= 172246
2274            172250                  SIPAR4= 172250
2275            172252                  SIPAR5= 172252
2276            172254                  SIPAR6= 172254
2277            172256                  SIPAR7= 172256
2278
2279                                    ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS
2280
2281            172260                  SDPAR0= 172260
2282            172262                  SDPAR1= 172262
2283            172264                  SDPAR2= 172264
2284            172266                  SDPAR3= 172266
2285            172270                  SDPAR4= 172270
2286            172272                  SDPAR5= 172272
2287            172274                  SDPAR6= 172274
2288            172276                  SDPAR7= 172276
2289
2290                                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
2291
2292            172300                  KIPDR0= 172300
2293            172302                  KIPDR1= 172302
2294            172304                  KIPDR2= 172304
2295            172306                  KIPDR3= 172306
2296            172310                  KIPDR4= 172310
2297            172312                  KIPDR5= 172312
2298            172314                  KIPDR6= 172314
2299            172316                  KIPDR7= 172316
2300
2301                                    ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS
2302
2303            172320                  KDPDR0= 172320
2304            172322                  KDPDR1= 172322
2305            172324                  KDPDR2= 172324
2306            172326                  KDPDR3= 172326
2307            172330                  KDPDR4= 172330
2308            172332                  KDPDR5= 172332
2309            172334                  KDPDR6= 172334
2310            172336                  KDPDR7= 172336
2311
2312                                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
2313
2314            172340                  KIPAR0= 172340
2315            172342                  KIPAR1= 172342
2316            172344                  KIPAR2= 172344
```

```
                        172346          KIPAR3= 172346
                        172350          KIPAR4= 172350
                        172352          KIPAR5= 172352
                        172354          KIPAR6= 172354
                        172356          KIPAR7= 172356

                                        ;*KERNEL "D" PAGE ADDRESS REGISTERS

                        172360          KDPAR0= 172360
                        172362          KDPAR1= 172362
                        172364          KDPAR2= 172364
                        172366          KDPAR3= 172366
                        172370          KDPAR4= 172370
                        172372          KDPAR5= 172372
                        172374          KDPAR6= 172374
                        172376          KDPAR7= 172376



                                        .SBTTL  UNIBUS MAP REGISTER DEFINITIONS

                                        ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
                                        ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'


                        170200          MAPL00 = 170200
                        170202          MAPH00 = 170202
                        170204          MAPL01 = 170204
                        170206          MAPH01 = 170206
                        170210          MAPL02 = 170210
                        170212          MAPH02 = 170212
                        170214          MAPL03 = 170214
                        170216          MAPH03 = 170216
                        170220          MAPL04 = 170220
                        170222          MAPH04 = 170222
                        170224          MAPL05 = 170224
                        170226          MAPH05 = 170226
                        170230          MAPL06 = 170230
                        170232          MAPH06 = 170232
                        170234          MAPL07 = 170234
                        170236          MAPH07 = 170236
                        170240          MAPL10 = 170240
                        170242          MAPH10 = 170242
                        170244          MAPL11 = 170244
                        170246          MAPH11 = 170246
                        170250          MAPL12 = 170250
                        170252          MAPH12 = 170252
                        170254          MAPL13 = 170254
                        170256          MAPH13 = 170256
                        170260          MAPL14 = 170260
                        170262          MAPH14 = 170262
                        170264          MAPL15 = 170264
                        170266          MAPH15 = 170266
                        170270          MAPL16 = 170270
```

```
2373            170272                  MAPH16 = 170272
2374            170274                  MAPL17 = 170274
2375            170276                  MAPH17 = 170276
2376            170300                  MAPL20 = 170300
2377            170302                  MAPH20 = 170302
2378            170304                  MAPL21 = 170304
2379            170306                  MAPH21 = 170306
2380            170310                  MAPL22 = 170310
2381            170312                  MAPH22 = 170312
2382            170314                  MAPL23 = 170314
2383            170316                  MAPH23 = 170316
2384            170320                  MAPL24 = 170320
2385            170320                  MAPH24 = 170320
2386            170324                  MAPL25 = 170324
2387            170326                  MAPH25 = 170326
2388            170330                  MAPL26 = 170330
2389            170332                  MAPH26 = 170332
2390            170334                  MAPL27 = 170334
2391            170336                  MAPH27 = 170336
2392            170340                  MAPL30 = 170340
2393            170342                  MAPH30 = 170342
2394            170344                  MAPL31 = 170344
2395            170346                  MAPH31 = 170346
2396            170350                  MAPL32 = 170350
2397            170352                  MAPH32 = 170352
2398            170354                  MAPL33 = 170354
2399            170356                  MAPH33 = 170356
2400            170360                  MAPL34 = 170360
2401            170362                  MAPH34 = 170362
2402            170364                  MAPL35 = 170364
2403            170366                  MAPH35 = 170366
2404            170370                  MAPL36 = 170370
2405            170372                  MAPH36 = 170372
2406            170374                  MAPL37 = 170374
2407            170376                  MAPH37 = 170376
2408                                    .EQUIV  MAPL00,MAPL0
2409                                    .EQUIV  MAPH00,MAPH0
2410                                    .EQUIV  MAPL01,MAPL1
2411                                    .EQUIV  MAPH01,MAPH1
2412                                    .EQUIV  MAPL02,MAPL2
2413                                    .EQUIV  MAPH02,MAPH2
2414                                    .EQUIV  MAPL03,MAPL3
2415                                    .EQUIV  MAPH03,MAPH3
2416                                    .EQUIV  MAPL04,MAPL4
2417                                    .EQUIV  MAPH04,MAPH4
2418                                    .EQUIV  MAPL05,MAPL5
2419                                    .EQUIV  MAPH05,MAPH5
2420                                    .EQUIV  MAPL06,MAPL6
2421                                    .EQUIV  MAPH06,MAPH6
2422                                    .EQUIV  MAPL07,MAPL7
2423                                    .EQUIV  MAPH07,MAPH7
2424
2425
2426
2427
2428
```

```
                                .SBTTL  TRAP CATCHER

        000000                          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

                                .SBTTL  STARTING ADDRESS(ES)
        000200                          .=200

000200  000137  001202                  JMP     @#START         ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                ;;**************************************************************

                                .SBTTL          ACT11 HOOKS

                                ;*THE FOLLOWING LOCATIONS ARE SETUP TO BE USED WITH ACT11
                                ;*
                                ;*LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOCICAL
                                ;*END OF THE PROGRAM.
                                ;*LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
                                ;*AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BITS
                                ;*TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:
                                ;*
                                ;*      BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
                                ;*            =0 NO POWER FAIL DESIRED
                                ;*
                                ;*      BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
                                ;*            =0 RUN TIME IS NOT MEMORY SIZE DEPENDENT
                                ;*
                                ;*      BITS 13-0  MUST BE ZERO'S

        000204                  $SVPC=.                 ;;SAVE LOCATION COUNTER
        000046                  .=46                    ;;SET LOCATION COUNTER
000046  010732                  .WORD   $ENDAD          ;;SET LOC.46 TO ADDRESS $ENDAD
        000052                  .=52                    ;;SET LOCATION COUNTER
000052  000000                  .WORD   0               ;;SET LOC.52 TO ZERO
        000204                  .=$SVPC                 ;; RESTORE LOCATION COUNTER
```

```
                                        ;;*****************************************************************
                                        .SBTTL  COMMON TAGS

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

                001100                          .=1100

001100                                  $CMTAG:                 ;;START OF COMMON TAGS
001100  000000                          $PASS:  .WORD   0       ;;CONTAINS PASS COUNT
001102     000                          $TSTNM: .BYTE   0       ;;CONTAINS THE TEST NUMBER
001103     000                          $ERFLG: .BYTE   0       ;;CONTAINS ERROR FLAG
001104  000000                          $ICNT:  .WORD   0       ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                          $LPADR: .WORD   0       ;;CONTAINS SCOPE LOOP
001110  000000                          $LPERR: .WORD   0       ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                          $ERTTL: .WORD   0       ;;CONTAINS TOTAL ERRORS DETECTED
001114     000                          $ITEMB: .BYTE   0       ;;CONTAINS ITEM CONTROL BYTE
001115     001                          $ERMAX: .BYTE   1       ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                          $ERRPC: .WORD   0       ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                          $GDADR: .WORD   0       ;;CONTAINS  OF 'GOOD' DATA
001122  000000                          $BDADR: .WORD   0       ;;CONTAINS  OF 'BAD' DATA
001124  000000                          $GDDAT: .WORD   0       ;;CONTAINS 'GOOD' DATA
001126  000000                          $BDDAT: .WORD   0       ;;CONTAINS 'BAD' DATA
001130  000000  000000  000000                  .WORD   0,0,0   ;;RESERVED--NOT TO BE USED
001136  177560                          $TKS:   177560          ;;TTY KBD STATUS
001140  177562                          $TKB:   177562          ;;TTY KBD BUFFER
001142  177564                          $TPS:   177564          ;;TTY PRINTER STATUS REG.
001144  177566                          $TPB:   177566          ;;TTY PRINTER BUFFER REG.
001146     000                          $NULL:  .BYTE   0       ;;CONTAINS NULL CHARACTER FOR FILLS
001147     002                          $FILLS: .BYTE   2       ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001150     012                          $FILLC: .BYTE   12      ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001151     000                          $TPFLG: .BYTE   0       ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001152  000000                          $REGAD: .WORD   0       ;;CONTAINS THE  FROM
                                                                ;;WHICH  ($REG0) WAS OBTAINED
001154  000000                          $REG0:  .WORD   0       ;;CONTAINS (($REGAD)+0)
001156  000000                          $REG1:  .WORD   0       ;;CONTAINS (($REGAD)+2)
001160  000000                          $REG2:  .WORD   0       ;;CONTAINS (($REGAD)+4)
001162  000000                          $TMP0:  .WORD   0       ;;USER DEFINED
001164  000000                          $TMP1:  .WORD   0       ;;USER DEFINED
001166  000000                          $TMP2:  .WORD   0       ;;USER DEFINED
001170  000000                          $TMP3:  .WORD   0       ;;USER DEFINED
001172     077                          $QUES:  .ASCII  /?/     ;;QUESTION MARK
001173     015                          $CRLF:  .ASCII  <15>    ;;CARRIAGE RETURN
001174  000012                          $LF:    .ASCIZ  <12>    ;;LINE FEED
001176                                  $ERPSW:
001176  000000                          .WORD
001200  000000                          0
```

```
                        ;;*****************************************************************
                        .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


001202                  $ERRTB:
                        ;;*****************************************************************
                        ;*      FOLLOWING IS AN INDEX OF THE POSSIBLE FAILURES THAT
                        ;*      CAUSE A TRAP TO LOCATIONS 4, 10, 14, OR 24 OR THAT
                        ;*      CAUSE THE PROCESSOR TO HANG (H). NGT=NOT GETTING THRU
                        ;*
                        ;*   TEST NUMBER        EFFECT                          CAUSE
                        ;*
                        ;*      25                10                  RACH NEG.B*DMO NOT GOING HIGH
                        ;*      25                24                  RACH A2 RABOO NOT GOING LOW
                        ;*      25                H                   RACH E59(6) NOT GOING LOW
                        ;*                                            OR NGT RACL RADR07
                        ;*      26                H                   RACL RADROO NOT GOING LOW
                        ;*      26                4                   EITHER IRCC SM357 STUCK H
                        ;*                                            OR RACL E70 BAD
                        ;*      26                14                  IRCC CO RAB03 NOT GOING H OR
                        ;*                                            RACL RADR03 INPUT STUCK LOW
                        ;*      26                4                   SRC CONST ADDED 1 OR 3
                        ;*
                        ;*      27                H                   RACK RADR01 NOT GOING LOW
                        ;*
                        ;*      31                10                  RACK AO RAB01 NOT GOING LOW OR
                        ;*                                            NGT RACL RADR01
                        ;*      31                4                   RACK BRCAB05 NOT GOING LOW OR
                        ;*                                            NGT PACL RADR05
                        ;*                                            1$ ON TOP OF STACK
                        ;*      31                4                   DST CONST ADDED 1 OR 3
                        ;*      32                10                  RACE AO RABOO DOES NOT GO LOW
                        ;*
                        ;*      33                10                  RACE AO RAB02 DOES NOT GO LOW
                        ;*
                        ;*      34                10                  RACE E44 IS BAD
                        ;*      34                10                  IRCB K/CLASS STUXK LOW
                        ;*      34                H                   GRAB OBD(1) STUCK H OR NGT RACL E71
                        ;*      34                H                   GRAD DRMXOO STUCK H OR GRAB E50 BAD
                        ;*
                        ;*      35                10                  RACE BIN*SMO H DID NOT GO HIGH
                        ;*      35                10                  IR DECODE ROM WORD BAD
                        ;*
                        ;*      36                10                  RACE BIN*SMO FAILED
```

```
2570                    ;*      36              10      IR DECODE ROM WORD BAD
2571                    ;*
2572                    ;*      37              10      RACE E45 BAD
2573                    ;*
2574                    ;*      40              10      RACE AD RAB00 DOES NOT GO LOW
2575                    ;*      40              10      IRCB(JMP+JSR) IS STUCK LOW
2576                    ;*      40              H       IRCB FJ CLASS IS STUCK HIGH
2577                    ;*
2578                    ;*      41              10      RACE E45 IS BAD
2579                    ;*
2580                    ;*      42              10      RACE E33 IS BAD
2581                    ;*
2582                    ;*      43              10      RACH U/CLASS NOT GOING HIGH
2583                    ;*                              5$-2 ON TOP OF STACK
2584                    ;*      43              10      EITHER RACE E10 BAD OR
2585                    ;*                              RACE BIN NOT GOING H
2586                    ;*                              5$ ON TOP OF STACK
2587                    ;*
2588                    ;*      44              10      RACJ AFIR 14(1) NGT RACE E42
2589                    ;*      44              10      IRCB E38(6) STUCK HIGH
2590                    ;*
2591                    ;*      45              H       GRAB OBD(0) STUCK H OR NGT RACK E51
2592                    ;*
2593                    ;*      46              10      RACE E33 BAD
2594                    ;*
2595                    ;*      54              10      RACF E3 DOES NOT GO HIGH
2596                    ;*
2597                    ;*      56              10      PART PCLASS FIELD BAD IN IR DECODE ROM
2598                    ;*
2599                    ;*      60              10      PART PCLASS FIELD BAD IN IR DECODE ROM
2600                    ;*
2601                    ;*      62              10      IRCC C0 RAB03 STUCK H OR NGT RACL RADR03
2602                    ;*                              OR FORK C MUX INPUT B0 STUCK H
2603                    ;*
2604                    ;*      65              10      B FORK MUX SELECT STUCK LOW
2605                    ;*      65              H       IRCB B0 RAB04 NGT RADR05
2606                    ;*      65              H       B FORK MUX INPUT B0 STUCK H OR
2607                    ;*                              IRCC B0 RAB00 STUCK H
2608                    ;*      65              4       B FORK MUX INPUT B3 OR
2609                    ;*                              IRCB B0 RAB00 STUCK L
2610                    ;*      65              H       IRCB E46(10) STUCK L-MICRO ADR 170
2611                    ;*
2612                    ;*      67              10      RACE E35(1) BAD
2613                    ;*
2614                    ;*      70              10      B FORK MUX STROBE STUCK L (CHIP FAILURE)
2615                    ;*
2616                    ;*      75              10      RACE JMP+JSR+SWAB NOT GOING HIGH
2617                    ;*      75              10      IRCB E63 BAD-R5 CONTAINS "T67+2"
2618                    ;*
2619                    ;*      105             4       RACF (HALT:OP CODE 7) DOES NOT GO HIGH
2620                    ;*      105             4       RACE E7 BAD-ODD ADR BIT SET IN ERROR REG
2621                    ;*
2622                    ;*********************************************************************
2623                    ;*      THE FOLLOWING FIVE CONDITION CODE AND BRANCH TESTS ARE A
2624                    ;*      FUNCTION OF RACK F TRUE 1. SECTION 1 OF EACH TEST IS
2625                    ;*      DEPENDENT ON TRUE 1 NOT GOING HIGH, WHILE SECTION 2 IS
```

```
                                        ;*      DEPENDENT ON TRUE ONE GOING HIGH.
                                        ;*
001202  012737  000014  177746  START:  MOV     #14,@#CONTRL    ;FORCE MISSES IN CACHE
                                        ;;*****************************************************************
                                        ;*TEST 1        CCC*BRANCH THRU FET.13
                                        ;*
                                        ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
                                        ;*         SECTION 1
                                        ;*              BCS     E58(13,12)      L,H
                                        ;*              BMI     E59(10,11,9)    H,L,H
                                        ;*              BVS     E48(5,3,4)      H,L,H
                                        ;*              BLOS    E59(4,3,5)      H,L,H
                                        ;;*****************************************************************
001210  000257                  TST1:   CCC                     ;CC=0000
                                ;SECTION 1
001212  103404                          BCS     1$
001214  100403                          BMI     1$
001216  102402                          BVS     1$
001220  101401                          BLOS    1$
001222  103001                          BCC     TST2            ;;GO TO NEXT TEST
001224                          1$:
001224  000000                          HALT                    ;RACF TRUE 1 WENT HIGH OR
                                                                ;RACH A2 RAB02 IS NOT GOING HIGH
                                                                ;OR NOT GETTING THRU RACL RADR02
                                                                ;FOR LOOPING CHANGE TO "BR TST1" (771)
                                        ;;*****************************************************************
                                        ;*TEST 2        SEC*BRANCH THRU FET.13 AND FET.11
                                        ;*
                                        ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
                                        ;*         SECTION 1
                                        ;*              BMI     E58(13,12)      H,L
                                        ;*              BVS     E58(13,12)      H,L
                                        ;*         SECTION 2
                                        ;*              BCC     E58(13,12)      H,H
                                        ;;*****************************************************************
001226  000261                  TST2:   SEC                     ;CC=0001
                                ;SECTION 1
001230  100402                          BMI     1$
001232  102401                          BVS     1$
001234  100001                          BPL     2$              ;GO TO SECTION 2
001236                          1$:
001236  000000                          HALT                    ;RACF E58 FAILED
                                                                ;FOR LOOPING CHANGE TO "BR TST2" (773)
                                ;SECTION 2
001240  103001                  2$:     BCC     3$
001242  103401                          BCS     TST3            ;;GO TO NEXT TEST
001244                          3$:
001244  000000                          HALT                    ;EITHER RACF TRUE 1 DID NOT GO HIGH
                                                                ;OR IT DID NOT GET THRU RACH A2 RAB00
                                                                ;FOR LOOPING CHANGE TO "BR 2$" (775)
                                        ;;*****************************************************************
                                        ;*TEST 3        SEV*BRANCH THRU FET.13 AND FET.11
                                        ;*
                                        ;*      THE FOLLOWING IS A LIST OF PATTERNS PUT ON THE GATES OF TRUE 1:
                                        ;*         SECTION 1
                                        ;*              BCS     E48(5,3,4)      L,H,H
```

```
                                ;*              BMI     E48(5,3,4)      H,H,L
                                ;*          SECTION 2
                                ;*              BVC     E48(5,3,4)      H,H,H
                                ;;*************************************************************
001246  000257                  TST3:   CCC                             ;CC=0000
                                ;SECTION 1
001250  000262                          SEV                             ;CC=0010
001252  103402                          BCS     1$
001254  100401                          BMI     1$
001256  100001                          BPL     2$
001260                          1$:
001260  000000                          HALT                            ;RACF E48 FAILED
                                                                        ;FOR LOOPING CHANGE TO "BR TST3" (772)
                                ;SECTION 2
001262  102001                  2$:     BVC     3$
001264  102401                          BVS     TST4                    ;;GO TO NEXT TEST
001266                          3$:
001266  000000                          HALT                            ;EITHER RACF E48 OR E47(1) FAILED
                                                                        ;FOR LOOPING CHANGE TO "BR 2$" (775)
                                ;;*************************************************************
                                ;*TEST 4         SEZ*BRANCH THRU FET.13 AND FET.11
                                ;*
                                ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
                                ;*          SECTION 1
                                ;*              BCS     E59(4,3,5)      H,H,L
                                ;*              BMI     E59(4,3,5)      L,H,H
                                ;*          SECTION 2
                                ;*              BHI     E59(4,3,5)      H,H,H
                                ;;*************************************************************
001270  000257                  TST4:   CCC                             ;CC=0000
                                ;SECTION 1
001272  000264                          SEZ                             ;CC=0100
001274  103402                          BCS     1$
001276  100401                          BMI     1$
001300  102001                          BVC     2$                      ;GO TO SECTION 2
001302                          1$:
001302  000000                          HALT                            ;RACF E59 FAILED
                                                                        ;FOR LOOPING CHANGE TO "BR TST4" (772)
                                ;SECTION 2
001304  101001                  2$:     BHI     3$
001306  101401                          BLOS    TST5                    ;;GO TO NEXT TEST
001310                          3$:
001310  000000                          HALT                            ;EITHER RACF E59 OR E47(11) FAILED
                                                                        ;FOR LOOPING CHANGE TO "BR 2$" (775)
                                ;;*************************************************************
                                ;*TEST 5         SEN*BRANCH THRU FET.13 AND FET.11
                                ;*
                                ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 1:
                                ;*          SECTION 1
                                ;*              BLOS    E59(10,11,9)    H,H,L
                                ;*              BVS     E59(10,11,9)    L,H,H
                                ;*          SECTION 2
                                ;*              BPL     E59(10,11,9)    H,H,H
                                ;;*************************************************************
001312  000257                  TST5:   CCC                             ;CC=0000
                                ;SECTION 1
```

```
2738  001314  000270                    SEN                     ;CC=1000
2739  001316  101402                    BLOS    1$
2740  001320  102401                    BVS     1$
2741  001322  101001                    BHI     2$              ;GO TO NEXT SECTION
2742  001324                    1$:
2743  001324  000000                    HALT                    ;RACF E59 FAILED
2744                                                            ;FOR LOOPING CHANGE TO "BR TST5" (772)
2745                            ;SECTION 2
2746  001326  100001            2$:     BPL     3$
2747  001330  100401                    BMI     TST6            ;;GO TO NEXT TEST
2748  001332                    3$:
2749  001332  000000                    HALT                    ;EITHER RACF E59 OR E47(13) FAILED
2750                                                            ;FOR LOOPING CHANGE TO "BR 2$" (775)
2751                            ;;**************************************************************
2752                            ;*      THE FOLLOWING SEVEN TESTS ARE A FUNCTION OF RACF TRUE 2.
2753                            ;*SECTION 1 OF EACH TEST IS DEPENDENT ON TRUE 2 NOT GOING HIGH
2754                            ;*WHILE SECTION 2 IS DEPENDENT ON TRUE 2 GOING HIGH.
2755                            ;*
2756                            ;;**************************************************************
2757                            ;*TEST 6        BRANCHES THRU FET.13
2758                            ;*
2759                            ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
2760                            ;*         SECTION 1
2761                            ;*              BLE     E58(2,1)H,L     E59(13,1,2)L,H,H        E48(1,13,2)H,H,L
2762                            ;*              BLT     E59(13,1,2)L,H,H        E48(1,13,2)H,H,L        E58(10,9)L,H
2763                            ;*              BEQ     E58(2,1)H,L     E58(10,9)H,L
2764                            ;;**************************************************************
2765  001334  000257            TST6:   CCC                     ;CC=0000
2766  001336  003403                    BLE     1$
2767  001340  002402                    BLT     1$
2768  001342  001401                    BEQ     1$
2769  001344  003001                    BGT     TST7            ;;GO TO NEXT TEST
2770  001346                    1$:
2771  001346  000000                    HALT                    ;RACF TRUE 2 WENT HIGH
2772                                                            ;FOR LOOPING CHANGE TO "BR TST6" (772)
2773                            ;;**************************************************************
2774                            ;*TEST 7        BRANCH THRU FET.13 AND FET.12
2775                            ;*
2776                            ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
2777                            ;*         SECTION 1
2778                            ;*              BEQ     E48(1,13,2)     L,H,H
2779                            ;*         SECTION 2
2780                            ;*              BGT     E48(1,13,2)     H,H,H
2781                            ;;**************************************************************
2782  001350  000257            TST7:   CCC                     ;CC=0000
2783                            ;SECTION 1
2784  001352  000262                    SEV                     ;CC=0010
2785  001354  001401                    BEQ     1$
2786  001356  001001                    BNE     2$              ;GO TO SECTION 2
2787  001360                    1$:
2788  001360  000000                    HALT                    ;RACF E48 FAILED
2789                                                            ;FOR LOOPING CHANGE TO "BR TST7" (773)
2790                            ;SECTION 2
2791  001362  003001            2$:     BGT     3$
2792  001364  003401                    BLE     TST10           ;;GO TO NEXT TEST
2793  001366                    3$:
```

```
2794  001366  000000                          HALT                    ;EITHER RACF TRUE 2 DID NOT GO HIGH
2795                                                                  ;OR IT DID NOT GET THRU RACH A2 RAB01
2796                                                                  ;FOR LOOPING CHANGE TO "BR 2$" (775)
2797                                  ;;*****************************************************************
2798                                  ;*TEST 10       BRANCH THRU FET.13 AND FET.12
2799                                  ;*
2800                                  ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
2801                                  ;*         SECTION 1
2802                                  ;*              BLT     E58(2,1)        L,H
2803                                  ;*         SECTION 2
2804                                  ;*              BNE     E58(2,1)        H,H
2805                                  ;;*****************************************************************
2806  001370  000257                  TST10:  CCC                     ;CC=0000
2807                                  ;SECTION 1
2808  001372  000264                         SEZ                     ;CC=0100
2809  001374  002401                         BLT     1$
2810  001376  002001                         BGE     2$              ;GO TO SECTION 2
2811  001400                          1$:
2812  001400  000000                         HALT                    ;RACF E58 FAILED
2813                                                                  ;FOR LOOPING CHANGE TO "BR TST10" (773)
2814                                  ;SECTION 2
2815  001402  001001                  2$:    BNE     3$
2816  001404  001401                         BEQ     TST11           ;;GO TO NEXT TEST
2817  001406                          3$:
2818  001406  000000                         HALT                    ;EITHER RACF E58 OR E46(12) FAILED
2819                                                                  ;FOR LOOPING CHANGE TO "BR 2$" (775)
2820                                  ;;*****************************************************************
2821                                  ;*TEST 11       BRANCH THRU FET.13 AND FET.12
2822                                  ;*
2823                                  ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
2824                                  ;*         SECTION 1
2825                                  ;*              BEQ     E59(13,1,2)     H,H,L
2826                                  ;*         SECTION 2
2827                                  ;*              BGE     E59(13,1,2)     H,H,H
2828                                  ;;*****************************************************************
2829  001410  000257                  TST11:  CCC                     ;CC=0000
2830                                  ;SECTION 1
2831  001412  000270                         SEN                     ;CC=1000
2832  001414  001401                         BEQ     1$
2833  001416  001001                         BNE     2$              ;GO TO NEXT SECTION
2834  001420                          1$:
2835  001420  000000                         HALT                    ;RACF E59 FAILED
2836                                                                  ;FOR LOOPING CHANGE TO "BR TST11" (773)
2837                                  ;SECTION 2
2838  001422  002001                  2$:    BGE     3$
2839  001424  002401                         BLT     TST12           ;;GO TO NEXT TEST
2840  001426                          3$:
2841  001426  000000                         HALT                    ;EITHER RACF E59 OR E46(13) FAILED
2842                                                                  ;FOR LOOPING CHANGE TO "BR 2$" (775)
2843                                  ;;*****************************************************************
2844                                  ;*TEST 12       BRANCHES THRU FET.13 AND FET.12
2845                                  ;*
2846                                  ;*      THIS TEST PUTS THE FOLLOWING PATTERNS ON THE GATES OF TRUE 2:
2847                                  ;*         SECTION 1
2848                                  ;*              BEQ     E58(2,1)H,L     E58(10,9)H,L
2849                                  ;*              BLT     E59(13,1,2)H,L,H        E48(1,13,2)H,L,H        E58(10,9)L,H
```

```
001430  000262                  ;;*****************************************************************
                        TST12:  SEV                     ;CC=1010
                        ;SECTION 1
001432  001402                  BEQ     1$
001434  002401                  BLT     1$
001436  001001                  BNE     TST13           ;;GO TO NEXT TEST
001440                  1$:
001440  000000                  HALT                    ;RACF TRUE 2 WENT HIGH
                                                        ;FOR LOOPING CHANGE TO "BR TST12" (773)
                        ;;*****************************************************************
                        ;*TEST 13       UNIARY AND BINARY (SMO)
                        ;*
                        ;*      THE FOLLOWING TEST TESTS ALL THE E/CLASS
                        ;*      INSTRUCTIONS WITH A DESTINATION MODE OF 0
                        ;*      AND DESTINATION FIELD OF NOT 7. THIS CLASS CONSISTS
                        ;*      OF ALL THE UNIARY (EXCEPT NEG) INSTRUCTIONS AND
                        ;*      ALL THE BINARY INSTRUCTIONS WITH A SOURCE MODE
                        ;*      OF 0.
                        ;;*****************************************************************
001442  000277          TST13:  SCC
001444  000244                  CLZ                     ;CC'S=1011
001446  005000                  CLR     R0              ;R0=000000, CC'S=0100
001450  103403                  BCS     CLRR0
001452  102402                  BVS     CLRR0
001454  100401                  BMI     CLRR0
001456  001401                  BEQ     .+4
001460                  CLRR0:
001460  000000                  HALT                    ;ERROR, INCORRECT CC'S AFTER CLR
                                                        ;FOR LOOPING CHANGE TO "BR TST13" (770)
001462  005000                  CLR     R0
001464  000277                  SCC                     ;CC'S=1111
001466  000244                  CLZ                     ;R0=000000, CC'S=1011
001470  005700                  TST     R0              ;R0=000000, CC'S=0100
001472  103403                  BCS     TSTR0
001474  102402                  BVS     TSTR0
001476  100401                  BMI     TSTR0
001500  001401                  BEQ     .+4
001502                  TSTR0:
001502  000000                  HALT                    ;ERROR, INCORRECT CC'S AFTER TST
                                                        ;FOR LOOPING CHANGE TO "BR CLRR0+2" (767)
001504  005000                  CLR     R0
001506  000257                  CCC                     ;CC'S=0000
001510  000266                  +SEZ!SEV                ;R0=000000, CC'S=0110
001512  005100                  COM     R0              ;R0=177777, CC'S=1001
001514  100003                  BPL     COMR0
001516  001402                  BEQ     COMR0
001520  102401                  BVS     COMR0
001522  103401                  BCS     .+4
001524                  COMR0:
001524  000000                  HALT                    ;ERROR, INCORRECT CC'S AFTER COM
                                                        ;FOR LOOPING CHANGE TO "BR TSTR0+2" (767)
001526  005000                  CLR     R0
001530  000277                  SCC                     ;R0=000000, CC'S=1111
```

```
2906  001532  005500                        ADC     R0              ;R0=000001, CC'S=0000
2907  001534  100403                        BMI     ADCR0
2908  001536  001402                        BEQ     ADCR0
2909  001540  102401                        BVS     ADCR0
2910  001542  103001                        BCC     .+4
2911  001544                          ADCR0:
2912  001544  000000                        HALT                    ;ERROR, INCORRECT CC'S AFTER ADC
2913                                                                ;FOR LOOPING CHANGE TO "BR COMR0+2" (770)
2914
2915  001546  005000                        CLR     R0
2916  001550  000261                        SEC
2917  001552  005500                        ADC     R0
2918  001554  000257                        CCC
2919  001556  000270                        SEN                     ;R0=000001, CC'S=1000
2920  001560  006000                        ROR     R0              ;R0=000000, CC'S=0111
2921  001562  100403                        BMI     RORR0
2922  001564  001002                        BNE     RORR0
2923  001566  102001                        BVC     RORR0
2924  001570  103401                        BCS     .+4
2925  001572                          RORR0:
2926  001572  000000                        HALT                    ;ERROR, INCORRECT CC'S AFTER ROR
2927                                                                ;FOR LOOPING CHANGE TO "BR ADCR0+2" (765)
2928
2929  001574  005000                        CLR     R0
2930  001576  000277                        SCC
2931  001600  000250                        CLN                     ;R0=000000, CC'S=0111
2932  001602  005300                        DEC     R0              ;R0=177777, CC'S=1001
2933  001604  100003                        BPL     DECR0
2934  001606  001402                        BEQ     DECR0
2935  001610  102401                        BVS     DECR0
2936  001612  103401                        BCS     .+4
2937  001614                          DECR0:
2938  001614  000000                        HALT                    ;ERROR, INCORRECT CC'S AFTER DEC
2939                                                                ;FOR LOOPING CHANGE TO "BR RORR0+2" (767)
2940
2941  001616  005000                        CLR     R0
2942  001620  000277                        SCC                     ;R0=000000, CC'S=1111
2943  001622  005200                        INC     R0              ;R0=000001, CC'S=0000
2944  001624  100403                        BMI     INCR0
2945  001626  001402                        BEQ     INCR0
2946  001630  102401                        BVS     INCR0
2947  001632  103401                        BCS     .+4
2948  001634                          INCR0:
2949  001634  000000                        HALT                    ;ERROR, INCORRECT CC'S AFTER INC
2950                                                                ;FOR LOOPING CHANGE TO "BR DECR0+2" (770)
2951
2952  001636  005000                        CLR     R0
2953  001640  000277                        SCC
2954  001642  000244                        CLZ                     ;R0=000000, CC'S=1011
2955  001644  006300                        ASL     R0              ;R0=000000, CC'S=0100
2956  001646  100403                        BMI     ASLR0
2957  001650  001002                        BNE     ASLR0
2958  001652  102401                        BVS     ASLR0
2959  001654  103001                        BCC     .+4
2960  001656                          ASLR0:
2961  001656  000000                        HALT                    ;ERROR, INCORRECT CC'S AFTER ASL
```

```
2962                                                    ;FOR LOOPING CHANGE TO "BR INCR0+2" (767)
2963
2964  001660  005000                    CLR     R0
2965  001662  000261                    SEC
2966  001664  006000                    ROR     R0        ;R0=100000, CC'S=1000
2967  001666  006100                    ROL     R0        ;R0=000000, CC'S=0111
2968  001670  100403                    BMI     ROLR0
2969  001672  001002                    BNE     ROLR0
2970  001674  102001                    BVC     ROLR0
2971  001676  103401                    BCS     .+4
2972  001700                    ROLR0:
2973  001700  000000                    HALT              ;ERROR, INCORRECT CC'S AFTER ROL
2974                                                    ;FOR LOOPING CHANGE TO "BR ASLR0+2" (767)
2975
2976  001702  005000                    CLR     R0
2977  001704  000261                    SEC
2978  001706  006000                    ROR     R0
2979  001710  005200                    INC     R0
2980  001712  000277                    SCC
2981  001714  000251                    +CLN!CLC          ;R0=100001, CC'S=0110
2982  001716  006200                    ASR     R0        ;R0=140000, CC'S=1001
2983  001720  100003                    BPL     ASRR0
2984  001722  001402                    BEQ     ASRR0
2985  001724  102401                    BVS     ASRR0
2986  001726  103401                    BCS     .+4
2987  001730                    ASRR0:
2988  001730  000000                    HALT              ;ERROR, INCORRECT CC'S AFTER ASR
2989                                                    ;FOR LOOPING CHANGE TO "BR RORR0+2" (721)
2990
2991  001732  005000                    CLR     R0
2992  001734  000277                    SCC
2993  001736  000250                    CLN               ;R0=000000, CC'S=0111
2994  001740  005600                    SBC     R0        ;R0=177777, CC'S=1001
2995  001742  100003                    BPL     SBCR0
2996  001744  001402                    BEQ     SBCR0
2997  001746  102401                    BVS     SBCR0
2998  001750  103401                    BCS     .+4
2999  001752                    SBCR0:
3000  001752  000000                    HALT              ;ERROR, INCORRECT CC'S AFTER SBC
3001                                                    ;FOR LOOPING CHANGE TO "BR ASRR0+2" (767)
3002
3003  001754  005000                    CLR     R0
3004  001756  000261                    SEC
3005  001760  006000                    ROR     R0
3006  001762  000277                    SCC
3007  001764  000250                    CLN               ;R0=100000, CC'S=0111
3008  001766  000300                    SWAB    R0        ;R0=000200, CC'S=1000
3009  001770  100003                    BPL     SWABR0
3010  001772  001402                    BEQ     SWABR0
3011  001774  102401                    BVS     SWABR0
3012  001776  103001                    BCC     .+4
3013  002000                    SWABR0:
3014  002000  000000                    HALT              ;ERROR, INCORRECT CC'S AFTER SWAB
3015                                                    ;FOR LOOPING CHANGE TO "BR SBCR0+2" (765)
3016
3017  002002  005000                    CLR     R0
```

```
3018  002004  005300                          DEC     R0
3019  002006  000257                          CCC
3020  002010  000262                          SEV                     ;R0=177777, CC'S=0010
3021  002012  006700                          SXT     R0              ;R0=000000, CC'S=0100
3022  002014  100403                          BMI     SXTR0
3023  002016  001002                          BNE     SXTR0
3024  002020  102401                          BVS     SXTR0
3025  002022  103001                          BCC     .+4
3026  002024                          SXTR0:
3027  002024  000000                          HALT                    ;ERROR, INCORRECT CC'S AFTER SXT
3028                                                                  ;FOR LOOPING CHANGE TO "BR SWABR0+2" (766)
3029  002026  005000                          CLR     R0
3030  002030  000270                          SEN                     ;R0=000000, CC'S=1XXX
3031  002032  006700                          SXT     R0              ;R0=177777, CC'S=1XXXX
3032  002034  005200                          INC     R0
3033  002036  001401                          BEQ     .+4
3034  002040                          SXT2:
3035  002040  000000                          HALT                    ;SIGN EXTEND FAILED WITH N SET
3036                                                                  ;FOR LOOPING CHANGE TO "BR SXTR0+2" (772)
3037
3038  002042  005000                          CLR     R0
3039  002044  005300                          DEC     R0
3040  002046  000277                          SCC
3041  002050  000244                          CLZ                     ;R0=177777, CC'S=1011
3042  002052  074000                          XOR     R0,R0           ;R0=000000, CC'S=0101
3043  002054  100403                          BMI     XORR0
3044  002056  001002                          BNE     XORR0
3045  002060  102401                          BVS     XORR0
3046  002062  103401                          BCS     .+4
3047  002064                          XORR0:
3048  002064  000000                          HALT                    ;ERROR, INCORRECT CC'S AFTER XOR
3049                                                                  ;FOR LOOPING CHANGE TO "BR SXT2+2" (766)
3050
3051  002066  005000                          CLR     R0
3052  002070  000261                          SEC
3053  002072  006000                          ROR     R0
3054  002074  000300                          SWAB    R0
3055  002076  000277                          SCC
3056  002100  000250                          CLN                     ;R0=000200, CC'S=0111
3057  002102  110000                          MOVB    R0,R0           ;R0=177600, CC'S=1001
3058  002104  100010                          BPL     MOVBR0
3059  002106  001407                          BEQ     MOVBR0
3060  002110  102406                          BVS     MOVBR0
3061  002112  103005                          BCC     MOVBR0
3062  002114  000250                          CLN
3063  002116  000264                          SEZ                     ;R0=177600, CC'S=0100
3064  002120  005700                          TST     R0              ;           CC'S=1000
3065  002122  100002                          BPL     MOVR0
3066  002124  001002                          BNE     .+6
3067  002126                          MOVBR0:
3068  002126  000000                          HALT                    ;ERROR, INCORRECT CC'S AFTER MOVB
3069                                                                  ;FOR LOOPING CHANGE TO "BR XORR0+2" (757)
3070  002130                          MOVR0:
3071  002130  000000                          HALT                    ;ERROR, MOVB DID NOT SIGN EXTEND
3072                                                                  ;FOR LOOPING CHANGE TO "BR XORR0+2" (756)
3073
```

```
3074  002132  005000                    CLR     R0
3075  002134  000277                    SCC
3076  002136  000244                    CLZ                     ;R0=000000, CC'S=1011
3077  002140  030000                    BIT     R0,R0           ;R0=000000, CC'S=0101
3078  002142  100403                    BMI     BITR0
3079  002144  001002                    BNE     BITR0
3080  002146  102401                    BVS     BITR0
3081  002150  103401                    BCS     .+4
3082  002152                    BITR0:
3083  002152  000000                    HALT                    ;ERROR, INCORRECT CC'S AFTER BIT
3084                                                            ;FOR LOOPING CHANGE TO "BR MOVR0+2" (767)
3085
3086  002154  005000                    CLR     R0
3087  002156  005200                    INC     R0
3088  002160  000277                    SCC
3089  002162  000244                    CLZ                     ;R0=000001, CC'S=1011
3090  002164  040000                    BIC     R0,R0           ;R0=000000, CC'S=0101
3091  002166  100403                    BMI     BICR0
3092  002170  001002                    BNE     BICR0
3093  002172  102401                    BVS     BICR0
3094  002174  103401                    BCS     .+4
3095  002176                    BICR0:
3096  002176  000000                    HALT                    ;ERROR, INCORRECT CC'S AFTER BIC
3097                                                            ;FOR LOOPING CHANGE TO "BR BITR0+2" (766)
3098
3099  002200  005000                    CLR     R0
3100  002202  005200                    INC     R0
3101  002204  000277                    SCC                     ;R0=000001, CC'S=1111
3102  002206  050000                    BIS     R0,R0           ;R0=000001, CC'S=0001
3103  002210  100403                    BMI     BISR0
3104  002212  001402                    BEQ     BISR0
3105  002214  102401                    BVS     BISR0
3106  002216  103401                    BCS     .+4
3107  002220                    BISR0:
3108  002220  000000                    HALT                    ;ERROR, INCORRECT CC'S AFTER BIS
3109                                                            ;FOR LOOPING CHANGE TO "BR BICR0+2" (767)
3110
3111  002222  005000                    CLR     R0
3112  002224  000261                    SEC
3113  002226  006000                    ROR     R0
3114  002230  006000                    ROR     R0
3115  002232  000277                    SCC
3116  002234  000252                    +CLN!CLV                ;R0=040000, CC'S=0101
3117  002236  060000                    ADD     R0,R0           ;R0=100000, CC'S=1010
3118  002240  100003                    BPL     ADDR0
3119  002242  001402                    BEQ     ADDR0
3120  002244  102001                    BVC     ADDR0
3121  002246  103001                    BCC     .+4
3122  002250                    ADDR0:
3123  002250  000000                    HALT                    ;ERROR, INCORRECT CC'S AFTER ADD
3124                                                            ;FOR LOOPING CHANGE TO "BR BISR0+2" (764)
3125
3126  002252  005000                    CLR     R0
3127  002254  005200                    INC     R0
3128  002256  000277                    SCC
3129  002260  000244                    CLZ                     ;R0=000001, CC'S=1011
```

```
3130  002262  160000                    SUB     R0,R0           ;R0=000000, CC'S=0100
3131  002264  100403                    BMI     SUBR0
3132  002266  001002                    BNE     SUBR0
3133  002270  102401                    BVS     SUBR0
3134  002272  103001                    BCC     .+4
3135  002274                    SUBR0:
3136  002274  000000                    HALT                    ;ERROR, INCORRECT CC'S AFTER SUB
3137                                                            ;FOR LOOPING CHANGE TO "BR ADDR0+2" (766)
3138
3139  002276  005000                    CLR     R0
3140  002300  000277                    SCC
3141  002302  000244                    CLZ                     ;R0=000000, CC'S=1011
3142  002304  020000                    CMP     R0,R0           ;R0=000000, CC'S=0100
3143  002306  100403                    BMI     CMPR0
3144  002310  001002                    BNE     CMPR0
3145  002312  102401                    BVS     CMPR0
3146  002314  103001                    BCC     .+4
3147  002316                    CMPR0:
3148  002316  000000                    HALT                    ;ERROR, INCORRECT CC'S AFTER CMP
3149                                                            ;FOR LOOPING CHANGE TO "BR SUBR0+2" (767)
3150                                    ;
3151                                    ;
3152                                    ;;*****************************************************************
3153                                    ;*TEST 14        REGISTER SELECTION TEST
3154                                    ;*
3155                                    ;*      THIS TEST ENSURES THAT THE 6 ADDRESS LINES INTO
3156                                    ;*      THE GENERAL PURPOSE REGISTERS (GPR) ARE NOT STUCK.
3157                                    ;*      THE LABELS OF THE ADDRESS LINES ARE:
3158                                    ;*              GSAX    GENERAL SOURCE ADDRESS LINE
3159                                    ;*              GDAX    GENERAL DESTINATION ADDRESS LINE
3160                                    ;*      WHERE X STANDS FOR LINE 0,1, OR 2.
3161                                    ;*      THE CLASSES OF ERRORS DESCRIBED IN THIS TEST
3162                                    ;*      ARE DEFINED AS FOLLOWS:
3163                                    ;*              CLASS A=GDAX OK
3164                                    ;*                      GSAX STUCK
3165                                    ;*              CLASS B=GSAX OK
3166                                    ;*                      GDAX STUCK
3167                                    ;*              CLASS C=GSAX STUCK
3168                                    ;*                      GDAX STUCK
3169                                    ;*****************************************************************
3170  002320  005000            TST14:  CLR     R0
3171  002322  005201                    INC     R1
3172  002324  005700                    TST     R0              ;DID INC AFFECT R0?
3173  002326  001406                    BEQ     OVER            ;BRANCH ON NOT CLASS B OR C
3174  002330  005000            ROUT0:  CLR     R0
3175  002332  005201                    INC     R1
3176  002334  010002                    MOV     R0,R2           ;DID S0 REMAIN 0 ON INC R1?
3177  002336  001001                    BNE     2$              ;BR IF YES-NOT CLASS B
3178  002340  000000                    HALT                    ;ERROR, CLASS B FAILURE ON GRA0
3179                                                            ;FOR LOOPING CHANGE TO "BR ROUT0" (773)
3180  002342                    2$:
3181  002342  000000                    HALT                    ;ERROR, CLASS C FAILURE ON GRA0
3182                                                            ;FOR LOOPING CHANGE TO "BR ROUT0" (772)
3183  002344  005201            OVER:   INC     R1              ;INCREMENT S1 AND D1
3184  002346  010001                    MOV     R0,R1           ;MOVE S0 TO S1 AND D1
3185  002350  001401                    BEQ     GRA1T           ;BRANCH-GRA0 OK
```

```
3186  002352  000000                  HALT                    ;ERROR, CLASS A FAILURE ON GRA0
3187                                                          ;FOR LOOPING CHANGE TO "BR TST14" (762)
3188
3189  002354  005000          GRA1T:  CLR     R0
3190  002356  005202                  INC     R2
3191  002360  005700                  TST     R0              ;DID INC AFFECT R2?
3192  002362  001406                  BEQ     OVER2           ;BRANCH ON NOT CLASS B OR C
3193  002364  005000          ROUT1:  CLR     R0
3194  002366  005202                  INC     R2
3195  002370  010001                  MOV     R0,R1           ;DID SO REMAIN 0 ON INC R2?
3196  002372  001001                  BNE     4$              ;BR IF YES-NOT CLASS B
3197  002374  000000                  HALT                    ;ERROR, CLASS B FAILURE ON GRA1
3198                                                          ;FOR LOOPING CHANGE TO "BR ROUT1" (773)
3199  002376                  4$:
3200  002376  000000                  HALT                    ;ERROR, CLASS C FAILURE ON GRA1
3201                                                          ;FOR LOOPING CHANGE TO "BR ROUT1" (772)
3202  002400  005202          OVER2:  INC     R2              ;INCREMENT S2 AND D2
3203  002402  010002                  MOV     R0,R2           ;MOVE S0 TO S2 AND D2
3204  002404  001401                  BEQ     GRA2T           ;BRANCH-GRA1 OK
3205  002406  000000                  HALT                    ;ERROR, CLASS A FAILURE ON GRA1
3206                                                          ;FOR LOOPING CHANGE TO "BR GRA1T" (762)
3207
3208  002410  005000          GRA2T:  CLR     R0
3209  002412  005204                  INC     R4
3210  002414  005700                  TST     R0              ;DID INC AFFECT R4?
3211  002416  001406                  BEQ     OVER3           ;YES, CLASS A
3212  002420  005000          ROUT2:  CLR     R0
3213  002422  005204                  INC     R4
3214  002424  010001                  MOV     R0,R1           ;DID SO REMAIN 0 AFTER INC R4?
3215  002426  001001                  BNE     6$              ;BR IF YES-NOT CLASS B
3216  002430  000000                  HALT                    ;ERROR, CLASS B FAILURE ON GRA2
3217                                                          ;FOR LOOPING CHANGE TO "BR ROUT2" (773)
3218  002432                  6$:
3219  002432  000000                  HALT                    ;ERROR, CLASS C FAILURE ON GRA2
3220                                                          ;FOR LOOPING CHANGE TO "BR ROUT2" (772)
3221  002434  005204          OVER3:  INC     R4              ;INCREMENT S4 AND D4
3222  002436  010004                  MOV     R0,R4           ;MOVE S0 TO S4 AND D4
3223  002440  001401                  BEQ     8$              ;;BRANCH IF OK
3224  002442  000000                  HALT                    ;ERROR, CLASS A FAILURE ON GRA2
3225                                                          ;FOR LOOPING CHANGE TO "BR GRA2T" (762)
3226  002444  005001          8$:     CLR     R1
3227  002446  005003                  CLR     R3
3228  002450  005005                  CLR     R5
3229  002452  005201                  INC     R1              ;CHANGE R1
3230  002454  005703                  TST     R3              ;DID R3 CHANGE?
3231  002456  001401                  BEQ     1$              ;BRANCH IF NO
3232  002460  000000                  HALT                    ;ADDRESS LINE 0 AND 1 TIED TOGETHER
3233                                                          ;FOR LOOPING CHANGE TO "BR 8$" (771)
3234  002462  010303          1$:     MOV     R3,R3           ;CHECK SRC ADDRESS LINES
3235  002464  001401                  BEQ     2$              ;BRANCH IF OK
3236  002466  000000                  HALT                    ;ADDRESS LINE 0 AND 1 TIED TOGETHER(SRC)
3237                                                          ;FOR LOOPING CHANGE TO "BR 8$" (766)
3238  002470  005705          2$:     TST     R5              ;DID R5 CHANGE?
3239  002472  001401                  BEQ     3$              ;BRANCH IF NO
3240  002474  000000                  HALT                    ;ADDRESS LINES 0 & 2 TIED TOGETHER(DST)
3241                                                          ;FOR LOOPING CHANGE TO "BR 8$" (763)
```

```
3242  002476  010505                 3$:     MOV     R5,R5
3243  002500  001401                         BEQ     4$
3244  002502  000000                         HALT                    ;ADDRESS LINES 0 & 2 TIED TOGETHER(SRC)
3245                                                                 ;FOR LOOPING CHANGE TO "BR 8$" (760)
3246  002504  005002                 4$:     CLR     R2
3247  002506  005006                         CLR     R6
3248  002510  005202                         INC     R2
3249  002512  005706                         TST     R6              ;DID R3 CHANGE?
3250  002514  001401                         BEQ     5$              ;BRANCH IF NO
3251  002516  000000                         HALT                    ;ADDRESS LINES 1 & 2 TIED TOGETHER(DST)
3252                                                                 ;FOR LOOPING CHANGE TO "BR 4$" (772)
3253  002520  010606                 5$:     MOV     R6,R6           ;GET R6 SRC
3254  002522  001401                         BEQ     TST15           ;;BRANCH IF SRC OK
3255  002524  000000                         HALT                    ;ADDRESS LINES 1 & 2 TIED TOGETHER(SRC)
3256                                                                 ;FOR LOOPING CHANGE TO "BR 4$" (767)
3257                                 ;;*****************************************************************
3258                                 ;*TEST 15       GPR1 STUCK BIT TEST
3259                                 ;*
3260                                 ;*     LOADS GPR1 WITH ZEROS AND ONES AND COMPARES R1 SOURCE AND
3261                                 ;*     DESTINATIONS WITH R0. IF THE COMPARISON FAILS A BIT IS STUCK.
3262                                 ;;*****************************************************************
3263  002526  005000                 TST15:  CLR     R0
3264  002530  005001                         CLR     R1
3265  002532  020001                         CMP     R0,R1           ;DID R1 DST CLR?
3266  002534  001401                         BEQ     1$              ;BRANCH IF YES
3267  002536  000000                         HALT                    ;ERROR, R1 SOURCE STUCK HIGH
3268                                                                 ;FOR LOOPING CHANGE TO "BR TST15" (773)
3269  002540  020100                 1$:     CMP     R1,R0           ;DID R1 SRC CLR?
3270  002542  001401                         BEQ     2$              ;BRANCH IF YES
3271  002544  000000                         HALT                    ;ERROR, R1 SOURCE STUCK HIGH
3272                                                                 ;FOR LOOPING CHANGE TO "BR TST15" (770)
3273  002546  005100                 2$:     COM     R0
3274  002550  005101                         COM     R1
3275  002552  020001                         CMP     R0,R1           ;DID R1 DST SET TO ALL ONES?
3276  002554  001401                         BEQ     3$              ;BRANCH IF YES
3277  002556  000000                         HALT                    ;ERROR, R1 DST STUCK LOW
3278                                                                 ;FOR LOOPING CHANGE TO "BR TST15" (763)
3279  002560  020100                 3$:     CMP     R1,R0           ;DID R1 SRC SET TO ALL ONES?
3280  002562  001401                         BEQ     TST16           ;;BRANCH IF YES
3281  002564  000000                         HALT                    ;ERROR, R1 SRC STUCK LOW
3282                                                                 ;FOR LOOPING CHANGE TO "BR TST15" (760)
3283                                 ;;*****************************************************************
3284                                 ;*TEST 16       GPR2 STUCK BIT TEST
3285                                 ;*
3286                                 ;*     LOADS GPR2 WITH ZEROS AND ONES AND COMPARES R2 SOURCE AND
3287                                 ;*     DESTINATION WITH R0.
3288                                 ;;*****************************************************************
3289  002566  005000                 TST16:  CLR     R0
3290  002570  005002                         CLR     R2
3291  002572  020200                         CMP     R2,R0           ;DID R2 SRC CLEAR?
3292  002574  001401                         BEQ     1$              ;BRANCH IF YES
3293  002576  000000                         HALT                    ;ERROR, R2 SRC STUCK HIGH
3294                                                                 ;FOR LOOPING CHANGE TO "BR TST16" (773)
3295  002600  020002                 1$:     CMP     R0,R2           ;DID R2 DST CLEAR?
3296  002602  001401                         BEQ     2$              ;BRANCH IF YES
3297  002604  000000                         HALT                    ;ERROR, R2 DST STUCK HIGH
```

```
                                                        ;FOR LOOPING CHANGE TO "BR TST16" (770)
002606  005100                  2$:     COM     R0
002610  005102                          COM     R2
002612  020002                          CMP     R0,R2           ;DID R2 DST SET?
002614  001401                          BEQ     3$              ;BRANCH IF YES
002616  000000                          HALT                    ;ERROR, R2 DST STUCK LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST16" (763)
002620  020200                  3$:     CMP     R2,R0           ;DID R2 SRC SET?
002622  001401                          BEQ     TST17           ;;BRANCH IF YES
002624  000000                          HALT                    ;ERROR, R2 SRC STUCK LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST16" (760)
                                ;;*****************************************************************
                                ;*TEST 17        GPR3 STUCK BIT TEST
                                ;*
                                ;*      LOADS GPR3 WITH ZEROS AND ONES AND COMPARES
                                ;*      R3 SOURCE AND DESTINATION WITH R0.
                                ;;*****************************************************************
002626  005000                  TST17:  CLR     R0
002630  005003                          CLR     R3
002632  020300                          CMP     R3,R0           ;DID R3 SRC CLEAR?
002634  001401                          BEQ     1$              ;BRANCH IF YES
002636  000000                          HALT                    ;ERROR, R3 SRC STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST17" (773)
002640  020003                  1$:     CMP     R0,R3           ;DID R3 DST CLEAR?
002642  001401                          BEQ     2$              ;BRANCH IF YES
002644  000000                          HALT                    ;ERROR, R3 DST STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST17" (770)
002646  005100                  2$:     COM     R0
002650  005103                          COM     R3
002652  020300                          CMP     R3,R0           ;DID R3 SRC SET TO ALL ONES?
002654  001401                          BEQ     3$              ;BRANCH IF YES
002656  000000                          HALT                    ;ERROR,R3 SRC STUCK LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST17" (763)
002660  020003                  3$:     CMP     R0,R3           ;DID R3 DST SET TO ALL ONES?
002662  001401                          BEQ     TST20           ;;BRANCH IF YES
002664  000000                          HALT                    ;ERROR, R3 DST STUCK LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST17" (760)
                                ;;*****************************************************************
                                ;*TEST 20        GPR4 STUCK BIT TEST
                                ;*
                                ;*      LOADS GPR4 WITH ZEROS AND ONES AND COMPARES
                                ;*      R4 SOURCE AND DESTINATION WITH R0.
                                ;;*****************************************************************
002666  005000                  TST20:  CLR     R0
002670  005004                          CLR     R4
002672  020400                          CMP     R4,R0           ;DID R4 SRC CLEAR?
002674  001401                          BEQ     1$              ;BRANCH IF YES
002676  000000                          HALT                    ;ERROR, R4 SRC STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST20" (773)
002700  020004                  1$:     CMP     R0,R4           ;DID R4 DST CLEAR?
002702  001401                          BEQ     2$              ;BRANCH IF YES
002704  000000                          HALT                    ;ERROR, R4 DST STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST20" (770)
002706  005100                  2$:     COM     R0
002710  005104                          COM     R4
002712  020004                          CMP     R0,R4           ;DID R4 DST SET?
```

```
3354  002714  001401                         BEQ     3$              ;BRANCH IF YES
3355  002716  000000                         HALT                    ;ERROR, R4 DST STUCK LOW
3356                                                                 ;FOR LOOPING CHANGE TO "BR TST20" (763)
3357  002720  020400                 3$:     CMP     R4,R0           ;DID R4 SRC SET?
3358  002722  001401                         BEQ     TST21           ;;BRANCH IF YES
3359  002724  000000                         HALT                    ;ERROR, R4 SRC STUCK LOW
3360                                                                 ;FOR LOOPING CHANGE TO "BR TST20" (760)
3361                                 ;;*****************************************************************
3362                                 ;*TEST 21       GPR5 STUCK BIT TEST
3363                                 ;*
3364                                 ;*      LOADS R5 WITH ZEROS AND ONES AND COMPARES
3365                                 ;*      R5 SOURCE AND DESTINATION WITH R0.
3366                                 ;;*****************************************************************
3367  002726  005000                 TST21:  CLR     R0
3368  002730  005005                         CLR     R5
3369  002732  020500                         CMP     R5,R0           ;DID R5 SRC CLEAR?
3370  002734  001401                         BEQ     1$              ;BRANCH IF YES
3371  002736  000000                         HALT                    ;ERROR, R5 SRC STUCK HIGH
3372                                                                 ;FOR LOOPING CHANGE TO "BR TST21" (773)
3373  002740  020005                 1$:     CMP     R0,R5           ;DID R5 DST CLEAR?
3374  002742  001401                         BEQ     2$              ;BRANCH IF YES
3375  002744  000000                         HALT                    ;ERROR, R5 DST STUCK HIGH
3376                                                                 ;FOR LOOPING CHANGE TO "BR TST21" (770)
3377  002746  005100                 2$:     COM     R0
3378  002750  005105                         COM     R5
3379  002752  020005                         CMP     R0,R5           ;DID R5 DST SET TO ALL ONES?
3380  002754  001401                         BEQ     3$              ;BRANCH IF YES
3381  002756  000000                         HALT                    ;ERROR, R5 DST STUCK LOW
3382                                                                 ;FOR LOOPING CHANGE TO "BR TST21" (763)
3383  002760  020500                 3$:     CMP     R5,R0           ;DID R5 SRC SET TO ALL ONES?
3384  002762  001401                         BEQ     TST22           ;;BRANCH IF YES
3385  002764  000000                         HALT                    ;ERROR, R5 SRC STUCK LOW
3386                                                                 ;FOR LOOPING CHANGE TO "BR TST21" (760)
3387                                 ;;*****************************************************************
3388                                 ;*TEST 22       GPR6 STUCK BIT TEST
3389                                 ;*
3390                                 ;*      LOADS R6 WITH ZEROS AND ONES AND COMPARES
3391                                 ;*      R6 SOURCE AND DESTINATION WITH R0.
3392                                 ;;*****************************************************************
3393  002766  005000                 TST22:  CLR     R0
3394  002770  005006                         CLR     R6
3395  002772  020006                         CMP     R0,R6           ;DID R6 DST CLEAR?
3396  002774  001401                         BEQ     1$              ;BRANCH IF YES
3397  002776  000000                         HALT                    ;ERROR, R6 DST STUCK HIGH
3398                                                                 ;FOR LOOPING CHANGE TO "BR TST22" (773)
3399  003000  020600                 1$:     CMP     R6,R0           ;DID R6 SRC CLEAR?
3400  003002  001401                         BEQ     2$              ;BRANCH IF YES
3401  003004  000000                         HALT                    ;ERROR, R6 SRC STUCK HIGH
3402                                                                 ;FOR LOOPING CHANGE TO "BR TST22" (770)
3403  003006  005000                 2$:     CLR     R0
3404  003010  005006                         CLR     R6
3405  003012  020006                         CMP     R0,R6           ;DID R6 DST SET?
3406  003014  001401                         BEQ     3$              ;BRANCH IF YES
3407  003016  000000                         HALT                    ;ERROR, R6 DST STUCK LOW
3408                                                                 ;FOR LOOPING CHANGE TO "BR TST22" (763)
3409  003020  020600                 3$:     CMP     R6,R0           ;DID R6 SRC SET?
```

```
3410  003022  001401                        BEQ     TST23           ;;BRANCH IF YES
3411  003024  000000                        HALT                    ;ERROR,R6 SRC STUCK LOW
3412                                                                  ;FOR LOOPING CHANGE TO "BR TST22" (760)
3413                                  ;;**************************************************************
3414                                  ;*TEST 23       GPR  SHORTED BIT TEST
3415                                  ;*
3416                                  ;*      TEST IF GPR'S 1 THRU 6 HAVE TWO BITS TIED TOGETHER
3417                                  ;*
3418                                  ;*      NOTE:   R0 IS CONSIDERED "HARDCORE"
3419                                  ;;**************************************************************
3420  003026  005000                  TST23:  CLR     R0              ;THIS SECTION OF CODE
3421  003030  005200                          INC     R0              ;   *
3422  003032  006100                          ROL     R0              ;   *
3423  003034  006100                          ROL     R0              ;   *
3424  003036  005200                          INC     R0              ;   *
3425  003040  006100                          ROL     R0              ;   *
3426  003042  006100                          ROL     R0              ;   *
3427  003044  005200                          INC     R0              ;   *
3428  003046  006100                          ROL     R0              ;   *
3429  003050  006100                          ROL     R0              ;   *
3430  003052  005200                          INC     R0              ;   *
3431  003054  006100                          ROL     R0              ;   *
3432  003056  006100                          ROL     R0              ;   *
3433  003060  005200                          INC     R0              ;   *
3434  003062  006100                          ROL     R0              ;   *
3435  003064  006100                          ROL     R0              ;   *
3436  003066  005200                          INC     R0              ;   *
3437  003070  006100                          ROL     R0              ;   *
3438  003072  006100                          ROL     R0              ;   *
3439  003074  005200                          INC     R0              ;   *
3440  003076  006100                          ROL     R0              ;   *
3441  003100  006100                          ROL     R0              ;   *
3442  003102  005200                          INC     R0              ;PUTS 52525 IN R0
3443  003104  010001                  2$:     MOV     R0,R1           ;PUT R0 SRC IN R1
3444  003106  020001                          CMP     R0,R1           ;IS R1 DST OK?
3445  003110  001401                          BEQ     3$              ;BRANCH IF YES
3446  003112  000000                          HALT                    ;ERROR, R1DST HAS SHORTED BITS
3447                                                                  ;FOR LOOPING CHANGE TO "BR TST15" (605)
3448  003114  020100                  3$:     CMP     R1,R0           ;IS R1 SRC OK?
3449  003116  001401                          BEQ     4$              ;BRANCH IF YES
3450  003120  000000                          HALT                    ;ERROR, R1 SRC HAS SHORTED BITS
3451                                                                  ;FOR LOOPING CHANGE TO "BR TST15" (602)
3452  003122  010002                  4$:     MOV     R0,R2
3453  003124  020002                          CMP     R0,R2           ;IS R2 DST OK?
3454  003126  001401                          BEQ     5$              ;BRANCH IF YES
3455  003130  000000                          HALT                    ;ERROR, R2 DST HAS SHORTED BITS
3456                                                                  ;FOR LOOPING CHANGE TO "BR TST15" (576)
3457  003132  020200                  5$:     CMP     R2,R0           ;IS R2 SRC OK?
3458  003134  001401                          BEQ     6$              ;BRANCH IF YES
3459  003136  000000                          HALT                    ;ERROR, R2 SRC HAS SHORTED BITS
3460                                                                  ;FOR LOOPING CHANGE TO "BR TST15" (573)
3461  003140  010003                  6$:     MOV     R0,R3           ;
3462  003142  020003                          CMP     R0,R3           ;IS R3 DST OK?
3463  003144  001401                          BEQ     7$              ;BRANCH IF YES
3464  003146  000000                          HALT                    ;ERROR, R3 DST HAS SHORTED BITS
3465                                                                  ;FOR LOOPING CHANGE TO "BR TST15" (567)
```

```
3466  003150  020300              7$:   CMP     R3,R0       ;IS R3 SRC OK?
3467  003152  001401                    BEQ     8$          ;BRANCH IF YES
3468  003154  000000                    HALT                ;ERROR R3 SRC HAS SHORTED BITS
3469                                                        ;FOR LOOPING CHANGE TO "BR TST15" (564)
3470  003156  010004              8$:   MOV     R0,R4       ;
3471  003160  020004                    CMP     R0,R4       ;IS R4 DST OK?
3472  003162  001401                    BEQ     9$
3473  003164  000000                    HALT                ;ERROR, R4 DST HAS SHORTED BITS
3474                                                        ;FOR LOOPING CHANGE TO "BR TST15" (560)
3475  003166  020400              9$:   CMP     R4,R0       ;IS R4 SRC OK?
3476  003170  001401                    BEQ     10$         ;BRANCH IF YES
3477  003172  000000                    HALT                ;ERROR, R4 SRC HAS SHORTED BITS
3478                                                        ;FOR LOOPING CHANGE TO "BR TST15" (555)
3479  003174  010005              10$:  MOV     R0,R5
3480  003176  020005                    CMP     R0,R5       ;IS R5 DST OK?
3481  003200  001401                    BEQ     11$         ;BRANCH IF YES
3482  003202  000000                    HALT                ;ERROR, R5 DST HAS SHORTED BITS
3483                                                        ;FOR LOOPING CHANGE TO "BR TST15" (551)
3484  003204  020500              11$:  CMP     R5,R0       ;IS R5 SRC OK?
3485  003206  001401                    BEQ     12$         ;BRANCH IF YES
3486  003210  000000                    HALT                ;ERROR, R5 SRC HAS SHORTED BITS
3487                                                        ;FOR LOOPING CHANGE TO "BR TST15" (546)
3488  003212  010006              12$:  MOV     R0,R6       ;
3489  003214  020006                    CMP     R0,R6       ;IS R6 DST OK?
3490  003216  001401                    BEQ     13$         ;BRANCH IF YES
3491  003220  000000                    HALT                ;ERROR, R6 DST HAS SHORTED BITS
3492                                                        ;FOR LOOPING CHANGE TO "BR TST15" (542)
3493  003222  020600              13$:  CMP     R6,R0       ;IS R6 SRC OK?
3494  003224  001401                    BEQ     14$         ;BRANCH IF YES
3495  003226  000000                    HALT                ;ERROR, R6 SRC HAS SHORTED BITS
3496                                                        ;FOR LOOPING CHANGE TO "BR TST15" (537)
3497  003230  005006              14$:  CLR     SP          ;THIS CODE PUTS
3498  003232  005206                    INC     SP          ;1100 IN THE SP
3499  003234  006106                    ROL     SP
3500  003236  006106                    ROL     SP
3501  003240  006106                    ROL     SP
3502  003242  005206                    INC     SP
3503  003244  006106                    ROL     SP
3504  003246  006106                    ROL     SP
3505  003250  006106                    ROL     SP
3506  003252  006106                    ROL     SP
3507  003254  006106                    ROL     SP
3508  003256  006106                    ROL     SP
3509                                ;;*****************************************************************
3510  003260  005000                    CLR     R0          ;THIS CODE
3511  003262  005200                    INC     R0          ;INITIALIZES
3512  003264  006100                    ROL     R0          ;THE
3513  003266  006100                    ROL     R0          ;TEST
3514  003270  006100                    ROL     R0          ;NUMBER
3515  003272  005200                    INC     R0
3516  003274  006100                    ROL     R0          ;STORAGE
3517  003276  005200                    INC     R0          ;REGISTER
3518  003300  012737  000044  177770    MOV     #44,@#177770 ;SETUP MICROPROCESSOR BREAK REG
3519                                    .SBTTL
3520                                ;;*****************************************************************
3521                                ;*TEST 24        ONE   MICROSTATE  (E/CLASS*DM0*DF7)
```

```
                                        ;*
                                        ;*      THIS TEST EXECUTES AN ADD INSTRUCTION WITH SMO,DMO, AND DF7.
                                        ;*      IF THE TEST FAILS THE SAME FLOW (EXC. 90) IS
                                        ;*      TRIED WITH A PENDING BRQ (T BIT TRAP) INSTEAD OF A DF7. IF THIS TEST FAILS
                                        ;*      A FORK A FAILURE IS REPORTED. IF IT PASSES, A TEST 1 FAILURE IS REPORTED.
                                        ;*
                                        ;*      ROM FLOW-30
                                        ;;*****************************************************************
003306  005200                  TST24:  INC     R0              ;INCREMENT TEST NUMBER
003310  005005                          CLR     R5              ;ENSURE R5 CLEAR
003312  005205                          INC     R5              ;SET R5 EQUAL
003314  006105                          ROL     R5              ;TO 52
003316  006105                          ROL     R5              ;WHICH
003320  005205                          INC     R5              ;WILL CAUSE
003322  006105                          ROL     R5              ;A JUMP
003324  006105                          ROL     R5              ;TO TESTCC
003326  005205                          INC     R5
003330  006105                          ROL     R5              ;TEST
003332                          SYNC24:
003332  000264                          SEZ                     ;ENSURE Z SET
003334                          IUT24:
003334  060507                          ADD     R5,PC           ;ADD SHOULD SKIP TO TAG TESTCC
                                ;FAILURE- TRY E/CLASS*BRQ*DMO
003336  012737  003366  000014          MOV     #FORKA,@#14     ;SETUP VECTOR FOR RETURN
003344  005005                          CLR     R5              ;ENSURE R5 CLEAR
                                ;;*****************************************************************
003346  012746  000020                  MOV     #BIT4,-(SP)     ;THIS CODE
003352  012746  003360                  MOV     #1$,-(SP)       ;SETS THE
003356  000006                          RTT                     ;T BIT
                                ;;*****************************************************************
003360  005205                  1$:     INC     R5              ;TRAP HERE IF FORK A OK
003362  005205                          INC     R5              ;WILL EXECUTE IF FORK A FAILED
003364  000240                          NOP                     ;ALLOW T BIT TRAP IF FORK FAILED
003366  012737  000016  000014 FORKA:  MOV     #16,@#14        ;RESTORE T BIT VECTOR
003374  062706  000004                  ADD     #4,SP           ;RESTORE SP
003400  006005                          ROR     R5              ;IS R5 1 OR 2?
003402  103401                          BCS     1$              ;BRANCH ON 1 (FORK A OK)
003404  000000                          HALT                    ;FORK A FAILURE INTO ROM STATE EXC.90
                                                                ;FOR LOOPING CHANGE TO "BR TST24+2" (741)
003406                          1$:
003406  000000                          HALT                    ;EITHER PCB DID NOT LOAD OR RACH DF7 STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST24+2" (740)
003410                          TESTCC:
003410  001001                          BNE     TST25           ;;GO TO NEXT TEST IF CCLD$ OK
003412  000000                          HALT                    ;STATE EXC.90 BAD
                                                                ;FOR LOOPING CHANGE TO "BR TST24+2" (736)
                                        .SBTTL
                                ;;*****************************************************************
                                ;*TEST 25        TWO   MICROSTATES (NEG*DMO)
                                ;*
                                ;*      THIS TEST EXECUTES A NEGATE INSTRUCTION WITH DMO.
                                ;*
                                ;*      IF FORK A FAILS EXECUTION WOULD GO TO EITHER RSD.00, ZAP.00,
                                ;*      OR FOP.00.
                                ;*      FOP.00 WILL CAUSE THE PROCESSOR TO HANG.
```

```
                                        ;*      RSD.00 WOULD CAUSE A TRAP TO LOCATION 10. THIS WOULD ONLY HAPPEN
                                        ;*      IF RACH NEG.B*DMO DID NOT GO HIGH.
                                        ;*      ZAP.00 WOULD CAUSE A TRAP TO LOCATION 24. THIS WOULD ONLY HAPPEN
                                        ;*      IF RACH A2 RABOO DID NOT GO LOW.
                                        ;*
                                        ;*      ROM FLOW-301,210
                                        ;;*****************************************************************
003414  005200                  TST25:  INC     R0              ;INCREMENT TEST NUMBER
003416  005004                          CLR     R4              ;INITIALIZE CC ERROR RECORD
003420  005005                          CLR     R5              ;SET UP R5
003422  005205                          INC     R5              ;FOR TEST
003424  000257                          CCC                     ;
003426                          SYNC25:
003426  000266                          +SEZ!SEV                ;CC'S=0110
003430                          IUT25:
003430  005405                          NEG     R5              ;EXECUTE NEGATE CC'S=1001
003432  000401                          BR      1$              ;GET OVER ERROR CALL
003434  000000                          HALT                    ;RACH E57(6) DOES NOT GO LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST25" (767)
003436  100003                  1$:     BPL     NEGR5
003440  001402                          BEQ     NEGR5
003442  102401                          BVS     NEGR5
003444  103401                          BCS     .+4
003446  005204                  NEGR5:  INC     R4              ;ERROR, INCORRECT CC'S AFTER NEG.
003450  005001                          CLR     R1              ;SET R1 TO
003452  005301                          DEC     R1              ;-1 WITHOUT NEGATING
003454  020501                          CMP     R5,R1           ;DID R5 GET -1?
003456  001414                          BEQ     R50K            ;BRANCH IF R5 OK
003460  005704                          TST     R4              ;IS THERE A CC PROBLEM?
003462  001405                          BEQ     3$              ;BRANCH IF NO
003464  022705  177776                  CMP     #177776,R5      ;DID NEG ONE'S COMPLEMENT?
003470  001001                          BNE     2$              ;BRANCH IF NO
003472  000000                          HALT                    ;CC'S BAD AND NEG.90 DID NOT ADD 1
                                                                ;FOR LOOPING CHANGE TO "BR TST25+2" (751)
003474                          2$:
003474  000000                          HALT                    ;CC'S BAD AND R5 BAD
                                                                ;FOR LOOPING CHANGE TO "BR TST25+2" (750)
003476  022705  177776          3$:     CMP     #177776,R5      ;DID NEGATE DO A ONE'S COMPLIMENT?
003502  001001                          BNE     4$              ;BRANCH IF NO
003504  000000                          HALT                    ;CC'S OK BUT NEG.90 DID NOT ADD 1
                                                                ;FOR LOOPING CHANGE TO "BR TST25+2" (744)
003506                          4$:
003506  000000                          HALT                    ;CC'S OK BUT R5 BAD
                                                                ;FOR LOOPING CHANGE TO "BR TST25+2" (743)
003510  005704                  R50K:   TST     R4              ;CC PROBLEM?
003512  001401                          BEQ     TST26           ;;GO TO NEXT TEST IF NO
003514  000000                          HALT                    ;NEGATE OK BUT INCORRECT CC'S
                                                                ;FOR LOOPING CHANGE TO "BR TST25+2" (740)
                                        .SBTTL
                                        ;;*****************************************************************
                                        ;*TEST 26        THREE MICROSTATES (BIN*SM1*DMO*-DF7*SR0(0)
                                        ;*
                                        ;*      IF FORK A FAILS EXECUTION WILL GO TO EITHER EXEC.80 OR D12.00.
                                        ;*      EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MICRO ADDRESS 343.
                                        ;*      THIS WILL ONLY HAPPEN IF RACL RADROO IS NOT GOING LOW DUE
                                        ;*      TO RACF A1 RABOO (AFIR59(1)*[-BIN+SM01]*U/CLASS).
```

```
3634                                    ;*      D12.00 WOULD MOV THE PC TO LOCATION 0.
3635                                    ;*
3636                                    ;*      IF FORK C FAILS EXECUTION WOULD GO FROM S13.10 TO D00.80 OR
3637                                    ;*      D45.01 OR S13.20 OR D12.00 OR JSR.10 OR ASC.80 OR RTI.50 OR ASH.20 OR FOP.50.
3638                                    ;*      D00.80 WOULD SWAP THE BYTES OF THE SOURCE OPERAND BEFORE
3639                                    ;*      PUTTING THEM IN R5.
3640                                    ;*      D45.01 WOULD MOVE THE PC TO LOCATION 0.
3641                                    ;*      S13.20 WILL EXECUTE A SM3 (AND NO AUTO INC) INSTR.  THIS WILL CAUSE
3642                                    ;*      AN ODD ADDRESS TRAP SINCE LOCATION POSERR CONTAINS AN ODD WORD.
3643                                    ;*      JSR.10 WOULD PUSH THE ADDR OF POSERR ONTO THE STACK.
3644                                    ;*      ASC.80 WILL HALT AT 8$.
3645                                    ;*      RTI.50 WILL CAUSE 1004XX TO BE PLACED IN THE PS WORD
3646                                    ;*      AND THE PROCESSOR WILL TRAP TO LOCATION 14.
3647                                    ;*      FOP.50 WILL ?????.
3648                                    ;*      ASH.20 WOULD CAUSE A BAD CC.
3649                                    ;*      IF THE SRC CONST FAILS IN STATE S13.00 AND ADDS 1 OR 3, AN ODD
3650                                    ;*      ADDRESS TRAP WILL OCCUR.
3651                                    ;*      IF THE SRC CONSTANT ADDS 2, THE ERROR AT 6$ WILL REPORT THE FAILURE.
3652                                    ;*
3653                                    ;*      ROM FLOW-21,27,205
3654                                    ;;*****************************************************************
3655  003516  005200                    TST26:  INC     R0              ;INCREMENT TEST NUMBER
3656  003520  005005                            CLR     R5              ;SETUP R5
3657  003522                            SYNC26:
3658  003522  000244                            CLZ                     ;ENSURE Z CLEAR TO CATCH A FAILURE TO ASC.80
3659  003524                            IUT26:
3660  003524  011705                            MOV     (PC), R5        ;MOVE 1004XX TO R5(XX MUST BE
3661                                                                    ;ODD TO CATCH A FAILURE TO S13.20)
3662  003526  100443                    POSERR: BMI     7$              ;BRANCH IF CC OK(ENSURE OFFSET IS ODD)
3663  003530  100441                            BMI     6$              ;BRANCH IF SRC CONST ADDED 2
3664                                    ;FAILURE
3665  003532  013767  177776  175436            MOV     @#PSW,$ERPSW    ;SAVE ERROR PSW
3666  003540  022737  003526  000000  1$:       CMP     #POSERR,@#0     ;DID FORK A GO TO D12.00 OR FORK C TO D45,01?
3667  003546  001006                            BNE     3$              ;BRANCH IF NO
3668                                    ;EITHER FORK A OR C FAILED-FIND OUT WHICH ONE
3669  003550  005005                            CLR     R5              ;SETUP R5
3670  003552  012501                            MOV     (R5)+,R1        ;TEST TO SEE IF FORK A OR FORK C FAILED
3671  003554  005705                            TST     R5              ;DID R5 INCREMENT
3672  003556  001401                            BEQ     2$              ;BRANCH IF NO
3673  003560  000000                            HALT                    ;FORK C FAILED,EITHER IRCC DMO NOT GETTING
3674                                                                    ;THRU TO RACL RADR07 OR IRCC DMO IS STUCK HIGH
3675                                                                    ;FOR LOOPING CHANGE TO "BR TST26+2" (757)
3676  003562                            2$:
3677  003562  000000                            HALT                    ;FORK A FAILED, RACH A1 RAB04 NOT GOING LOW
3678                                                                    ;DUE TO EITHER RACE:BF1=7 OR
3679                                                                    ;BF1=0 OR SM0 STUCK LOW OR RACH E11 BAD
3680                                                                    ;FOR LOOPING CHANGE TO "BR TST26+2" (756)
3681  003564  000305                    3$:     SWAB    R5              ;SETUP R5 TO TEST IF INSTR. WENT THRU D00.80
3682  003566  020537  003526                    CMP     R5,@#POSERR     ;DID INSTR GO THRU D00.3?
3683  003572  001001                            BNE     4$              ;BRANCH IF NO
3684  003574  000000                            HALT                    ;IRCC RAB00 NOT GETTING TO RACL RADR00
3685                                                                    ;FOR LOOPING CHANGE TO "BR TST26+2" (751)
3686  003576  026627  000010  003526  4$:       CMP     10(SP),#POSERR  ;DID FORK C GO TO JSR.10?
3687  003604  001001                            BNE     5$              ;BRANCH IF NO
3688  003606  000000                            HALT                    ;INPUT TO IRCC E40 PIN 5 STUCK HIGH
3689                                                                    ;FOR LOOPING CHANGE TO "BR TST26+2" (744)
```

```
3690  003610  032767  000017  175360  5$:     BIT     #17,$ERPSW      ;DID ALL CC'S CLEAR?
3691  003616  001404                          BEQ     9$              ;BRANCH IF YES
3692  003620  032767  000020  175350          BIT     #BIT4,$ERPSW    ;DID Z BIT SET?
3693  003626  001401                          BEQ     8$              ;BRANCH IF NO
3694  003630                          9$:
3695  003630  000000                          HALT                    ;BAD CONDITION CODE
3696                                                                  ;FOR LOOPING CHANGE TO "BR TST26+2" (733)
3697  003632                          8$:
3698  003632  000000                          HALT                    ;EITHER INPUT TO C MUX STUCK LOW OR MUX BAD OR
3699                                                                  ;;CO RABO1 STUCK LOW OR RACL RADRO1 INPUT STUCK LOW
3700                                                                  ;FOR LOOPING CHANGE TO "BR TST26+2" (732)
3701  003634                          6$:
3702  003634  000000                          HALT                    ;SRC CONST EQUAL TO 2 SHOULD BE 0
3703                                                                  ;FOR LOOPING CHANGE TO "BR TST26+2" (731)
3704  003636  000241                  7$:     CLC                     ;ENSURE C CLEAR
3705  003640  006105                          ROL     R5              ;CHECK IF R5 WAS LOADED
3706  003642  103401                          BCS     TST27           ;;GO TO NEXT TEST IF C SET
3707  003644  000000                          HALT                    ;ERROR, R5 DID NOT LOAD
3708                                                                  ;FOR LOOPING CHANGE TO "BR TST26+2" (725)
3709                                  ;;**************************************************************
3710                                  ;*TEST 27       THREE MICROSTATES (BIN*SM2*DMO*-DF7*SRO(0)
3711                                  ;*
3712                                  ;*      THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR THE
3713                                  ;*      SM. A WORD AND BYTE INSTRUCTION IS EXECUTED TO VERIFY THAT STATE
3714                                  ;*      S13.01 ADDS THE CORRECT SOURCE CONSTANT.
3715                                  ;*
3716                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO EXC.80.
3717                                  ;*      EXC.80 WILL HANG THE PROCESSOR IN THE PAUSE STATE AT MICRO ADDRESS 343.
3718                                  ;*      THIS WILL ONLY HAPPEN IF RACL RADRO1
3719                                  ;*      IS NOT GOING LOW DUE TO RACF A1 RABO1 (AFIR10(1)*U/CLASS).
3720                                  ;*
3721                                  ;*      ROM FLOW-22,27,205
3722                                  ;;**************************************************************
3723  003646  005200                  TST27:  INC     R0              ;INCREMENT TEST NUMBER
3724  003650  005005                          CLR     R5              ;SETUP R5
3725  003652  000240                  SYNC27: NOP
3726  003654                          IUT27:
3727  003654  012705                          MOV     (PC)+,R5        ;MOVE 000401 TO R5
3728  003656  000401                  ER25:   BR      3$
3729  003660  000412                          BR      4$
3730                                  ;FAILURE-SOURCE CONSTANT FAILED. TRY SM3
3731  003662  012705  011704          3$:     MOV     #SUBTAB,R5      ;GET ADDRESS OF LOCATION THAT CONTAINS ADDR.
3732  003666  010501                          MOV     R5,R1           ;SAVE R5 IN R1
3733  003670  013502                          MOV     @(R5)+,R2       ;EXECUTE AN SM3 INSTRUCTION
3734  003672  005201                          INC     R1              ;ADJUST R1 TO
3735  003674  005201                          INC     R1              ;LOOK LIKE R5
3736  003676  020501                          CMP     R5,R1           ;DID R5 AUTO INCREMENT?
3737  003700  001401                          BEQ     2$              ;BRANCH IF YES
3738  003702  000000                          HALT                    ;SOURCE CONST FAILURE ON IRC
3739                                                                  ;FOR LOOPING CHANGE TO "BR TST27+2" (762)
3740  003704                          2$:
3741  003704  000000                          HALT                    ;SRC CONST FAILURE EITHER ON IRC OR DAP
3742                                                                  ;FOR LOOPING CHANGE TO "BR TST27+2" (761)
3743  003706  010705                  4$:     MOV     PC,R5           ;SETUP R5 TO HOLD ADDRESS
3744  003710  010501                          MOV     R5,R1           ;SAVE R5 IN R1
3745  003712  112502                          MOVB    (R5)+,R2        ;TEST TO SEE IF SCR CONST=1 ON BYTE
```

```
3746  003714  005201                        INC     R1              ;SETUP R1 TO LOOK LIKE R5
3747  003716  020501                        CMP     R5,R1           ;DID R5 AUTOINCREMENT BY 1?
3748  003720  001410                        BEQ     TST30           ;BRANCH IF YES
3749                                ;FAILURE-SOURCE CONSTANT FAILED ON BYTE. TRY SM4
3750  003722  010705                        MOV     PC,R5           ;PUT ADDRESS IN R5
3751  003724  010501                        MOV     R5,R1           ;SAVE R5 IN R1
3752  003726  114502                        MOVB    -(R5),R2        ;EXECUTE AN SM4 INSTRUCTION
3753  003730  005301                        DEC     R1              ;ADJUST R1 TO LOOK LIKE R5
3754  003732  020501                        CMP     R5,R1           ;DID R5 AUTO DECREMENT?
3755  003734  001001                        BNE     5$              ;BRANCH IF NO
3756  003736  000000                        HALT                    ;IRCC SRCM2 STUCK HIGH INTO IRC E8
3757                                                                ;FOR LOOPING CHANGE TO "BR TST27+2" (744)
3758  003740                          5$:
3759  003740  000000                        HALT                    ;BYTE SRC CONST FAILURE EITHER ON IRC OR DAP
3760                                                                ;FOR LOOPING CHANGE TO "BR TST27+2" (743)
3761                                ;;**************************************************************
3762                                ;*TEST 30       ALU CARRY FUNCTIONAL TEST
3763                                ;*
3764                                ;*      THIS TEST DOES A COMPLETE CHECK OF THE ALU CARRY FUNCTIONS
3765                                ;*      THE FIRST SECTION ENSURES THAT ALL THE INPUT AND OUTPUT LINES ARE
3766                                ;*      OK AND THE REST OF THE TEST ENSURES THAT THE CARRY LOGIC IS OK.
3767                                ;*      FOLLOWING ARE THE LOGIC EQUATIONS FOR A 74S181 AND 74S182 THAT
3768                                ;*      DICTATED THE PATTERNS USED IN EACH SECTION:
3769                                ;*         74S181
3770                                ;*              G=A3*B3+A2*B2*(A3+B3)+A1*B1*(A2+B2)*(A3+B3)+A0*B0*(A1+B1)*(A2+B2)*(A3+B3
3771                                ;*              P=(A3+B3)*(A2+B2)*(A1+B1)*(A0+B0)
3772                                ;*              COUT=G+P*CIN
3773                                ;*         74S182
3774                                ;*              CX=G0+P0*CIN
3775                                ;*              CY=G1+P1*G0+P1*P0*CIN
3776                                ;*              CZ=G2+P2*G1+P2*P1*G0+P2*P1*P0*CIN
3777                                ;;**************************************************************
3778  003742  005200                TST30:  INC     R0
3779                                ;SECTION 1-INPUT/OUTPUT BIT TEST
3780  003744  012701  125252                MOV     #125252,R1      ;PUT DATA PATTERN IN R1
3781  003750  062701  052525                ADD     #52525,R1       ;ADD COMPLIMENT PATTERN
3782  003754  005101                        COM     R1              ;MAKE RESULT 0
3783  003756  001401                        BEQ     2$              ;BRANCH IF IT IS ZERO
3784  003760  000000                        HALT                    ;BIT FAILED DURING ADD
3785                                                                ;FOR LOOPING CHANGE TO "BR TST30+2" (771)
3786  003762  012701  052525          2$:   MOV     #52525,R1       ;PUT PATTERN IN R1
3787  003766  062701  125252                ADD     #125252,R1      ;ADD COMPLIMENT PATTERN
3788  003772  005101                        COM     R1              ;MAKE IT ZERO
3789  003774  001401                        BEQ     3$              ;BRANCH IF IT WENT TO ZERO
3790  003776  000000                        HALT                    ;BIT FAILED DURING ADD
3791                                                                ;FOR LOOPING CHANGE TO "BR 2$" (771)
3792                                ;;**************************************************************
3793                                ;SECTION 2-G=A3*B3
3794  004000  012702  167357          3$:   MOV     #167357,R2      ;PUT COMPLIMENT OF EXPECTED PATTERN IN R2
3795  004004  012701  104210                MOV     #104210,R1      ;PUT PATTERN IN R1
3796  004010  060101                        ADD     R1,R1           ;ADD IT TO ITSELF
3797  004012  103401                        BCS     4$              ;BRANCH IF DAPH COUT15 OK
3798  004014  000000                        HALT                    ;DAPH COUT15 DID NOT GO LOW
3799                                                                ;FOR LOOPING CHANGE TO "BR 3$" (771)
3800  004016  050201          4$:   BIS     R2,R1           ;MAKE R1 -1
3801  004020  005101                        COM     R1              ;MAKE IT ZERO
```

```
3802  004022  001401                        BEQ     5$              ;BRANCH IF IT IS
3803  004024  000000                        HALT                    ;A G LINE FAILED
3804                                                                ;FOR LOOPING CHANGE TO "BR 3$" (765)
3805                                  ;;*************************************************************
3806                                  ;SECTION 3-G=A2*B2*(A3+B3)
3807  004026  012701  146314         5$:     MOV     #146314,R1      ;PUT PATTERN IN R1
3808  004032  062701  042104                 ADD     #42104,R1       ;ADD PATTERN
3809  004036  050201                         BIS     R2,R1           ;MAKE R1 -1
3810  004040  005101                         COM     R1              ;MAKE IT ZERO
3811  004042  001401                         BEQ     6$              ;BRANCH IF IT IS ZERO
3812  004044  000000                         HALT                    ;A G LINE FAILED
3813                                                                ;FOR LOOPING CHANGE TO "BR 5$" (770)
3814  004046  012701  042104         6$:     MOV     #42104,R1       ;REVERSE INPUTS
3815  004052  062701  146314                 ADD     #146314,R1      ;TO ALU
3816  004056  050201                         BIS     R2,R1           ;MAKE R1 -1
3817  004060  005101                         COM     R1              ;MAKE IT ZERO
3818  004062  001401                         BEQ     7$              ;BRANCH IF IT IS
3819  004064  000000                         HALT                    ;A G LINE FAILED
3820                                                                ;FOR LOOPING CHANGE TO "BR 6$" (770)
3821                                  ;;*************************************************************
3822                                  ;SECTION 4-G=A1*B1*(A2+B2)*(A3+B3)
3823  004066  012701  167356         7$:     MOV     #167356,R1      ;PUT PATTERN IN R1
3824  004072  062701  021042                 ADD     #21042,R1       ;ADD PATTERN
3825  004076  050201                         BIS     R2,R1           ;MAKE R1 -1
3826  004100  005101                         COM     R1              ;MAKE IT ZERO
3827  004102  001401                         BEQ     8$              ;BRANCH IF IT IS
3828  004104  000000                         HALT                    ;A G LINE FAILED
3829                                                                ;FOR LOOPING CHANGE TO "BR 7$" (770)
3830  004106  012701  021042         8$:     MOV     #21042,R1       ;REVERSE INPUTS
3831  004112  062701  167356                 ADD     #167356,R1      ;TO THE ALU
3832  004116  050201                         BIS     R2,R1           ;MAKE R1 -1
3833  004120  005101                         COM     R1              ;MAKE IT ZERO
3834  004122  001401                         BEQ     9$              ;BRANCH IF IT IS
3835  004124  000000                         HALT                    ;A G LINE FAILED
3836                                                                ;FOR LOOPING CHANGE TO "BR 8$" (770)
3837                                  ;;*************************************************************
3838                                  ;SECTION 5-G=A0*B0*(A1+B1)*(A2+B2)*(A3+B3)
3839  004126  012701  177777         9$:     MOV     #-1,R1          ;PUT PATTERN IN R1
3840  004132  062701  010421                 ADD     #10421,R1       ;ADD PATTERN TO R1
3841  004136  050201                         BIS     R2,R1           ;MAKE R1 -1
3842  004140  005101                         COM     R1              ;MAKE IT ZERO
3843  004142  001401                         BEQ     10$             ;BRANCH IF IT IS
3844  004144  000000                         HALT                    ;A G LINE FAILED
3845                                                                ;FOR LOOPING CHANGE TO "BR 9$" (770)
3846  004146  012701  010421         10$:    MOV     #10421,R1       ;REVERSE INPUTS
3847  004152  062701  177777                 ADD     #-1,R1          ;TO THE ALU
3848  004156  050201                         BIS     R2,R1           ;MAKE R1 -1
3849  004160  005101                         COM     R1              ;MAKE IT ZERO
3850  004162  001401                         BEQ     11$             ;BRANCH IF IT IS
3851  004164  000000                         HALT                    ;A G LINE FAILED
3852                                                                ;FOR LOOPING CHANGE TO "BR 10$" (770)
3853                                  ;;*************************************************************
3854                                  ;SECTION 6-P OUTPUTS AND CX=P0*CIN, CY=P1*P0*CIN, CZ=P2*P1*P0*CIN
3855  004166  012701  177777         11$:    MOV     #-1,R1          ;PUT PATTERN IN R1
3856  004172  000261                         SEC                     ;SET C
3857  004174  005501                         ADC     R1              ;CAUSES CARRY TO GO ALL THE WAY
```

```
3858  004176  103401                  BCS     12$             ;BRANCH IF CARRY CAME OUT
3859  004200  000000                  HALT                    ;DAPH COUT15 DID NOT GO LOW
3860                                                          ;FOR LOOPING CHANGE TO "BR 11$" (772)
3861  004202  001401          12$:    BEQ     13$             ;BRANCH IF R1 WENT TO ZERO
3862  004204  000000                  HALT                    ;EITHER A P LINE OR THE 74S182 FAILED
3863                                                          ;FOR LOOPING CHANGE TO "BR 11$" (770)
3864                          ;;*****************************************************************
3865                          ;SECTION 7-CY=P1*G0
3866  004210  012701  000370  13$:    MOV     #370,R1         ;PUT PATTERN IN R1
3867  004214  062701  000010          ADD     #10,R1          ;ADD DATA TO R1
3868  004220  052701  177377          BIS     #177377,R1      ;MAKE R1 -1
3869  004224  005101                  COM     R1              ;MAKE IT ZERO
3870  004226  001401                  BEQ     14$             ;BRANCH IF IT WORKED
3871  004230  000000                  HALT                    ;DAPF E44 FAILED
3872                                                          ;FOR LOOPING CHANGE TO "BR 13$" (767)
3873                          ;;*****************************************************************
3874                          ;SECTION 8-CZ=P2*G1
3875  004232  012701  007600  14$:    MOV     #7600,R1        ;PUT DATA IN R1
3876  004236  062701  000200          ADD     #200,R1         ;ADD DATA TO R1
3877  004242  052701  167777          BIS     #167777,R1      ;MAKE R1 -1
3878  004246  005101                  COM     R1              ;MAKE IT ZERO
3879  004250  001401                  BEQ     15$             ;BRANCH IF IT WORKED
3880  004252  000000                  HALT                    ;DAPF E44 FAILED
3881                                                          ;FOR LOOPING CHANGE TO "BR 14$" (767)
3882                          ;;*****************************************************************
3883                          ;SECTION 9-CZ=P2*P1*G0
3884  004254  012701  007770  15$:    MOV     #7770,R1        ;PUT DATA IN R1
3885  004260  062701  000010          ADD     #10,R1          ;ADD DATA TO R1
3886  004264  052701  167777          BIS     #167777,R1      ;MAKE R1 -1
3887  004270  005101                  COM     R1              ;MAKE IT ZERO
3888  004272  001401                  BEQ     TST31           ;BRANCH IF IT WORKED
3889  004274  000000                  HALT                    ;DAPF E44 FAILED
3890                                                          ;FOR LOOPING CHANGE TO "BR 15$" (767)
3891                          ;;*****************************************************************
3892                          ;*TEST 31       THREE MICROSTATES (DAC*DM2*0/CLASS)
3893                          ;*
3894                          ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00. THIS WILL ONLY HAPPEN
3895                          ;*      IF RACE A0 RAB01 DOES NOT GO LOW OR DOES NOT GET THRU
3896                          ;*      TO RACL RADR01.  THIS WILL CAUSE A TRAP TO LOCATION 10.
3897                          ;*
3898                          ;*      IF BEN15 FAILS EXECUTION WILL GO TO D45.80.
3899                          ;*      THIS WILL CAUSE A TRAP TO 4 WITH THE ADDRESS OF 1$ ON THE STACK.
3900                          ;*
3901                          ;*      IF THE DESTINATION CONSTANT FAILS(ADDS 1 OR 3) IN STATE D12.60
3902                          ;*      AN ODD ADDRESS TRAP WILL OCCUR.
3903                          ;*      IF THE DST CONST ADDS 0, THE ERROR AT 3$ WILL REPORT THE FAILURE.
3904                          ;*      IF THE DESTINATION IS NOT LOADED WITH THE SOURCE, THE
3905                          ;*      ERROR AFTER 5$ WILL REPORT THE FAILURE.
3906                          ;*
3907                          ;*      ROM FLOW-2,155,312
3908                          ;;*****************************************************************
3909  004276  005200  TST31:  INC     R0              ;INCREMENT TEST NUMBER
3910  004300  012705          MOV     (PC)+,R5        ;PUT "BR 4$" IN R5
3911  004302  000401          .WORD   000401          ;CONTAINS BINARY OF "BR 4$"
3912  004304  000240  SYNC31: NOP
3913  004306          IUT:
```

```
3914  004304  010527                  1$:     MOV     R5,(PC)+        ;EXECUTE INSTRUCTION UNDER TEST
3915  004306  000401                          BR      4$              ;WILL EXECUTE THIS IF INSTR FAILS TO AUTO INC
3916  004310  000415                          BR      5$              ;AUTO INC OK, GO CHECK IF LOAD OK
3917                                  ;FAILURE-DESTINATION CONSTANT FAILED. TRY DM4.
3918  004312  012705  001164          4$:     MOV     #$TMP1,R5       ;PUT ADDRESS IN R5
3919  004316  010125                          MOV     R1,(R5)+        ;EXECUTE INSTRUCTION UNDER TEST
3920  004320  022705  001164                  CMP     #$TMP1,R5       ;DID R5 STAY THE SAME?
3921  004324  001006                          BNE     2$              ;BRANCH IF NO
3922  004326  010145                          MOV     R1,-(R5)        ;SEE IF AUTO DEC, WORKS
3923  004330  022705  001162                  CMP     #$TMP1-2,R5     ;DID R5 AUTO DEC?
3924  004334  001001                          BNE     3$              ;BRANCH IF NO
3925  004336  000000                          HALT                    ;AUTO DEC WORKS SO ROM STATE D12.60 PROBABLY BAD
3926                                                                  ;FOR LOOPING CHANGE TO "BR TST31+2" (757)
3927  004340                          3$:
3928  004340  000000                          HALT                    ;IRCD DSTCON=2 EITHER STUCK LOW OR
3929                                                                  ;NOT GETTING THRU KOMUX
3930                                                                  ;FOR LOOPING CHANGE TO "BR TST31+2" (756)
3931  004342                          2$:
3932  004342  000000                          HALT                    ;BEN15 OK & DST CONST OK BUT INSTR DOSEN'T WORK
3933                                                                  ;FOR LOOPING CHANGE TO "BR TST31+2" (755)
3934  004344  012705  004304          5$:     MOV     #1$,R5          ;GET ADDR OF INSTR UNDER TEST
3935  004350  011505                          MOV     (R5),R5         ;GET INSTR UNDER TEST
3936  004352  022705  000401                  CMP     #401,R5         ;DID AUTO DEC OCCUR?
3937  004356  001005                          BNE     8$              ;BRANCH IF NO
3938  004360  012705  010027                  MOV     #10027,R5       ;GET OP CODE OF INSTR UNDER TEST
3939  004364  010567  177714                  MOV     R5,1$           ;RESTORE INSTR UNDER TEST
3940  004370  000000                          HALT                    ;EITHER RACK BRCAB05 NOT GOING LOW OR
3941                                                                  ;IT IS NOT GETTING THRU RACL RADR05
3942                                                                  ;FOR LOOPING CHANGE TO "BR TST31+2" (742)
3943  004372  010527                  8$:     MOV     R5,(PC)+        ;TEST TO SEE IF (PC)+ WAS LOADED
3944  004374  100000                  7$:     .WORD   100000          ;STORAGE LOCATION FOR PREVIOUS INSTR.
3945  004376  012701  004374                  MOV     #7$,R1          ;PUT ADDRESS OF 7$ IN R1
3946  004402  011102                          MOV     (R1),R2         ;GET CONTENTS OF 7$
3947  004404  100001                          BPL     6$              ;BRANCH IF LOAD OK
3948  004406  000000                          HALT                    ;ERROR, (PC)+ DID NOT LOAD CORRECTLY
3949                                                                  ;FOR LOOPING CHANGE TO "BR TST31+2" (733)
3950  004410  012702  100000          6$:     MOV     #BIT15,R2       ;SET SIGN BIT IN R2
3951  004414  010221                          MOV     R2,(R1)+        ;RESTORE 7$
3952                                  ;;*****************************************************************
3953                                  ;*TEST 32       THREE MICROSTATES (DAC*DM1*0/CLASS)
3954                                  ;*
3955                                  ;*      THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT FOR THE DM.
3956                                  ;*      IF FORK A FAILS, EXECUTION WILL GO TO RSD.00.
3957                                  ;*      THIS WILL CAUSE A TRAP TO LOCATION 10 WITH AN ODD ADDRESS ERROR.
3958                                  ;*      THIS WILL ONLY HAPPEN IF RACE A0 RAB00 DOES NOT GO LOW OR
3959                                  ;*      DOES NOT GET TO RACL.
3960                                  ;*      EITHER E44 OR E6 IS BAD.
3961                                  ;*
3962                                  ;*      IF THE INSTRUCTION FAILS TO MOVE R5 TO THE PC INDIRECT,
3963                                  ;*      THE ERROR AFTER 1$ WILL REPORT THE FAILURE.
3964                                  ;*
3965                                  ;*      ROM FLOW-1,155,312
3966                                  ;*****************************************************************
3967  004416  005200                  TST32:  INC     R0              ;INCREMENT TEST NUMBER
3968  004420  012705                          MOV     (PC)+,R5        ;PUT BR IN R5
3969  004422  000401                          .WORD   000401          ;BINARY WORD FOR BR .+2
```

```
3970  004424  000240                  SYNC32: NOP
3971  004426                          IUT32:
3972  004426  010517                          MOV     R5,(PC)         ;EXECUTE INSTRUCTION UNDER TST
3973  004430  000240                  1$:     .WORD   240             ;SHOULD REPLACE THIS WITH BR 2$
3974  004432  000000                          HALT                    ;LOCATION POINTED TO BY PC DID NOT LOAD
3975                                                                  ;FOR LOOPING CHANGE TO "BR TST32+2" (772)
3976  004434  012705  000240        2$:     MOV     #240,R5         ;THIS CODE
3977  004440  012701  004430                MOV     #1$,R1          ;RESTORES THE NOP
3978  004444  010511                        MOV     R5,(R1)         ;AT LOCATION 1$
3979  004446  000264                        SEZ                     ;ENSURE Z SET
3980  004450  010517                        MOV     R5,(PC)         ;EXECUTE INSTRUCTION UNDER TEST
3981  004452  000000                        .WORD   0
3982  004454  001001                        BNE     TST33           ;;CC OK, GO TO NEXT TEST
3983  004456  000000                        HALT                    ;STATE D12.20 BAD
3984                                                                  ;FOR LOOPING CHANGE TO "BR TST32+2" (760)
3985                                  ;;**************************************************************
3986                                  ;*TEST 33       THREE MICROSTATES (DAC*DM4*0/CLASS)
3987                                  ;*
3988                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00.
3989                                  ;*      THIS WILL CAUSE A TRAP TO LOCATION 10. THIS WILL ONLY HAPPEN IF
3990                                  ;*      RACE A0 RAB02 IS NOT GOING LOW. EITHER RACE E33 OR
3991                                  ;*      E6(9&8) IS BAD.
3992                                  ;*
3993                                  ;*      IF THE DST CONST FAILS TO SUBTRACT 2, THE ERROR AT EITHER 1$
3994                                  ;*      OR 1$-2 WILL REPORT THE FAILURE.
3995                                  ;*
3996                                  ;*      IF BEN01 FAILS (CAUSED BY IRCD DM357 STUCK HIGH)
3997                                  ;*      EXECUTION WILL GO TO D10.00 WHICH WILL EXECUTE A MODE 5
3998                                  ;*      INSTEAD OF MODE 4.
3999                                  ;*
4000                                  ;*      IF THE DESTINATION IS NOT LOADED PROPERLY,
4001                                  ;*      THE ERROR AT 3$ WILL REPORT THE FAILURE.
4002                                  ;*
4003                                  ;*      ROM FLOW-4,122,157
4004                                  ;;**************************************************************
4005  004460  005200                  TST33:  INC     R0              ;INCREMENT TEST NUMBER
4006  004462  012702  000001                MOV     #1,R2           ;SETUP R2
4007  004466  012705  001166                MOV     #$TMP2,R5       ;PUT ADDRESS OF $TMP2 IN R5
4008  004472  010501                        MOV     R5,R1           ;SAVE R5
4009                                  ;;**************************************************************
4010                                  ;THESE THREE INSTRUCTIONS ARE USED TO CATCH IRCD DM357 STUCK HIGH
4011  004474  012703  001164                MOV     #$TMP1,R3       ;PUT ADDR OF $TMP1 IN R3
4012  004500  010513                        MOV     R5,(R3)         ;PUT ADDR. OF $TMP2 IN $TMP1
4013                                  ;;**************************************************************
4014  004502  000240                  SYNC33: NOP
4015  004504                          IUT33:
4016  004504  010245                          MOV     R2,-(R5)        ;EXECUTE INSTRUCTION UNDER TEST
4017  004506  020503                          CMP     R5,R3           ;DID R5 AUTO DECREMENT?
4018  004510  001411                          BEQ     4$              ;BRANCH IF YES.
4019  004512  012705  001166                MOV     #$TMP1+2,R5     ;PUT ADDR. IN R5
4020  004516  012701  001164                MOV     #$TMP1,R1       ;PUT ADDR IN R1
4021  004522  011145                        MOV     (R1),-(R5)      ;EXECUTE INSTRUCTION
4022  004524  020105                        CMP     R1,R5           ;DID R5 AUTO DECREMENT?
4023  004526  001401                        BEQ     1$              ;BRANCH IF YES
4024  004530  000000                        HALT                    ;IRCC DSTM4 STUCK HIGH
4025                                                                  ;FOR LOOPING CHANGE TO "BR TST33+2" (754)
```

```
4026  004532                    1$:
4027  004532  000000                    HALT                    ;IRCC DSTM4 OK BUT D45.00 DOES NOT DECREMENT
4028                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (753)
4029  004534  011505              4$:   MOV     (R5),R5           ;GET CONTENTS OF $TMP1
4030  004536  020205                    CMP     R2,R5             ;DID DEST. GET LOADED?
4031  004540  001005                    BNE     2$                ;BRANCH IF NO
4032  004542  005205                    INC     R5                ;ADJUST R5 TO EVEN ADDRESS
4033  004544  000264                    SEZ                       ;ENSURE Z SET
4034  004546  010245                    MOV     R2,-(R5)          ;EXECUTE INSTRUCTION UNDER TEST
4035  004550  001020                    BNE     TST34             ;CC'S OK
4036  004552  000000                    HALT                      ;STATE D10.40 BAD
4037                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (743)
4038  004554  011101              2$:   MOV     (R1),R1           ;GET CONTENTS OF $TMP2
4039  004556  020201                    CMP     R2,R1             ;DID A MODE 5 TAKE PLACE?
4040  004560  001001                    BNE     3$                ;BRANCH IF NO
4041  004562  000000                    HALT                      ;EITHER IRCD DM357 STUCK HIGH OR RACK E49(C1) BAD
4042                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (737)
4043  004564                    3$:
4044  004564  000000                    HALT                    ;DST($TMP1) DID NOT GET LOADED FROM SRC(R2)
4045                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (736)
4046                                    ;;**************************************************************
4047                                    ;*TEST 34       THREE MICROSTATES (DAC*DM1*TST.B*DR0(0))
4048                                    ;*
4049                                    ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4050                                    ;*      THIS WILL ONLY HAPPEN IF RA SAME?
4051  004566  001372                    BNE     2$                ;BRANCH IF NO
4052  004570  010145                    MOV     R1,-(R5)          ;SEE IF AUTO DEC. WORKS
4053  004572  022705  001162            CMP     #$TMP1-2,R5       ;DID R5 AUTO DEC?
4054  004576  001372                    BNE     3$                ;BRANCH IF NO
4055  004600  000000                    HALT                      ;AUTO DEC WORKS SO ROM STATE D12.60 PROBABLY BAD
4056                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (730)
4057  004602                    3$:
4058  004602  000000                    HALT                    ;IRCD DSTCON=2 EITHER STUCK LOW OR
4059                                                            ;NOT GETTING THRU KOMUX
4060                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (727)
4061  004604                    2$:
4062  004604  000000                    HALT                    ;BEN15 OK & DST CONST OK BUT INSTR DOSEN'T WORK
4063                                                            ;FOR LOOPING CHANGE TO "BR TST33+2" (726)
4064  004606  012705  004532      5$:   MOV     #1$,R5            ;GET ADDR OF INSTR UNDER TCE E44 IS BAD (AFIR53(1)*R/CLA
4065                                    ;*
4066                                    ;*      IF BEN15 FAILS EXECUTION WILL GO TO STATE D12.60.
4067                                    ;*
4068                                    ;*      IF B FORK FAILS (AFTER STATE D12.10) CAUSED BY IRCB
4069                                    ;*      K/CLASS STUCK LOW EXECUTION WILL GO TO STATE RSD.00 CAUSING A TRAP TO 10.
4070                                    ;*      IF IRCB B1 RAB00 IS STUCK HIGH EXECUTION WILL GO FROM
4071                                    ;*      D12.10 TO JSR.40. THIS WILL CAUSE THE PROCESSOR TO HANG.
4072                                    ;*      IF EITHER GRAB OBD(1) IS STUCK HIGH OR NOT GETTING THRU TO RACL E71,
4073                                    ;*      EXECUTION WILL GO TO STATE D12.30.
4074                                    ;*      IF GRAD DRMX00 IS STUCK HIGH OR GRAB E50 IS BAD, EXECUTION
4075                                    ;*      WILL GO TO D10.60 WHICH WILL  HANG THE PROCESSOR.
4076                                    ;*
4077                                    ;*      ROM FLOW-1,175,33
4078                                    ;;**************************************************************
4079  004612  005200              TST34:  INC     R0                ;INCREMENT TEST NUMBER
4080  004614  012702  001164            MOV     #$TMP1,R2         ;PUT ADDRESS OF $TMP1 IN R2
4081  004620  012705  177400            MOV     #177400,R5        ;PUT 177400 IN R5
```

```
4082  004624  010512                        MOV     R5,(R2)         ;PUT -1 IN $TMP1
4083  004626  000240              SYNC34:   NOP
4084  004630                     IUT34:
4085  004630  005712                        TST     (R2)            ;EXECUTE INSTRUCTION UNDER TEST
4086  004632                     1$:
4087  004632  100416                        BMI     TST35           ;;BRANCH IF INSTRUCTION SET CC'S
4088                             ;FAILURE-TRY ROM FLOW 1,175,31,132
4089  004634  012737  177777  001164        MOV     #-1,@#$TMP1     ;PUT -1 IN $TMP1
4090  004642  012705  001164                MOV     #$TMP1,R5       ;PUT ADDR OF $TMP1 IN R5
4091  004646  005215                        INC     (R5)            ;INCREMENT $TMP1
4092  004650  001401                        BEQ     2$              ;BRANCH IF INC WORKED
4093  004652  000000                        HALT                    ;EITHER D12.10 FAILED OR BEN15 FAILED
4094                                                                 ;FOR LOOPING CHANGE TO "BR TST34+2" (760)
4095  004654  022737  000000  001164  2$:   CMP     #0,@#$TMP1      ;DID $TMP1 GO TO ZERO?
4096  004662  001401                        BEQ     3$              ;BRANCH IF YES
4097  004664  000000                        HALT                    ;CANNOT DIAGNOSE ERROR
4098                                                                 ;FOR LOOPING CHANGE TO "BR TST34+2" (753)
4099  004666                     3$:
4100  004666  000000                        HALT                    ;TST.10 FAILED
4101                                                                 ;FOR LOOPING CHANGE TO "BR TST34+2" (752)
4102                             ;;*****************************************************************
4103                             ;*TEST 35       THREE MICROSTATES (DAC*DM1*BIT.B*DR0(0))
4104                             ;*
4105                             ;*      THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT A BIT INSTRUCTION IS USED.
4106                             ;*      IF FORK A FAILS RACE BIN*SM0 H FAILED.
4107                             ;*
4108                             ;*      IF FORK B FAILS THE INSTRUCTION DECODE ROM WORD IS BAD.
4109                             ;*
4110                             ;*      IF THE RESULTANT DATA IS BAD STATE TST.10 FAILED.
4111                             ;*
4112                             ;*      ROM FLOW-1,175,33
4113                             ;;*****************************************************************
4114  004670  005200             TST35:  INC     R0              ;INCREMENT TEST NUMBER
4115  004672  005005                     CLR     R5              ;CLEAR R5
4116  004674  012701  001164             MOV     #$TMP1,R1       ;PUT ADDR OF $TMP1 IN R1
4117  004700  010511                     MOV     R5,(R1)         ;CLEAR $TMP1.
4118  004702  012705  100000             MOV     #BIT15,R5       ;PUT 100000 IN R5
4119  004706  000240             SYNC35: NOP
4120  004710                     IUT35:
4121  004710  030511                     BIT     R5,(R1)         ;EXECUTE INSTRUCTION UNDER TEST
4122  004712  001401                     BEQ     TST36           ;;BRANCH IF CC OK
4123  004714  000000                     HALT                    ;STATE TST.10 FAILED
4124                                                             ;FOR LOOPING CHANGE TO "BR TST35+2" (766)
4125                             ;;*****************************************************************
4126                             ;*TEST 36       THREE MICROSTATES (DAC*DM1*CMP.B*DR0(0))
4127                             ;*
4128                             ;*      THIS TEST IS THE SAME AS THE PREVIOUS TWO TESTS
4129                             ;*      EXCEPT A CMP INSTRUCTION IS USED.
4130                             ;*
4131                             ;*      ROM FLOW-1,175,33
4132                             ;;*****************************************************************
4133  004716  005200             TST36:  INC     R0              ;INCREMENT TEST NUMBER
4134  004720  005005                     CLR     R5              ;CLEAR R5
4135  004722  012701  001164             MOV     #$TMP1,R1       ;PUT ADDR OF $TMP1 IN R1
4136  004726  010511                     MOV     R5,(R1)         ;CLEAR $TMP1
4137  004730             SYNC36:
```

```
4138  004730  000244                          CLZ                     ;ENSURE Z CLEAR
4139  004732                        IUT36:
4140  004732  020511                          CMP     R5,(R1)         ;EXECUTE INSTRUCTION UNDER TEST
4141  004734  001401                          BEQ     TST37           ;;BRANCH IF CC OK
4142  004736  000000                          HALT                    ;STATE TST.10 FAILED
4143                                                                  ;FOR LOOPING CHANGE TO "BR TST36+2" (770)
4144                                ;;*****************************************************************
4145                                ;*TEST 37       THREE MICROSTATES (DAC*DM2*TST.B*DR0(0))
4146                                ;*
4147                                ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4148                                ;*      THIS WILL ONLY HAPPEN IF RACE E45 IS BAD (AFIR04(1)*R/CLASS).
4149                                ;*
4150                                ;*      BEN15 & FORK B HAVE ALREADY BEEN TESTED
4151                                ;*      IF THE AUTO INC FAILS IT WILL BE DUE TO A BAD
4152                                ;*      FIELD IN ROM STATE D12.10.
4153                                ;*
4154                                ;*      ROM FLOW-2,175,33
4155                                ;;*****************************************************************
4156  004740  005200                TST37:  INC     R0              ;INCREMENT TEST NUMBER
4157  004742  005001                        CLR     R1              ;CLEAR R1
4158  004744  012705  001164               MOV     #$TMP1,R5       ;PUT ADDR. OF $TMP1 IN R5
4159  004750  010115                        MOV     R1,(R5)         ;CLEAR $TMP1
4160  004752  000240                SYNC37: NOP
4161  004754                        IUT37:
4162  004754  005725                        TST     (R5)+           ;EXECUTE INSTR. UNDER TEST
4163  004756  001006                        BNE     1$              ;BRANCH IF BAD CC
4164  004760  022705  001166               CMP     #$TMP1+2,R5     ;DID R5 AUTO INC?
4165  004764  001407                        BEQ     TST40           ;;BRANCH IF TEST OK
4166  004766  010567  174162               MOV     R5,$REG0        ;SAVE REG 0 FOR TYPEOUT
4167  004772  000000                        HALT                    ;NO AUTO INC STATE D12.10 BAD
4168                                                                ;FOR LOOPING CHANGE TO "BR TST37+2" (763)
4169  004774  013767  177776  174174  1$:   MOV     @#PSW,$ERPSW    ;SAVE PSW FOR TYPEOUT
4170  005002  000000                        HALT                    ;BAD CC
4171                                                                ;FOR LOOPING CHANGE TO "BR TST37+2" (757)
4172
4173
4174                                ;;*****************************************************************
4175                                ;*THE LOGICAL SEQUENCE WOULD NEXT TEST THE BIT.B AND CMP.B INSTRUCTIONS BUT
4176                                ;*THESE WILL NOT BE TESTED WITH INDIVIDUAL TESTS SINCE THEY DO NOT USE
4177                                ;*ANY HARDWARE THAT HAS NOT ALREADY BEEN TESTED.
4178                                ;;*****************************************************************
4179
4180
4181                                ;;*****************************************************************
4182                                ;*TEST 40       THREE MICROSTATES (JMP*DM1)
4183                                ;*
4184                                ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4185                                ;*      THIS WILL ONLY HAPPEN IF RACE A0 RAB00 DOES NOT GO LOW
4186                                ;*      (AFIR03(1)*[JMP+JSR+SWAB]).
4187                                ;*
4188                                ;*      A FORK B FAILURE WOULD BE ONE OF THE FOLLOWING:
4189                                ;*      IF IRCB (JMP+JSR) IS STUCK LOW EXECUTION WILL GO TO RSD.00 CAUSING
4190                                ;*      A TRAP TO 10.
4191                                ;*      IF IRCB IR(14:9) 04 IS STUCK LOW EXECUTION WILL
4192                                ;*      GO TO JSR.00 WHICH WILL EXECUTE A JSR INSTEAD OF A JMP.
4193                                ;*      IF IRCB B FORK MUX FAILS EXECUTION WILL GO TO
```

```
4194                                    ;*      FUP.00.
4195                                    ;*      IF IRCB FJ CLASS IS STUCK HIGH EXECUTION WILL GO TO
4196                                    ;*      STATE D12.00 AND THE JMP WON'T JUMP.
4197                                    ;*
4198                                    ;*      IF THE INSTRUCTION FAILS TO JUMP, STATE JMP.00 IS REPORTED AS BAD.
4199                                    ;*
4200                                    ;*      ROM FLOW-1,135,35
4201                                    ;;*****************************************************************
4202  005004  005200                   TST40:  INC     R0              ;INCREMENT TEST NUMBER
4203  005006  005001                           CLR     R1              ;ENSURE R1 CLEAR
4204  005010  012705  005032                   MOV     #JMP1ADR,R5     ;PUT JMP ADDR. IN R5
4205  005014  010502                           MOV     R5,R2           ;PUT JUMP ADDR IN R2
4206  005016  000240                   SYNC40: NOP
4207  005020                           IUT40:
4208  005020  000115                           JMP     (R5)            ;EXECUTE INSTRUCTION UNDER TEST
4209  005022  020205                           CMP     R2,R5           ;DID NEGATE OCCUR?
4210  005024  001001                           BNE     2$              ;BRANCH IF YES
4211  005026  000000                           HALT                    ;STATE JMP.00 DID NOT LOAD PCB OR BEN15 FAILED
4212                                                                    ;FOR LOOPING CHANGE TO "BR TST40+2" (767)
4213  005030                           2$:
4214  005030  000000                           HALT                    ;IRCB FJ/CLASS NOT GETTING THRU B FORK MUX
4215                                                                    ;FOR LOOPING CHANGE TO "BR TST40+2" (766)
4216  005032  005701                   JMP1ADR:TST     R1              ;IS R1 STILL ZERO?
4217  005034  001403                           BEQ     TST41           ;;BRANCH IF YES
4218  005036  062706  000002                   ADD     #2,SP           ;JSR OCCURRED RE-ADJUST SP
4219  005042  000000                           HALT                    ;RACB IR(14:9)04 STUCK LOW
4220                                                                    ;FOR LOOPING CHANGE TO "BR TST40+2" (761)
4221                                    ;;*****************************************************************
4222                                    ;*TEST 41        THREE MICROSTATES (JMP*DM2)
4223                                    ;*
4224                                    ;*      THIS TEST IS THE SAME AS THE PREVIOUS TEST EXCEPT THE DM=2.
4225                                    ;*      IF FORK A FAILS RAC E45 IS BAD (AFIR04(1)*[JMP+JSR+SWAB]).
4226                                    ;*
4227                                    ;*      ROM FLOW-2,135,35
4228                                    ;;*****************************************************************
4229  005044  005200                   TST41:  INC     R0              ;INCREMENT TEST NUMBER
4230  005046  012705  005056                   MOV     #JMP2ADR,R5     ;PUT ADDRESS OF 1$ IN R5
4231  005052  000240                   SYNC41: NOP
4232  005054                           IUT41:
4233  005054  000125                           JMP     (R5)+           ;EXECUTE INSTRUCTION UNDER TEST
4234  005056  022705  005060           JMP2ADR:CMP     #JMP2ADR+2,R5   ;DID R5 AUTO INC?
4235  005062  001401                           BEQ     TST42           ;;BRANCH IF YES
4236  005064  000000                           HALT                    ;NO AUTO INC
4237                                                                    ;FOR LOOPING CHANGE TO "BR TST41+2" (770)
4238                                    ;;*****************************************************************
4239                                    ;*TEST 42        THREE MICROSTATES (JMP*DM4)
4240                                    ;*
4241                                    ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4242                                    ;*      THIS WILL ONLY HAPPEN IF RACE E33(AFIR05(1)*[JMP+JSR+SWAB]) IS BAD.
4243                                    ;*
4244                                    ;*      ALL OTHER LOGIC HAS BEEN TESTED.
4245                                    ;*
4246                                    ;*      ROM FLOW-4,122,35
4247                                    ;;*****************************************************************
4248  005066  005200                   TST42:  INC     R0              ;INCREMENT TEST NUMBER
4249  005070  012701  005102                   MOV     #NEXTT,R1       ;PUT ADDRESS OF 1$+2 IN R1
```

```
 4250  005074  000240                SYNC42: NOP
 4251  005076                        IUT42:
 4252  005076  000141                        JMP     -(R1)           ;EXECUTE INSTRUCTION UNDER TEST
 4253  005100  000240                        NOP
 4254  005102  000240                NEXTT:  NOP
 4255                                ;;*****************************************************************
 4256                                ;*TEST 43       THREE MICROSTATES (SOB)
 4257                                ;*
 4258                                ;*      IF RACH U/CLASS IS NOT GOING HIGH EXECUTION WILL GO TO RSD.00
 4259                                ;*      CAUSING A TRAP TO LOCATION 10 WITH THE ADDRESS OF 5$-2
 4260                                ;*      ON THE TOP OF THE STACK.
 4261                                ;*      IF EITHER RACE E10 IS BAD OR RACE BIN DOES NOT GO HIGH
 4262                                ;*      EXECUTION WILL GO TO D67.01. EXECUTION WILL EVENTUALL GO
 4263                                ;*      TO RSD.00 AFTER STATE D10.60 AND THE STACKED PC WILL BE AT 5$.
 4264                                ;*      IF RACF E8 FAILS EXECUTION WILL GO TO ASC.10 WHICH WILL
 4265                                ;*      PERFORM AN ASHC*DM0 OPERATION.
 4266                                ;*
 4267                                ;*      IF THE SOB BRANCHES WHEN IT IS NOT SUPPOSE TO EITHER
 4268                                ;*      GRAE SR EQ ONE IS  STUCK HIGH OR RACK E63 IS BAD
 4269                                ;*      OR STATE SOB.10 DOES NOT RESTORE THE OLD PC.
 4270                                ;*
 4271                                ;*      IF THE SOB DOES NOT BRANCH WHEN IT IS SUPPOSE TO EITHER
 4272                                ;*      STATE SOB.00 IS BAD OR GRAE SR EQ ONE STUCK LOW OR
 4273                                ;*      RACK E63(C1) IS BAD.
 4274                                ;*
 4275                                ;*      IF THE REGISTER DOES NOT DECREMENT STATE SOB.20 IS BAD.
 4276                                ;*
 4277                                ;*      ROM FLOW-57,242/262.262
 4278                                ;;*****************************************************************
 4279  005104  005200                TST43:  INC     R0              ;INCREMENT TEST NUMBER
 4280  005106  012706  001100                MOV     #STACK,SP       ;INITIALIZE THE SP
 4281  005112  012705  007777                MOV     #7777,R5        ;SETUP R5
 4282  005116  000401                        BR      SKIP            ;SKIP NEXT INSTRUCTION
 4283  005120  000410                GOOD:   BR      BAD+2           ;LOCATION FOR SOB TO BRANCH TO
 4284  005122  005004                SKIP:   CLR     R4              ;SETUP R4
 4285  005124  000240                SYNC43: NOP
 4286  005126                        IUT43:
 4287  005126  077404                        SOB     R4,GOOD         ;EXECUTE SOB WITH BRANCH
 4288  005130  005705                        TST     R5              ;DID R5 CLEAR?
 4289  005132  001001                5$:     BNE     3$              ;BRANCH IF NO
 4290  005134  000000                        HALT                    ;RACF E8 IS BAD(BUF AFIR11(1)*U/CLASS)
 4291                                                                ;FOR LOOPING CHANGE TO "BR TST43+2" (764)
 4292  005136                        3$:
 4293  005136  000000                        HALT                    ;EITHER GRAE SR EQ ONE IS STUCK LOW
 4294                                                                ;OR RACK E63(C1) IS BAD OR SOB.10 IS BAD
 4295                                                                ;FOR LOOPING CHANGE TO "BR TST43+2" (763)
 4296  005140                        BAD:
 4297  005140  000000                        HALT                    ;SOB FAILED TO BRANCH. SOB.00 BAD
 4298                                                                ;EITHER GRAE SR EQ ONE STUCK HIGH OR
 4299                                                                ;RACH E63(C1) BAD OR SOB.00 BAD
 4300                                                                ;FOR LOOPING CHANGE TO "BR TST43+2" (762)
 4301  005142  005005                        CLR     R5              ;SET UP R5 FOR
 4302  005144  005205                        INC     R5              ;NO BRANCH CONDITION
 4303  005146  000240                SYNC43: NOP
 4304  005150                        IUT43:
 4305  005150  077505                        SOB     R5,BAD          ;EXECUTE SOB WITHOUT BRANCH
```

```
4306  005152  005705                    TST     R5              ;DID R5 DECREMENT?
4307  005154  001401                    BEQ     1$              ;BRANCH IF YES
4308  005156  000000                    HALT                    ;R5 DID NOT DECREMENT. SOB.20 BAD
4309                                                            ;FOR LOOPING CHANGE TO "BR BAD+2+2" (772)
4310  005160  005005            1$:     CLR     R5
4311  005162  005001                    CLR     R1
4312  005164  005201            2$:     INC     R1
4313  005166  077502                    SOB     R5,2$           ;CHECK ALL PATTERNS OF R5 FOR SOB
4314  005170  005705                    TST     R5              ;DID R5 GET ALL THE WAY BACK TO 0?
4315  005172  001002                    BNE     3$              ;BRANCH IF NO
4316  005174  005701                    TST     R1              ;DID R1 ROLL OVER?
4317  005176  001401                    BEQ     TST44           ;;BRANCH IF YES
4318  005200                    3$:
4319  005200  000000                    HALT                    ;THE SIGNAL "SR EQ ONE" FAILED
4320                                                            ;FOR LOOPING CHANGE TO "BR 1$" (767)
4321                                      .SBTTL
4322                            ;;******************************************************************
4323                            ;*TEST 44       FOUR  MICROSTATES (DAC*DM12*P/CLASS*DRO(0)
4324                            ;*
4325                            ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4326                            ;*      THIS WILL ONLY HAPPEN IF RACJ AFIR 14(1) DOES NOT
4327                            ;*      GET THRU RAC E42.
4328                            ;*
4329                            ;*      THE FOLLOWING COULD BE FORK B FAILURES:
4330                            ;*      IF IRCB B0 RAB00 IS STUCK HIGH OR NOT GETTING THRU
4331                            ;*      TO RACL RADR00 EXECUTION WILL GO TO EXC.90 WHICH WOULD
4332                            ;*      PUT THE RESULT INTO A REGISTER RATHER THAN MEMORY.
4333                            ;*      IF IRCB E38(6) IS STUCK HIGH EXECUTION WILL GO TO
4334                            ;*      RSD.00 CAUSING A TRAP TO 10.
4335                            ;*      IF THE BIC DOES NOT HAPPEN THEN EXC.00 IS BAD.
4336                            ;*
4337                            ;*      ROM FLOW-2,175,31,132
4338                            ;;******************************************************************
4339  005202  005200            TST44:  INC     R0              ;INCREMENT TEST NUMBER
4340  005204  012701  001164            MOV     #$TMP1,R1       ;PUT ADDRESS OF $TMP1 IN R1
4341  005210  005005                    CLR     R5              ;PUT A 1
4342  005212  005205                    INC     R5              ;IN R5
4343  005214  010511                    MOV     R5,(R1)         ;PUT A 1 IN $TMP1
4344  005216  000240            SYNC44: NOP
4345  005220                    ;UT44:
4346  005220  040521                    BIC     R5,(R1)+        ;EXECUTE INSTR. UNDER TEST
4347  005222  022701  001166            CMP     #$TMP1+2,R1     ;DID R1 AUTO INC?
4348  005226  001404                    BEQ     1$              ;BRANCH IF YES
4349  005230  005701                    TST     R1              ;DID INSTR GO THRU EXC.90?
4350  005232  001001                    BNE     2$              ;BRANCH IF NO
4351  005234  000000                    HALT                    ;EITHER IRCB B0 RAB00 IS STUCK HIGH
4352                                                            ;OR IT IS NOT GETTING THRU TO RACL RADR50
4353                                                            ;FOR LOOPING CHANGE TO "BR TST44+2" (763)
4354  005236                    2$:
4355  005236  000000                    HALT                    ;INSTRUCTION FAILED. CAN'T DETERMINE CAUSE
4356                                                            ;FOR LOOPING CHANGE TO "BR TST44+2" (762)
4357  005240  012701  001164    1$:     MOV     #$TMP1,R1       ;PUT ADDR OF $TMP1 IN R1
4358  005244  005711                    TST     (R1)            ;DID BIC WORK?
4359  005246  001401                    BEQ     3$              ;BRANCH IF YES
4360  005250  000000                    HALT                    ;STATE EXC.00 FAILED
4361                                                            ;FOR LOOPING CHANGE TO "BR TST44+2" (755)
```

```
 4362  005252  010511                  3$:     MOV     R5,(R1)         ;PUT 1 IN $TMP2 & CLEAR Z
 4363  005254  040521                          BIC     R5,(R1)+        ;EXECUTE INSTRUCTION UNDER TEST
 4364  005256  001401                          BEQ     TST45           ;;CC'S OK
 4365  005260  000000                          HALT                    ;STATE EXC.00 BAD
 4366                                                                  ;FOR LOOPING CHANGE TO "BR TST44+2" (751)
 4367                                  ;;**************************************************************
 4368                                  ;*TEST 45       FOUR  MICROSTATES (DAC*DM12*TST.B*DRO(1))
 4369                                  ;*
 4370                                  ;*      AFTER STATE D12.10 IF EITHER GRAB OBD(1) DOES NOT GO HIGH
 4371                                  ;*      OR DOES NOT GET THRU RACL E71 EXECUTION WILL GO
 4372                                  ;*      TO TST.10. IF EITHER GRAB OBD(0) IS STUCK HIGH OR NOT GETTING
 4373                                  ;*      THRU RACK E41, EXECUTION WILL GO TO D10.60. THIS WILL CAUSE
 4374                                  ;*      THE PROCESSOR TO HANG UP IN THE PAUSE STATE AT MICRO
 4375                                  ;*      ADDRESS 177.  IF THE TEST FAILS THEN STATE D12.30 FAILED.
 4376                                  ;*
 4377                                  ;*      ROM FLOW-1,175,137,33
 4378                                  ;;**************************************************************
 4379  005262  005200                  TST45:  INC     R0              ;INCREMENT TEST NUMBER
 4380  005264  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDRESS OD $TMP1 IN R5
 4381  005270  012701  100000                  MOV     #BIT15,R1       ;PUT NEGATIVE UPPER BYTE IN R1
 4382  005274  010115                          MOV     R1,(R5)         ;MOVE R1 TO $TMP1
 4383  005276  005205                          INC     R5              ;PUT ODD ADDR IN R5
 4384  005300  000240                  SYNC45: NOP
 4385  005302                          IUT45:
 4386  005302  105715                          TSTB    (R5)            ;EXECUTE INSTR UNDER TEST
 4387  005304  100401                          BMI     TST46           ;;TEST OK, GO TO NEXT TEST
 4388  005306  000000                          HALT                    ;EITHER STATE D12.30 FAILED OR
 4389                                                                  ;BEN05*FEN2 FAILED(SEE ABOVE)
 4390                                                                  ;FOR LOOPING CHANGE TO "BR TST45+2" (766)
 4391                                  ;;**************************************************************
 4392                                  ;*TEST 46       FOUR  MICROSTATES (DAC*DM4*TST.B*DRO(0))
 4393                                  ;*
 4394                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
 4395                                  ;*      THIS WILL ONLY OCCUR IF RACE E33(AFIR05(1)*R/CLASS) IS BAD.
 4396                                  ;*
 4397                                  ;*      AFTER D10.30 IF IRCB FJ/CLASS DOES NOT GET TO RACL E71
 4398                                  ;*      OR IF RACL E71 IS BAD EXECUTION WILL GO TO SVC.50.
 4399                                  ;*
 4400                                  ;*      IF THE INSTRUCTION DOESN'T WORK THEN D10.60 IS BAD.
 4401                                  ;*
 4402                                  ;*      ROM FLOW-4,122,177,33
 4403                                  ;;**************************************************************
 4404  005310  005200                  TST46:  INC     R0              ;INCREMENT TEST NUMBER
 4405  005312  012701  100000                  MOV     #BIT15,R1       ;SET SIGN BIT
 4406  005316  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDR OF $TMP1 IN R5
 4407  005322  010125                          MOV     R1,(R5)+        ;SET SIGN BIT IN $TMP1
 4408  005324  005015                          CLR     (R5)            ;ENSURE $TEMP2 CLEAR
 4409  005326  000240                  SYNC46: NOP
 4410  005330                          IUT46:
 4411  005330  005745                          TST     -(R5)           ;EXECUTE INSTRUCTION UNDER TEST
 4412  005332  100407                          BMI     TST47           ;;TEST OK, GO TO NEXT TEST
 4413  005334  022706  001160                  CMP     #$TMP1-4,SP     ;DID EXECUTION GO TO SVC.50?
 4414  005340  001003                          BNE     1$              ;BRANCH IF NO
 4415  005342  012706  001100                  MOV     #STACK,SP       ;RESTORE THE SP
 4416  005346  000000                          HALT                    ;BEN15*FEN2 FAILED(SEE ABOVE)
 4417                                                                  ;FOR LOOPING CHANGE TO "BR TST46+2" (761)
```

```
4418  005350                          1$:
4419  005350  000000                          HALT                    ;STATE D10.60 FAILED
4420                                                                  ;FOR LOOPING CHANGE TO "BR TST46+2" (760)
4421
4422
4423                                  ;;*****************************************************************
4424                                  ;*THE LOGICAL SEQUENCE HERE WOULD BE TO TEST THE BIT & CMP INSTRUCTIONS
4425                                  ;*BUT NO ADDITIONAL LOGIC IS TESTED BY THEM.
4426                                  ;;*****************************************************************
4427
4428
4429                                  ;;*****************************************************************
4430                                  ;*TEST 47       FOUR  MICROSTATES (DAC*DM6*0/CLASS)
4431                                  ;*
4432                                  ;*      FORK A SHOULD NOT FAIL ON THIS TEST.
4433                                  ;*
4434                                  ;*      BEN01 SHOULD NOT FAIL.
4435                                  ;*
4436                                  ;*      BEN15*FEN2 SHOULD NOT FAIL.
4437                                  ;*
4438                                  ;*      IF D67.00 FAILS TO INCREMENT THE PC AN RTI INSTRUCTION
4439                                  ;*      WILL BE EXECUTED.
4440                                  ;*      IF D67.00 FAILS TO CLOCK THE BR THE INSTRUCTION
4441                                  ;*      WILL BE ADDED AS THE INDEX WORD.
4442                                  ;*      IF D67.10 FAILS TO ADD THE INDEX NUMBER THE SOURCE
4443                                  ;*      WILL BE PUT IN THE WRONG LOCATION.
4444                                  ;*
4445                                  ;*      ROM FLOW-6,251,122,157
4446                                  ;;*****************************************************************
4447  005352  005200                  TST47:  INC     R0              ;INCREMENT TEST NUMBER
4448  005354  005306                          DEC     SP              ;THIS
4449  005356  005306                          DEC     SP              ;GROUP
4450  005360  012705  000340                  MOV     #340,R5         ;OF INSTRUCTIONS
4451  005364  010516                          MOV     R5,(SP)         ;SETS UP THE
4452  005366  005306                          DEC     SP              ;STACK
4453  005370  005306                          DEC     SP              ;TO HANDLE
4454  005372  012705  005432                  MOV     #BAD1,R5        ;AN ERONEOUS
4455  005376  010516                          MOV     R5,(SP)         ;RTI INSTRUCTION
4456  005400  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDR OF $TMP1 IN R5
4457  005404  005015                          CLR     (R5)            ;CLEAR $TMP1
4458  005406  012701  100000                  MOV     #BIT15,R1       ;SET SIGN BIT IN R1
4459  005412  000240          SYNC47: NOP
4460  005414                  IUT47:
4461  005414  010165  000002                  MOV     R1,2(R5)        ;EXECUTE INSTRUCTION UNDER TEST
4462  005420  012706  001100                  MOV     #STACK,SP       ;RESTORE THE SP
4463  005424  005715                          TST     (R5)            ;DID INDEX WORD GET ADDED?
4464  005426  100002                          BPL     TST50           ;;BRANCH IF YES
4465  005430  000000                          HALT                    ;D67.10 FAILED TO ADD INDEX
4466                                                                  ;FOR LOOPING CHANGE TO "BR TST47+2" (751)
4467  005432                          BAD1:
4468  005432  000000                          HALT                    ;D67.00 FAILED TO INC PC
4469                                                                  ;FOR LOOPING CHANGE TO "BR TST47+2" (750)
4470                                  ;;*****************************************************************
4471                                  ;*TEST 50       FOUR  MICROSTATES (BIN*SM12*DM0*-DF7*SR0(1))
4472                                  ;*
4473                                  ;*      IF FORK A FAILS EXECTUTION WILL  GO TO STATE D12.00.
```

```
4474                                    ;*      THIS WILL HAPPEN IF RACE BF1=7 DOES NOT GO HIGH.
4475                                    ;*      STATE D12.00 WOULD BITB R5 & THE CONTENTS OF LOCATION 200
4476                                    ;*      WHICH IS 137.
4477                                    ;*
4478                                    ;*      THE FOLLOWING COULD BE BEN14*FORK C FAILURES:
4479                                    ;*      IF IRCC CO RABOO DOES NOT GO HIGH EXECUTION WILL GO TO
4480                                    ;*      DOO.90 WHICH WILL TEST THE LOW BYTE INSTEAD OF THE HIGH BYTE.
4481                                    ;*
4482                                    ;*      IF STATE DOO.80 FAILS TO SWAP THE BYTES IT WILL LOOK LIKE
4483                                    ;*      A FORK C FAILURE.
4484                                    ;*
4485                                    ;*      ROM FLOW-21,27,204,205
4486                                    ;;*****************************************************************
4487  005434  005200                    TST50:  INC     R0              ;INCREMENT TEST NUMBER
4488  005436  012705  001164                    MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
4489  005442  012701  100000                    MOV     #BIT15,R1       ;PUT NEG HIGH & POS LOW BYTE IN R1
4490  005446  010115                            MOV     R1,(R5)         ;PUT IN $TMP1
4491  005450  005205                            INC     R5              ;PUT ADDR OF $TMP1 H BYTE IN R5
4492  005452  012701  000200                    MOV     #200,R1         ;PUT POS HIGH & NEG LOW BYTE IN R1
4493  005456  000240                SYNC50: NOP
4494  005460                         IUT50:
4495  005460  131501                            BITB    (R5),R1         ;EXECUTE INSTRUCTION UNDER TEST
4496  005462  100405                            BMI     TST51           ;;TEST OK, GO TO NEXT TEST
4497  005464  010215                            MOV     R2,(R5)         ;PUT 200 IN $TMP1
4498  005466  131501                            BITB    (R5),R1         ;EXECUTE FAILED INSTRUCTION
4499  005470  100401                            BMI     1$              ;BRANCH IF FORK C FAILED
4500  005472  000000                            HALT                    ;RACE BF1=7 NOT GOING HIGH
4501                                                                    ;EITHER RACJ AFIR14(1) DOES NOT
4502                                                                    ;GET TO RACE E40 OR E40 BAD
4503                                                                    ;FOR LOOPING CHANGE TO "BR TST50+2" (751)
4504  005474                           1$:
4505  005474  000000                            HALT                    ;EITHER IRCC CO RABOO DOES NOT GO HIGH
4506                                                                    ;OR STATE DOO.80 FAILED TO SWAP THE BYTES
4507                                                                    ;FOR LOOPING CHANGE TO "BR TST50+2" (760)
4508                                    ;;*****************************************************************
4509                                    ;*TEST 51       FOUR  MICROSTATES (BIN*SM12*DMD*DF7*SRD(0))
4510                                    ;*
4511                                    ;*      IF FORK A FAILS EXECUTION WILL EITHER GO TO STATE D12.00 OR EXC.80.
4512                                    ;*      STATE D12.00 WOULD ADD R5 TO THE CONTENTS OF THE PC.
4513                                    ;*      STATE EXC.80 WOULD NOT CHANGE THE PC.
4514                                    ;*
4515                                    ;*      IF FORK C FAILS EXECUTION WILL EITHER GO TO JSR.10 OR ASC.80.
4516                                    ;*      JSR.10 WILL CAUSE R5 TO BE STACKED, THE PC TO BE PUT
4517                                    ;*      IN R5, AND THE PC REPLACED BY R5 WHICH WILL CAUSE AND RTI SINCE
4518                                    ;*      THE CONTENTS OF THE LOCATION POINTED TO BY R5 IS 000002.
4519                                    ;*      ASC.80 WILL CAUSE THE ADD TO LOOK LIKE IT FAILED.
4520                                    ;*
4521                                    ;*      IF STATE D07.10 FAILS TO LOAD THE SHFTR THE PC WILL
4522                                    ;*      DOUBLE AND THE PROGRAM WILL BLOW UP.
4523                                    ;*
4524                                    ;*      IF THE SHFTR FAILS TO BE PUT IN THE SR THE PC
4525                                    ;*      WILL NOT CHANGE.
4526                                    ;*
4527                                    ;*      ROM FLOW-21,27,203,30
4528                                    ;;*****************************************************************
4529  005476  005200                    TST51:  INC     R0              ;INCREMENT TEST NUMBER
```

```
4530  005500  012706  001100                  MOV     #STACK,SP       ;INITIALIZE SP
4531  005504  012701  000340                  MOV     #340,R1         ;PUT PRIORITY LEVEL 7 IN R5
4532  005510  010146                          MOV     R1,-(SP)        ;PUT ON STACK
4533  005512  012701  005610                  MOV     #$RTI,R1        ;PUT RETURN ADDR IN R1
4534  005516  010146                          MOV     R1,-(SP)        ;PUT ON STACK FOR FORK C FAILURE
4535  005520  005306                          DEC     SP              ;ADJUST THE
4536  005522  005306                          DEC     SP              ;SP
4537  005524  005005                          CLR     R5              ;PUT ADDRESS 0 IN R5
4538  005526  012701  000002                  MOV     #2,R1           ;PUT OFFSET IN R1
4539  005532  010115                          MOV     R1,(R5)         ;PUT OFFSET IN LOCATION 0
4540  005534  000240                  SYNC51: NOP
4541  005536                          IUT51:
4542  005536  061507                          ADD     (R5),PC         ;EXECUTE INSTRUCTION UNDER TEST
4543  005540  000401                          BR      6$              ;EITHER FORK A OR FORK C OR D07.10 FAILED
4544  005542  000423                          BR      TST52           ;;TEST OK GO TO NEXT TEST
4545  005544  012705  000004          6$:     MOV     #4,R5           ;SET BIT 2 IN R5
4546  005550  061507                          ADD     (R5),PC         ;EXECUTE FAILED INSTR.
4547  005552  000261                  3$:     SEC                     ;WILL CHANGE TO SEC!SEZ IF THE INSTR GOES
4548                                                                  ;THRU D12.00.  STATE EXC.8 OR ASC.8
4549                                                                  ;WOULD NOT SET Z WHILE A FAILURE OF
4550                                                                  ;STATE D07.10 WILL CAUSE THE ERROR
4551                                                                  ;1$-2 TO REPORT.
4552  005554  000401                          BR      1$              ;SKIP NEXT INSTRUCTION
4553  005556  000000                          HALT                    ;STATE D07.10 DID NOT LOAD SR
4554                                                                  ;FOR LOOPING CHANGE TO "BR TST51+2" (750)
4555  005560  001004                  1$:     BNE     2$              ;BRANCH IF Z DID NOT SET
4556  005562  012767  000261  177762          MOV     #261,3$         ;RESTORE ORIGINAL VALUE OF LOCATION 3$
4557  005570  000000                          HALT                    ;RACE BF1=7 DID NOT GO HIGH
4558                                                                  ;FOR LOOPING CHANGE TO "BR TST51+2" (743)
4559  005572  012705  100000          2$:     MOV     #BIT15,R5       ;SET SIGN BIT IN R5
4560  005576  000250                          CLN                     ;ENSURE N CLEAR
4561  005600  061507                          ADD     (R5),PC         ;EXECUTE FAILED INSTR
4562  005602  100401                          BMI     4$              ;BRANCH IF ADD OCCURED IN STATE EXC.80
4563  005604  000000                          HALT                    ;EITHER IRCC CO RAB02 IS BEING HELD LOW
4564                                                                  ; OR IT IS NOT GETTING THRU
4565                                                                  ;TO RACL RADR02
4566                                                                  ;FOR LOOPING CHANGE TO "BR TST51+2" (735)
4567  005606                          4$:
4568  005606  000000                          HALT                    ;RACE BIN IS NOT GOING LOW
4569                                                                  ;FOR LOOPING CHANGE TO "BR TST51+2" (734)
4570  005610                          $RTI:
4571  005610  000000                          HALT                    ;EITHER IRCC CO RAB01 IS STUCK HIGH
4572                                                                  ;OR IRC E40 IS BAD OR IRC E40(14)
4573                                                                  ;IS STUCK HIGH
4574                                                                  ;OR CO RAB01 IS NOT GETTING THRU TO RACL RADR01
4575                                                                  ;FOR LOOPING CHANGE TO "BR TST51+2" (733)
4576                                  ;;*****************************************************************
4577                                  ;*TEST 52       FOUR  MICROSTATES (BIN*SM12*DMO*DF7*SRO(1))
4578                                  ;*
4579                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO D12.00.
4580                                  ;*
4581                                  ;*      FORK C SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN TESTED.
4582                                  ;*
4583                                  ;*      IF STATE D07.00 FAILS TO SWAP THE BYTES THE CC'S WILL
4584                                  ;*      BE BAD.
4585                                  ;*      IF D07.00 FAILS TO LOAD THE SHIFTER, THE THIRD CMPB WILL FAIL.
```

```
4586                                    ;*      IF D07.00 FAILS TO LOAD THE SR THE SECOND CMPB WILL FAIL.
4587                                    ;*
4588                                    ;*      ROM FLOW-21,27,202,30
4589                                    ;;*****************************************************************
4590  005612  005200                    TST52:  INC     R0              ;INCREMENT TEST NUMBER
4591  005614  005005                            CLR     R5              ;PUT ADDRESS 0 IN R5
4592  005616  012701  005634                    MOV     #POINT,R1       ;PUT ADDR OF POINT IN R1
4593  005622  000301                            SWAB    R1              ;EXCHANGE BYTES
4594  005624  010115                            MOV     R1,(R5)         ;PUT DATA IN LOCATION 0
4595  005626  005205                            INC     R5              ;CHANGE R5 TO HIGH BYTE ADDRESS
4596  005630  000240                    SYNC52: NOP
4597  005632                            IUT52:
4598  005632  121507                            CMPB    (R5),PC         ;EXECUTE INSTRUCTION UNDER TEST
4599  005634  001406                    POINT:  BEQ     4$              ;BRANCH IF OK
4600  005636  016705  000002                    MOV     2$,R5           ;PUT CONTENTS OF 2$ IN R5
4601  005642  121507                            CMPB    (R5),PC         ;EXECUTE FAILED INSTRUCTION
4602  005644  001401                    2$:     BEQ     3$              ;BRANCH IF FORK A FAILED
4603  005646  000000                            HALT                    ;STATE D07.00 FAILED TO SWAB OR LOAD SR
4604                                            ;FOR LOOPING CHANGE TO "BR TST52+2" (762)
4605  005650                            3$:
4606  005650  000000                            HALT                    ;RACE BF1=0 NOT GOING HIGH
4607                                            ;FOR LOOPING CHANGE TO "BR TST52+2" (761)
4608  005652  121507                    4$:     CMPB    (R5),PC         ;DID SHFTR GET LOADED
4609  005654  001001                            BNE     TST53           ;;BRANCH IF YES
4610  005656  000000                            HALT                    ;D07.00 DID NOT LOAD SHFTR
4611                                            ;FOR LOOPING CHANGE TO "BR TST52+2" (756)
4612
4613                                    ;;*****************************************************************
4614                                    ;*TEST 53       FOUR  MICROSTATES (BIN*SM4*DM0*-DF7*SR0(0))
4615                                    ;*
4616                                    ;*      IF FORK A FAILS EXECUTION WILL
4617                                    ;*      GO TO D45.00 WHICH WILL EXECUTE A SM0*DM4 INSTRUCTION EXCEPT
4618                                    ;*      THE DESTINATION REGISTER WILL NOT DECREMENT.
4619                                    ;*
4620                                    ;*      FORK C WILL NOT FAIL SINCE IT HAS ALREADY BEEN TESTED.
4621                                    ;*
4622                                    ;*      IF THE SRC FAILS TO AUTO DECREMENT STATE S45.00 IS BAD.
4623                                    ;*
4624                                    ;*      ROM FLOW-24,23,27,205
4625                                    ;;*****************************************************************
4626  005660  005200                    TST53:  INC     R0              ;INCREMENT TEST NUMBER
4627  005662  012705  001164                    MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
4628  005666  012701  100000                    MOV     #BIT15,R1       ;SET SIGN BIT IN R1
4629  005672  010125                            MOV     R1,(R5)+        ;SET SIGN BIT $TMP1 & STEP R5 TO $TMP2
4630  005674  005015                            CLR     (R5)            ;CLEAR $TMP2
4631  005676  012701  000002                    MOV     #2,R1           ;PUT ADDRESS OF LOC 2 IN R1
4632  005702  005011                            CLR     (R1)            ;CLEAR LOCATION ZERO
4633  005704  000240                    SYNC53: NOP
4634  005706                            IUT53:
4635  005706  054501                            BIS     -(R5),R1        ;EXECUTE INSTRUCTION UNDER TEST
4636  005710  100411                            BMI     TST54           ;;TEST OK, GO TO NEXT TEST
4637  005712  020537  000002                    CMP     R5,@#2          ;DID FORK A FAIL?
4638  005716  001001                            BNE     1$              ;BRANCH IF NO
4639  005720  000000                            HALT                    ;EITHER RACE BF1=7 OR SM0 DID NOT GO HIGH
4640                                            ;FOR LOOPING CHANGE TO "BR TST53+2" (760)
4641  005722  020527  001166            1$:     CMP     R5,#$TMP2       ;DID R5 FAIL TO DECREMENT?
```

```
4642  005726  001001                          BNE     2$              ;BRANCH IF NO
4643  005730  000000                          HALT                    ;STATE S45.00 DID NOT DECREMENT R5
4644                                                                  ;FOR LOOPING CHANGE TO "BR TST53+2" (754)
4645  005732                          2$:
4646  005732  000000                          HALT                    ;INSTRUCTION FAILED
4647                                                                  ;FOR LOOPING CHANGE TO "BR TST53+2" (753)
4648                                  ;;**************************************************************
4649                                  ;*TEST 54       FOUR  MICROSTATES (RTS)
4650                                  ;*
4651                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
4652                                  ;*      THIS WILL ONLY HAPPEN IF RACF E3 DOES NOT GO HIGH.
4653                                  ;*
4654                                  ;*      IF THE PC OR R5 FAILS THE TEST WILL HALT.
4655                                  ;*
4656                                  ;*      ROM FLOW-40,223,224,342
4657                                  ;;**************************************************************
4658  005734  005200                  TST54:  INC     R0              ;INCREMENT TEST NUMBER
4659  005736  012706  001100                  MOV     #STACK,SP       ;INITIALIZE THE STACK
4660  005742  012705  100000                  MOV     #BIT15,R5       ;SET SIGN BIT IN R5
4661  005746  010546                          MOV     R5,-(SP)        ;PUT R5 ON THE STACK
4662  005750  012705  005760                  MOV     #1$,R5          ;PUT ADDRESS TO RETURN TO IN R5
4663  005754  000205                          RTS     R5              ;EXECUTE INSTRUCTION UNDER TEST
4664  005756  000000                          HALT                    ;STATE RTS.00 FAILED TO PUT R5 IN THE PC
4665                                                                  ;FOR LOOPING CHANGE TO "BR TST54+2" (767)
4666  005760  005705                  1$:     TST     R5              ;DID R5 GET THE TOP OF THE STACK?
4667  005762  100401                          BMI     TST55           ;;BRANCH IF YES
4668  005764  000000                          HALT                    ;THE RTS FAILED TO PUT THE STACK IN R5
4669                                                                  ;FOR LOOPING CHANGE TO "BR TST54+2" (764)
4670                                  ;;**************************************************************
4671                                  ;*TEST 55       FOUR  MICROSTATES (JMP*DM6)
4672                                  ;*
4673                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO STATE D12.01.
4674                                  ;*      THIS WOULD CAUSE A JMP*DM1 TO EXECUTE.
4675                                  ;*
4676                                  ;*      NEITHER BEN01 NOR BEN15*FEN2 SHOULD FAIL SINCE THEY HAVE ALREADY BEEN
4677                                  ;*      TESTED.
4678                                  ;*
4679                                  ;*      ROM FLOW-6,251,122,35
4680                                  ;;**************************************************************
4681  005766  005200                  TST55:  INC     R0              ;INCREMENT TEST NUMBER
4682  005770  012705  006002                  MOV     #JMPTO,R5       ;PUT ADDRESS-2 TO JUMP TO IN R5
4683  005774  000240                  SYNC55: NOP
4684  005776                          IUT55:
4685  005776  000165  000002                  JMP     2(R5)           ;EXECUTE INSTRUCTION UNDER TEST
4686  006002                          JMPTO:
4687  006002  000000                          HALT                    ;JMP*DM6 DID NOT JUMP OR THE OFFSET
4688                                                                  ;DID NOT GET ADDED OR RACE E33 FAILED
4689                                                                  ;AND A JMP*DM1 WAS EXECUTED
4690                                                                  ;FOR LOOPING CHANGE TO "BR TST55+2" (772)
4691                                          .SBTTL
4692                                  ;;**************************************************************
4693                                  ;*TEST 56       FIVE  MICROSTATES (DAC*DM12*P/CLASS*DR0(1))
4694                                  ;*
4695                                  ;*      FORK A SHOULDN'T FAIL.
4696                                  ;*
4697                                  ;*      BEN15 SHOULDN'T FAIL SINCE THIS LOGIC HAS BEEN TESTED.
```

```
                                        ;*      NEITHER SHOULD BEN05*FEN2.
                                        ;*
                                        ;*      IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10. .
                                        ;*      THIS FAILURE WOULD BE CAUSED BY A BAD FIELD (PART PCLASS)
                                        ;*      IN THE IR DECODE ROM.
                                        ;*
                                        ;*      ROM FLOW-1,175,137,31,132
                                        ;;*****************************************************************
006004  005200                  TST56:  INC     R0              ;INCREMENT TEST NUMBER
006006  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
006012  012701  040000                  MOV     #BIT14,R1       ;SET BIT 14 IN R1
006016  010115                          MOV     R1,(R5)         ;SET BIT 14 IN $TMP1
006020  005205                          INC     R5              ;SET R5 TO HIGH BYTE OF $TMP1
006022  000240                  SYNC56: NOP
006024                          IUT56:
006024  106115                          ROLB    (R5)            ;EXECUTE INSTRUCTION UNDER TEST
006026  100401                          BMI     TST57           ;;TEST OK, GO TO NEXT TEST
006030  000000                          HALT                    ;ROLB*DM1*DR5(1)) FAILED (BAD CC)
                                                                ;FOR LOOPING CHANGE TO "BR TST56+2" (766)
                                        ;;*****************************************************************
                                        ;*TEST 57       FIVE  MICROSTATES (DAC*DM3*0/CLASS)
                                        ;*
                                        ;*      FORK A SHOULD NOT FAIL SINCE THE LOGIC HAS ALREADY BEEN TESTED.
                                        ;*
                                        ;*      IF THE DR DOES NOT AUTO INC THEN STATE D30.10 IS BAD.
                                        ;*
                                        ;*      IF THE CONDITION CODES ARE BAD THEN EITHER STATE D10.50
                                        ;*      DID NOT LOAD THE BR OR THE DOUBLE DEFERED DIDN'T WORK.
                                        ;*
                                        ;*      ROM FLOW-3,221,233,311,157
                                        ;;*****************************************************************
006032  005200                  TST57:  INC     R0              ;INCREMENT TEST NUMBER
006034  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
006040  012701  001166                  MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
006044  010115                          MOV     R1,(R5)         ;PUT ADDRESS OF $TMP2 IN $TMP1
006046  005003                          CLR     R3
006050  010311                          MOV     R3,(R1)         ;CLEAR $TMP2
006052  012703  100000                  MOV     #BIT15,R3       ;SET SIGN BIT IN R3
006056  000240                  SYNC57: NOP
006060                          IUT57:
006060  010335                          MOV     R3,@(R5)+       ;EXECUTE INSTRUCTION UNDER TEST
006062  100401                          BMI     1$              ;BRANCH IF N BIT SET
006064  000000                          HALT                    ;BAD CONDITION CODES
                                                                ;FOR LOOPING CHANGE TO "BR TST57+2" (763)
006066  005711                  1$:     TST     (R1)            ;DID MOVE ACTUALLY TAKE PLACE?
006070  100401                          BMI     2$              ;BRANCH IF YES
006072  000000                          HALT                    ;SOURCE DID NOT GET MOVED TO DESTINATION
                                                                ;FOR LOOPING CHANGE TO "BR TST57+2" (760)
006074  020501                  2$:     CMP     R5,R1           ;DID R5 AUTO INC?
006076  001401                          BEQ     TST60           ;;BRANCH IF YES
006100  000000                          HALT                    ;STATE D30.10 DID NOT INC R5
                                                                ;FOR LOOPING CHANGE TO "BR TST57+2" (755)
                                        ;;*****************************************************************
                                        ;*TEST 60       FIVE  MICROSTATES (DAC*DM4*P/CLASS*DR0(0))
                                        ;*
                                        ;*      FORK A SHOULD NOT FAIL.
```

```
4754                                    ;*
4755                                    ;*      IF FORK B FAILS, AFTER D10.60, EXECUTION WILL GO TO
4756                                    ;*      RSD.00 CAUSING A TRAP TO 10. THIS WILL ONLY HAPPEN IF
4757                                    ;*      THE IR DECODE ROM HAS A BAD FIELD (PART PCLASS).
4758                                    ;*
4759                                    ;*      ROM FLOW-4,122,177,31,132
4760                                    ;;*****************************************************************
4761  006102  005200                   TST60:  INC     R0              ;INCREMENT TEST NUMBER
4762  006104  012705  001164                   MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
4763  006110  005025                           CLR     (R5)+           ;CLEAR $TMP1 & STEP R5 TO $TMP2
4764  006112  000270                           SEN                     ;SET N
4765  006114  000240                   SYNC60: NOP
4766  006116                            IUT60:
4767  006116  006745                           SXT     -(R5)           ;EXECUTE INSTRUCTION UNDER TEST.
4768  006120  005215                           INC     (R5)            ;SHOULD MAKE $TMP1=0
4769  006122  001401                           BEQ     TST61           ;;BRANCH IF TEST OK
4770  006124  000000                           HALT                    ;SXT DID NOT WORK
4771                                                                    ;FOR LOOPING CHANGE TO "BR TST60+2" (767)
4772
4773
4774                                    ;;*****************************************************************
4775                                    ;*THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A DAC*DM4*[TST.B+
4776                                    ;*BIT.B+CMP.B]*DR0(1) INSTRUCTION FOLLOWED BY A DAC*DM6*[TST.B+BIT.B+CMP.B]*
4777                                    ;*DR0(0) INSTRUCTION TEST BUT NO ADDITIONAL LOGIC IS TESTED.
4778                                    ;;*****************************************************************
4779
4780
4781
4782
4783                                    ;;*****************************************************************
4784                                    ;*THE LOGICAL SEQUENCE AT THIS POINT WOULD BE TO EXECUTE A BIN*SM4*DM0*-DF7*SR0(1)
4785                                    ;*FOLLOWED BY A BIN*SM4*DM0*DF7*SR0(0)
4786                                    ;*FOLLOWED BY A BIN*SM4*DM0*DF7*SR0(1) INSTRUCTION BUT NO ADDITIONAL LOGIC IS TESTED.
4787                                    ;;*****************************************************************
4788
4789
4790                                    ;;*****************************************************************
4791                                    ;*TEST 61        FIVE  MICROSTATES (BIN*SM6*DM0*-DF7*SR0(0))
4792                                    ;*
4793                                    ;*      IF FORK A FAILS EXECUTION WILL GO TO STATE D67.00
4794                                    ;*      WHICH WOULD EXECUTE A SM0*DM6 INSTRUCTION.
4795                                    ;*
4796                                    ;*      BEN14*FEN4 SHOULD NOT FAIL SINCE IT HAS ALREADY BEEN TESTED
4797                                    ;*
4798                                    ;*      ROM FLOW-26,54,141,142,205
4799                                    ;;*****************************************************************
4800  006126  005200                   TST61:  INC     R0              ;INCREMENT TEST NUMBER
4801  006130  012705  001164                   MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
4802  006134  010565  000002                   MOV     R5,2(R5)        ;PUT ADDRESS OF $TMP1 IN $TMP2
4803  006140  010501                           MOV     R5,R1           ;PUT ADDRESS OF $TMP1 IN R1
4804  006142  000240                   SYNC61: NOP
4805  006144                            IUT61:
4806  006144  046501  000002                   BIC     2(R5),R1        ;EXECUTE INSTRUCTION UNDER TEST
4807  006150  005701                           TST     R1              ;DID R1 GO TO ZERO?
4808  006152  001405                           BEQ     TST62           ;;BRANCH IF YES
4809  006154  005767  173006                   TST     $TMP2           ;DID $TMP2 GO TO ZERO?
```

```
 4810  006160  001001                        BNE     1$              ;BRANCH IF NO
 4811  006162  000000                        HALT                    ;RACE BF1=0 DID NOT GO HIGH ON BIC
 4812                                                                ;FOR LOOPING CHANGE TO "BR TST61+2" (762)
 4813  006164                          1$:
 4814  006164  000000                        HALT                    ;INSTRUCTION FAILED
 4815                                                                ;FOR LOOPING CHANGE TO "BR TST61+2" (761)
 4816                                  ;;*****************************************************************
 4817                                  ;*TEST 62       FIVE  MICROSTATES (BIN*SM12*DM12*0/CLASS)
 4818                                  ;*
 4819                                  ;*      IF FORK A FAILS EXECUTION WILL GO TO D12.01.THIS WOULD CAUSE R5
 4820                                  ;*      TO BE WRITTEN INTO $TMP2.
 4821                                  ;*
 4822                                  ;*      IF FORK C FAILS EXECUTION WOULD GO TO ONE OF THE
 4823                                  ;*      FOLLOWING STATES: D45.80,D30.80,FOP.00,WAT.00, D00.90, AND ASH.20.
 4824                                  ;*      STATE D45.80 WOULD EXECUTE A SM2*DM4 INSTEAD OF SM2*DM1.
 4825                                  ;*      STATE D30.80 WOULD EXECUTE A SM1*DM3.
 4826                                  ;*      STATE FOP.00 WOULD CAUSE A TRAP TO LOCATION 10.
 4827                                  ;*      THIS WILL ONLY HAPPEN IF EITHER IRCC CO RAB03 IS STUCK
 4828                                  ;*      OR IT IS NOT GETTING THRU RACL RADR03 OR
 4829                                  ;*      IRCC FORK C MUX INPUT B0 IS HIGH.
 4830                                  ;*      THE LA30 PRINTER BUFFER WILL BE SET UP TO GENERATE AN INTERRUPT IN
 4831                                  ;*      CASE THE TEST FAILS TO THE WAT.00 STATE.
 4832                                  ;*      STATE D00.90 WILL EXECUTE A DM0 INSTEAD OF A DM1.
 4833                                  ;*      STATE ASH.20 WILL CLEAR THE C BIT.
 4834                                  ;*
 4835                                  ;*      ROM FLOW-22,27,111,155,312
 4836                                  ;;*****************************************************************
 4837  006166  005200                TST62:  INC     R0              ;INCREMENT TEST NUMBER
 4838  006170  016767  172746  000130        MOV     $TPS,3$         ;SETUP ADDRESSES OF TP
 4839  006176  016767  172740  000134        MOV     $TPS,5$         ;STATUS AND TP BUFFER
 4840  006204  016767  172732  000140        MOV     $TPS,POINT1+2   ;INCASE THAY ARE NOT
 4841  006212  016767  172724  000202        MOV     $TPS,$$TPS
 4842  006220  016767  172720  000074        MOV     $TPB,2$         ;STANDARD ADDRESSES
 4843  006226  016767  172712  000100        MOV     $TPB,4$
 4844  006234  012706  001100                MOV     #STACK,SP       ;INITIALIZE THE SP
 4845  006240  012705  001164                MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
 4846  006244  012701  001166                MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
 4847  006250  012702  100000                MOV     #BIT15,R2       ;SET SIGN BIT IN R2
 4848  006254  010215                        MOV     R2,(R5)         ;SET SIGN BIT IN $TMP1
 4849  006256  005011                        CLR     (R1)            ;CLEAR $TMP2
 4850  006260  005002                        CLR     R2              ;PUT ADDRESS OF LOCATION 0 IN R2
 4851  006262  005012                        CLR     (R2)            ;CLEAR LOCATION ZERO
 4852  006264  012702  000064                MOV     #64,R2          ;PUT ADDRESS OF PRINTER VECTOR IN R2
 4853  006270  012704  006350                MOV     #POINT1,R4      ;PUT ADDRESS OF POINT1 IN R4
 4854  006274  010412                        MOV     R4,(R2)         ;PUT ADDRESS OF POINT1 IN PRINTER VECTOR
 4855  006276  012702  177776                MOV     #PSW,R2         ;PUT ADDRESS OF PSW IN R2
 4856  006302  005767  172572                TST     $PASS           ;IS THIS PASS 1?
 4857  006306  001015                        BNE     SYNC62          ;BRANCH IF NO
 4858  006310  012703  000101                MOV     #101,R3         ;PUT ASCII FOR "A" IN R3
 4859  006314  012704  000100                MOV     #BIT06,R4       ;PUT INTERRUPT ENABLE BIT IN R4
 4860  006320  010337                        MOV     R3,@(PC)+       ;SEND A TO PRINTER
 4861  006322  177566                2$:     .WORD   177566          ;ADDRESS OF TP BUFFER
 4862  006324  105737                1$:     TSTB    @(PC)+          ;WAIT FOR PRINTER DONE
 4863                                                                ;INCASE DOUBLE BUFFERED
 4864  006326  177564                3$:     .WORD   177564          ;ADDRESS OF TP STATUS
 4865  006330  100375                        BPL     1$
```

```
4866  006332  010337                        MOV     R3,@(PC)+       ;SEND SECOND A
4867  006334  177566              4$:       .WORD   177566
4868  006336  010437                        MOV     R4,@(PC)+       ;SET THE INTERRUPT FLSG IN TPS
4869  006340  177564              5$:       .WORD   177564
4870  006342                      SYNC62:
4871  006342  000261                        SEC                     ;SET C TO CATCH FAILURE TO ASH.20
4872  006344                      IUT62:
4873  006344  012511                        MOV     (R5)+,(R1)      ;EXECUTE INSTRUCTION UNDER TEST
4874  006346  011202                        MOV     (R2),R2         ;SAVE PSW IN R2
4875  006350  040437              POINT1:   BIC     R4,@(PC)+       ;CLEAR THE INTERR FLAG
4876  006352  177564                        .WORD   177564
4877  006354  005701                        TST     R1              ;DID A DMO GET EXECUTED?
4878  006356  100001                        BPL     3$              ;BRANCH IF NO
4879  006360  000000                        HALT                    ;IRCC DMO L STUCK LOW
4880                                                                ;FOR LOOPING CHANGE TO "BR TST62+2" (703)
4881  006362  005711              3$:       TST     (R1)            ;DID A SM2*DM4 GET EXECUTED?
4882  006364  100401                        BMI     5$              ;BRANCH IF NO
4883  006366  000000                        HALT                    ;IRCC C FORK MUX INPUT B1 IS STUCK LOW
4884                                                                ;FOR LOOPING CHANGE TO "BR TST62+2" (700)
4885  006370  011104              5$:       MOV     (R1),R4         ;GET CONTENTS OF $TMP2
4886  006372  020504                        CMP     R5,R4           ;DID D12.01 GET EXECUTED?
4887  006374  001001                        BNE     6$              ;BRANCH IF NO
4888  006376  000000                        HALT                    ;RACE E29 IS BAD
4889                                                                ;FOR LOOPING CHANGE TO "BR TST62+2" (674)
4890  006400  022706  001074      6$:       CMP     #1074,SP        ;DID INSTRUCTION CAUSE WAIT TO OCCUR?
4891  006404  001001                        BNE     2$              ;BRANCH IF NO
4892  006406  000000                        HALT                    ;EITHER IRCC DSTMO H IS STUCK LOW OR
4893                                                                ;IT IS NOT GETTING THRU RACL RADR05
4894                                                                ;FOR LOOPING CHANGE TO "BR TST62+2" (670)
4895  006410  005004              2$:       CLR     R4              ;PUT ADDR. OF LOCATION ZERO IN R4
4896  006412  005714                        TST     (R4)            ;DID A SM1*DM3 GET EXECUTED?
4897  006414  100001                        BPL     IT              ;BRANCH IF NO
4898  006416  000000                        HALT                    ;IRCC C FORK MUX INPUT B2 IS STUCK LOW
4899                                                                ;FOR LOOPING CHANGE TO "BR TST62+2" (664)
4900  006420  105737              IT:       TSTB    @(PC)+          ;IS PRINTER DONE?
4901  006422  177564              $$TPS:    .WORD   177564
4902  006424  100375                        BPL     IT              ;BRANCH IF NO
4903  006426  032702  000001              BIT     #BIT0,R2        ;DID INSTRUCTION LEAVE C BIT SET?
4904  006432  001001                        BNE     TST63           ;;BRANCH IF YES
4905  006434  000000                        HALT                    ;IRCC C FORK MUX SELECT NOT GOING HIGH(ON CHIP)
4906                                                                ;FOR LOOPING CHANGE TO "BR TST62+2" (655)
4907
4908
4909                                ;;*****************************************************************
4910                                ;*THE LOGICAL FLOW AT THIS POINT WOULD TEST A BIN*SM12*DM12*SR0(0)*DR0(0)
4911                                ;**[TST.B+BIT.B+CMP.B] INSTRUCTION BUT NO ADDITIONAL LOGIC WOULD BE TESTED.
4912                                ;;*****************************************************************
4913
4914
4915                                ;;*****************************************************************
4916                                ;*TEST 63       FIVE  MICROSTATES (BIN*SM12*DM12*SR0(1)*DR0(0)*CMPB)
4917                                ;*
4918                                ;*      THE ONLY THING THAT SHOULD FAIL WOULD BE STATE D12.90(110).
4919                                ;*
4920                                ;*      ROM FLOW-21,27,110,175,33
4921                                ;;*****************************************************************
```

```
 4922  006436  005200                TST63:  INC     R0              ;INCREMENT TEST NUMBER
 4923  006440  012705  001164                MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
 4924  006444  112715  000377                MOVB    #377,(R5)       ;SET LOW BYTE OF $TMP1 TO ALL ONE'S
 4925  006450  012701  001166                MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
 4926  006454  012711  177400                MOV     #177400,(R1)    ;SET HIGH BYTE OF $TMP2 TO ALL ONES
 4927  006460  005201                        INC     R1              ;ADJUST R1 TO $TEMP2 HIGH BYTE
 4928  006462  000240                SYNC63: NOP
 4929  006464                         IUT63:
 4930  006464  121115                        CMPB    (R1),(R5)       ;EXECUTE INSTRUCTION UNDER TEST
 4931  006466  001401                        BEQ     TST64           ;;TEST OK, GO TO NEXT TEST
 4932  006470  000000                        HALT                    ;STATE D12.90 FAILED
 4933                                                                ;FOR LOOPING CHANGE TO "BR TST63+2" (763)
 4934                                 ;;*****************************************************************
 4935                                 ;*TEST 64       FIVE  MICROSTATES (BIN*SM12*DM4*0/CLASS)
 4936                                 ;*
 4937                                 ;;      IF FORK C FAILS EXECUTION WILL GO TO D12.80
 4938                                 ;*      WHICH WILL EXECUTE A SM1*DM2 TYPE INSTRUCTION.
 4939                                 ;*
 4940                                 ;*      IF THE INSTRUCTION FAILS EITHER D45.80 OR D40.20 FAILED.
 4941                                 ;*
 4942                                 ;*      ROM FLOW-21,27,115,121,157
 4943                                 ;;*****************************************************************
 4944  006472  005200                TST64:  INC     R0              ;INCREMENT TEST NUMBER
 4945  006474  012705  001164                MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
 4946  006500  010501                        MOV     R5,R1           ;SAVE ADDRESS OF $TMP1
 4947  006502  005025                        CLR     (R5)+           ;CLEAR $TMP1 AND STEP R5 TO $TMP2
 4948  006504  012715  100000                MOV     #BIT15,(R5)     ;SET SIGN BIT IN $TMP2
 4949  006510  000240                SYNC64: NOP
 4950  006512                         IUT64:
 4951  006512  011545                        MOV     (R5),-(R5)      ;EXECUTE INSTRUCTION UNDER TEST
 4952  006514  022705  001170                CMP     #$TMP2+2,R5     ;DID R5 AUTO INCREMENT?
 4953  006520  001001                        BNE     1$              ;BRANCH IF YES
 4954  006522  000000                        HALT                    ;IRCC FORK C MUX INPUT B1 STUCK HIGH
 4955                                                                ;FOR LOOPING CHANGE TO "BR TST64+2" (764)
 4956  006524  005711                1$:     TST     (R1)            ;DID INSTRUCTION WORK?
 4957  006526  100401                        BMI     TST65           ;;BRANCH IF YES
 4958  006530  000000                        HALT                    ;EITHER STATE D45.80 OR D40.20 FAILED
 4959                                                                ;FOR LOOPING CHANGE TO "BR TST64+2" (761)
 4960                                         .SBTTL
 4961                                 ;;*****************************************************************
 4962                                 ;*TEST 65       SIX   MICROSTATES (DAC*DM12*ASRB*DRO(1))
 4963                                 ;*
 4964                                 ;*      NEITHER FORK A NOR BEN15 NOR BEN05*FEN2 SHOULD FAIL
 4965                                 ;*      SINCE THEY HAVE ALREADY BEEN TESTED.
 4966                                 ;*
 4967                                 ;*      IF FORK B FAILS AFTER D12.30 EXECUTION WILL GO TO
 4968                                 ;*      ONE OF THE FOLLOWING: RSD.00,D45.00,EXC.00,S45.00,
 4969                                 ;*      CCP.00,MUL.00,SVC.10,MFP.00 OR DEP.00.
 4970                                 ;*      RSD.00 WILL CAUSE A TRAP TO LOCATION 10.
 4971                                 ;*      THIS WILL HAPPEN IF THE B FORK MUX SELECT IS STUCK LOW.
 4972                                 ;*      IF STATE D45.00 IS ENTERED THE PROCESSOR WILL HANG UP
 4973                                 ;*      IN A LOOP BETWEEN STATES D45.00 AND D10.30.
 4974                                 ;*      IF STATE S45.00 IS ENTERED EXECUTION WILL GO TO STATE
 4975                                 ;*      D12.80 AFTER S13.10 WHICH WILL HANG UP THE PROCESSOR.
 4976                                 ;*      STATE CCP.00 WOULD SET OR CLEAR THE CONDITION CODES
 4977                                 ;*      ACCORDING TO IR(4:0).
```

```
                                        ;*      STATE MUL.00 WOULD CAUSE THE DESTINATION OPERAND TO
                                        ;*      BE MULTIPLIED BY REGISTER 2 AND THE RESULT WOULD BE
                                        ;*      STORED IN REGISTER 2 AND 3.
                                        ;*      IF SVC.10 IS ENTERED A TRAP TO 4 WILL OCCUR BECAUSE
                                        ;*      THE DESTINATION REGISTER IS ODD. THIS WILL ONLY HAPPEN
                                        ;*      IF EITHER B FORK MUX INPUT B3 OR IRCB B0 RABO0 IS STUCK LOW.
                                        ;*      IF MFP.00 IS ENTERED AN MFPI INSTRUCTION WILL BE EXECUTED
                                        ;*      THIS WILL PUSH THE ADDRESS OF 1$ ONTO THE STACK.
                                        ;*      DEP.00 WILL CAUSE THE PROCESSOR TO HANG IN MICRO ADDRESS
                                        ;*      170 WITH THE RUN LIGHT ON.
                                        ;*
                                        ;*      ROM FLOW-1,175,137,64,123,132
                                        ;*****************************************************************
006532  005200                  TST65:  INC     R0              ;INCREMENT TEST NUMBER
006534  012706  001074                  MOV     #1074,SP        ;INITIALIZE THE STACK
006540  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
006544  012701  006572                  MOV     #AFTER,R1       ;PUT ADDRESS OF AFTER IN R1
006550  010115                        MOV     R1,(R5)         ;STORE IN $TMP1
006552  005205                        INC     R5              ;SET R5 TO HIGH BYTE OF $TMP1
006554  005002                        CLR     R2              ;ENSURE R2 CLEAR
006556  012703  000001                MOV     #1,R3           ;ENSURE R3 NOT CLEAR
006562  012701  177776                MOV     #PSW,R1         ;PUT ADDRESS OF PSW IN R1
006566  000264                        SEZ                     ;ENSURE Z BIT SET
006570                          SYNC65:
006570                          IUT65:
006570  106215                        ASRB    (R5)            ;EXECUTE INSTRUCTION UNDER TEST
006572  011102                  AFTER:  MOV     (R1),R2         ;SAVE PSW
006574  010567  172354                MOV     R5,$REG0        ;SAVE R5
006600  005703                        TST     R3              ;DID MULTIPLY OCCUR & CLEAR R3?
006602  001001                        BNE     3$              ;BRANCH IF NO
006604  000000                        HALT                    ;B FORK MUX INPUT B1 STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST65+2" (753)
006606  012701  006572          3$:     MOV     #AFTER,R1       ;PUT ADDRESS OF 1$ IN R1
006612  000301                        SWAB    R1              ;REVERSE BYTES
006614  106001                        RORB    R1              ;MAKE IT LOOK LIKE EXC.00 WAS ENTERED
006616  012703  001164                MOV     #$TMP1,R3       ;PUT ADDRESS OF $TMP1 IN R3
006622  020113                        CMP     R1,(R3)         ;DID EXC.00 GET ENTERED?
006624  001001                        BNE     4$              ;BRANCH IF NO
006626  000000                        HALT                    ;IRCB 0BD(ASRB OR RORB) STUCK HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST65+2" (742)
006630  022716  006572          4$:     CMP     #AFTER,(SP)     ;DID MFP.00 EXECUTE?
006634  001001                        BNE     5$              ;BRANCH IF NO
006636  000000                        HALT                    ;B FORK MUX INPUT B2 STUCK LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST65+2" (736)
006640  032702  000004          5$:     BIT     #4,R2           ;DID CCP.00 EXECUTE?
006644  001401                        BEQ     6$              ;BRANCH IF NO
006646  000000                        HALT                    ;IRCC B0 RAB04 NOT GETTING THRU RACL RADR54
                                                                ;FOR LOOPING CHANGE TO "BR TST65+2" (732)
006650  012701  006572          6$:     MOV     #AFTER,R1       ;GET ADDRESS OF AFTER
006654  010105                        MOV     R1,R5           ;SAVE R1
006656  006001                        ROR     R1              ;RIGHT SHIFT R1 WITHOUT USING ASR
006660  042701  000377                BIC     #377,R1         ;CLEAR LOWER BYTE OF R1
006664  042705  177400                BIC     #177400,R5      ;CLEAR UPPER BYTE OF R5
006670  050501                        BIS     R5,R1           ;MAKE LOWER BYTE OF R10W
                                                                ;BYTE OF DST. OPERAND
006672  020167  172266                CMP     R1,$TMP1        ;DID DESTINATION GET ASR'D PROPERLY?
```

```
5034  006676  001401                        BEQ     TST66           ;;BRANCH IF YES
5035  006700  000000                        HALT                    ;EITHER SHR.00 OR SHR.10 FAILED
5036                                                                ;FOR LOOPING CHANGE TO "BR TST65+2" (715)
5037                                ;;**************************************************************
5038                                ;*TEST 66       SIX   MICROSTATES (DAC*DM12*RORB*DRO(1))
5039                                ;*
5040                                ;*      THIS TEST IS THE SAME AS THE LAST ONE EXCEPT A RORB
5041                                ;*      IS USED INSTEAD OF AN ASRB.
5042                                ;*
5043                                ;*      FORK B WILL ONLY FAIL IF IRCB E36(13) IS STUCK HIGH
5044                                ;*      WHICH WILL CAUSE EXECUTION TO GO TO EXC.00.
5045                                ;*
5046                                ;*      ROM FLOW-2,175,137,64,123,132
5047                                ;;**************************************************************
5048  006702  005200                TST66:  INC     R0              ;INCREMENT TEST NUMBER
5049  006704  012705  001164                MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 HIGH BYTE IN R5
5050  006710  112725  000100                MOVB    #100,(R5)+      ;SET BIT 6 IN LOW BYTE OF $TMP1
5051  006714  112715  000200                MOVB    #200,(R5)       ;SET SIGN BIT IN HIGH BYTE OF $TMP1
5052  006720  000240                SYNC66: NOP
5053  006722                         IUT66:
5054  006722  106025                        RORB    (R5)+           ;EXECUTE INSTRUCTION UNDER TEST
5055  006724  022745  040100                CMP     #40100,-(R5)    ;DID INSTRUCTION WORK?
5056  006730  001401                        BEQ     TST67           ;;BRANCH IF YES
5057  006732  000000                        HALT                    ;IRCB E36(13) NOT GOING LOW ON RORB
5058                                                                ;FOR LOOPING CHANGE TO "BR TST66+2" (764)
5059
5060
5061
5062                                ;;**************************************************************
5063                                ;*THE LOGICAL FLOW AT THIS POINT WOULD TEST A DAC*DM3*[TST.B+BIT.B+CMP.B]*DRO(0)
5064                                ;*FOLLOWED BY A DAC*DM4*P/CLASS*DRO(0) INSTRUCTION BUT NO ADDITIONAL LOGIC
5065                                ;*IS TESTED
5066                                ;;**************************************************************
5067
5068
5069
5070                                ;;**************************************************************
5071                                ;*TEST 67       SIX   MICROSTATES (DAC*DM6*XOR*DRO(0))
5072                                ;*
5073                                ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO 10.
5074                                ;*      THIS SHOULD ONLY HAPPEN IF RACE E35(1) IS BAD.
5075                                ;*
5076                                ;*      IF THE INSTRUCTION DOESN'T WORK IT WILL HALT IN THIS TEST.
5077                                ;*
5078                                ;*      ROM FLOW-6,251,122,177,31,132
5079                                ;;**************************************************************
5080  006734  005200                TST67:  INC     R0              ;INCREMENT TEST NUMBER
5081  006736  005004                        CLR     R4              ;SETUP R4
5082  006740  005104                        COM     R4              ;SET ALL BITS IN R4
5083  006742  010437  001164                MOV     R4,@#$TMP1      ;SET ALL BITS IN $TMP1
5084  006746  000240                SYNC67: NOP
5085  006750                         IUT67:
5086  006750  074467  172210                XOR     R4,$TMP1        ;EXECUTE INSTRUCTION UNDER TEST
5087  006754  005767  172204                TST     $TMP1           ;DID $TMP1 CLEAR?
5088  006760  001401                        BEQ     TST70           ;;BRANCH IF YES
5089  006762  000000                        HALT                    ;INSTRUCTION FAILED
```

```
5090                                                    ;FOR LOOPING CHANGE TO "BR TST67+2" (765)
5091
5092
5093
5094
5095                            ;;*********************************************************************
5096                            ;*THE LOGICAL SEQUENCE WOULD TEST A DAC*DM6*[TST.B+BIT.B+CMP.B]*DRO(1) BUT ALL
5097                            ;*THE LOGIC HAS BEEN TESTED.
5098                            ;;*********************************************************************
5099
5100
5101
5102                            ;;*********************************************************************
5103                            ;*TEST 70       SIX   MICROSTATES (NEG.B*DM12*DRO(0))
5104                            ;*
5105                            ;*      NEITHER FORK A NOR BGN15 SHOULD FAIL.
5106                            ;*
5107                            ;*      IF FORK B FAILS EXECUTION WILL GO TO RSD.00 CAUSING
5108                            ;*      A TRAP TO LOCATION 10 OR EXC.00.
5109                            ;*      RSD.00 SHOULD ONLY OCCUR IF THE B FORK MUX
5110                            ;*      STROBE IS BEING HELD LOW(CHIP FAILURE).
5111                            ;*
5112                            ;*      ROM FLOW-1,175,67,271,163,132
5113                            ;;*********************************************************************
5114  006764  005200            TST70:  INC     R0              ;INCREMENT TEST NUMBER
5115  006766  005067  172172            CLR     $TMP1           ;ENSURE $TMP1 CLEAR
5116  006772  005267  172166            INC     $TMP1           ;PUT 1 IN $TMP1
5117  006776  012705  001164            MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
5118  007002  000240            SYNC70: NOP
5119  007004                    IUT70:
5120  007004  005415                    NEG     (R5)            ;EXECUTE INSTRUCTION UNDER TEST
5121  007006  022715  177777            CMP     #177777,(R5)    ;DID $TMP1 NEGATE?
5122  007012  001411                    BEQ     3$              ;BRANCH IF YES
5123  007014  022715  177776            CMP     #177776,(R5)    ;DID $TMP1 COMPLEMENT?
5124  007020  001001                    BNE     1$              ;BRANCH IF NO
5125  007022  000000                    HALT                    ;EITHER STATE NEG.10 FAILED
5126                                                            ;OR FORK B FAILED. IRCB NEG.B H
5127                                                            ;IS STUCK HIGH.
5128                                                            ;FOR LOOPING CHANGE TO "BR TST70+2" (761)
5129  007024  022715  000001    1$:     CMP     #1,(R5)         ;DID $TMP1 STAY THE SAME?
5130  007030  001001                    BNE     2$              ;BRANCH IF NO
5131  007032  000000                    HALT                    ;$TMP1 DID NOT GET LOADED
5132                                                            ;FOR LOOPING CHANGE TO "BR TST70+2" (755)
5133  007034                    2$:
5134  007034  000000                    HALT                    ;INSTRUCTION FAILED
5135                                                            ;FOR LOOPING CHANGE TO "BR TST70+2" (754)
5136  007036  000264            3$:     SEZ                     ;ENSURE Z SET
5137  007040  005415                    NEG     (R5)            ;EXECUTE INSTRUCTION UNDER TEST
5138  007042  001001                    BNE     TST71           ;;CC'S OK
5139  007044  000000                    HALT                    ;STATE NEG.10 BAD
5140                                                            ;FOR LOOPING CHANGE TO "BR TST70+2" (750)
5141
5142
5143
5144                            ;;*********************************************************************
5145                            ;*THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A JMP*DM3 BUT NO
```

```
                                        ;*ADDITIONAL LOGIC IS TESTED.
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 71       SIX   MICROSTATES (BIN*SM3*DM0*-DF7*SR0(0))
                                        ;*
                                        ;*      FORK A SHOULD NOT FAIL.
                                        ;*
                                        ;*      IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90. THIS
                                        ;*      WILL CAUSE A SM2 TO BE EXECUTED. THIS SHOULD ONLY
                                        ;*      HAPPEN IF IRCC SM357 IS STUCK LOW OR NOT GETTING THRU RACL E70.
                                        ;*
                                        ;*      ROM FLOW-22,27,317,143,146,205
                                        ;;*****************************************************************
005200                          TST71:  INC     R0              ;INCREMENT TEST NUMBER
012705  001164                          MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
012715  007064                          MOV     #POINT2,(R5)    ;PUT ADDRESS OF POINT2 IN $TMP1
000240                          SYNC71: NOP
                                IUT71:
013501                                  MOV     @(R5)+,R1       ;EXECUTE INSTRUCTION UNDER TEST
026701  177774                  POINT2: CMP     POINT2,R1       ;DID R1 GET CORRECT DATA?
001405                                  BEQ     TST72           ;;BRANCH IF YES
022701  007064                          CMP     #POINT2,R1      ;DID A SM2 GET EXECUTED?
001001                                  BNE     2$              ;BRANCH IF NO
000000                                  HALT                    ;EITHER IRCC SM357 STUCK LOW
                                                                ;OR NOT GETTING THRU RACL E70
                                                                ;FOR LOOPING CHANGE TO "BR TST71+2" (763)
007102                          2$:
007102  000000                          HALT                    ;EITHER S13.20 OR S13.30 OR S13.40 FAILED
                                                                ;FOR LOOPING CHANGE TO "BR TST71+2" (762)


                                        ;;*****************************************************************
                                        ;*THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM6*DM0*DF7*SR0(1), THEN A BIN+SM12*
                                        ;*DM12*SR0(0)*DR0(1)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM12*DM12*SR0(0)*
                                        ;*DR0(0)*P/CLASS THEN A BIN*SM12*DM12*SR0(1)*DR0(1)*[TST.B+BIT.B+CMP.B]
                                        ;*THEN A BIN*SM12*DM12*SR0(1)*DR0(0)*P/CLASS AND THEN A BIN*SM12*DM4*
                                        ;*SR0(0)*DR0(0)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT NO ADDITIONAL LOGIC
                                        ;*IS TESTED.
                                        ;;*****************************************************************


                                        ;;*****************************************************************
                                        ;*TEST 72       SIX   MICROSTATES (BIN*SM12*DM4*SR0(1)*DR0(0)*CMPB)
                                        ;*
                                        ;*      A FAILURE WILL ONLY OCCUR IF STATE D45.90 FAILS.
                                        ;*
                                        ;*      IF THE DST REG. DOES NOT DECREMENT STATE D45.90 IS BAD.
                                        ;*      IF THE CONDITION CODES ARE BAD THEN STATE D40.30
                                        ;*      PROBABLY DID NOT SWAP THE BYTES OF THE SRC OPERAND.
                                        ;*
                                        ;*      ROM FLOW-1,27,114,131,177,33
                                        ;;*****************************************************************
005200                          TST72:  INC     R0              ;INCREMENT TEST NUMBER
012705  100000                          MOV     #BIT15,R5       ;SET SIGN BIT IN R5
```

```
5202  007112  010567  172046                  MOV     R5,$TMP1        ;SET SIGN BIT IN $TMP1
5203  007116  012701  001166                  MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
5204  007122  010105                          MOV     R1,R5           ;PUT ADDRESS OF $TMP2 IN R5
5205  007124  112721  000200                  MOVB    #BIT7,(R1)+     ;SET LOW BYTE SIGN IN $TMP2 & STEP R1
5206  007130  105011                          CLRB    (R1)            ;ENSURE $TMP2 HIGH BYTE CLEAR
5207  007132  005305                          DEC     R5              ;ADJUST R5 TO POINT AT $TMF1 HIGH BYTE
5208  007134  000240                  SYNC72: NOP
5209  007136                          IUT72:
5210  007136  121541                          CMPB    (R5),-(R1)      ;EXECUTE INSTRUCTION UNDER TEST
5211  007140  001405                          BEQ     TST73           ;;TEST OK, GO TO NEXT TEST
5212  007142  020127  001166                  CMP     R1,#$TMP2       ;DID R1 DECREMENT?
5213  007146  001001                          BNE     1$              ;BRANCH IF YES
5214  007150  000000                          HALT                    ;STATE D45.90 DID NOT DECREMENT
5215                                                                  ;FOR LOOPING CHANGE TO "BR TST72+2" (756)
5216  007152                          1$:
5217  007152  000000                          HALT                    ;INSTRUCTION FAILED
5218                                                                  ;FOR LOOPING CHANGE TO "BR TST72+2" (755)
5219
5220
5221                                  ;;**************************************************************
5222                                  ;*THE LOGICAL FLOW WOULD NEXT TEST A BIN*SM4*DM12*SRO(0)*DRO(0)*0/CLASS
5223                                  ;*THEN A BIN*SM4*DM12*SRO(0)*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM4*
5224                                  ;*DM12*SRO(1)*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A BIN*SM4*DM4*SRO(0)*DRO(0)*
5225                                  ;*0/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.
5226                                  ;;**************************************************************
5227
5228
5229                                  ;;**************************************************************
5230                                  ;*TEST 73       SIX   MICROSTATES (DAC*DM5*DRO(0)*0/CLASS)
5231                                  ;*
5232                                  ;*      FORK A SHOULD NOT FAIL.
5233                                  ;*
5234                                  ;*      IF IRCD DM357 IS STUCK LOW OR NOT GETTING THRU
5235                                  ;*      TO RACK E51 A DM4 WILL BE EXECUTED.
5236                                  ;*      IF STATE D10.00 OR D10.10 FAIL TO FETCH THE DEFERED ADDRESS
5237                                  ;*      THE SOURCE WILL BE STORED IN THE DESTINATION.
5238                                  ;*
5239                                  ;*      ROM FLOW-5,162,231,233,311,157
5240                                  ;**************************************************************
5241  007154  005200                  TST73:  INC     R0              ;INCREMENT TEST NUMBER
5242  007156  012705  100000                  MOV     #BIT15,R5       ;SET SIGN BIT IN R5
5243  007162  012701  001166                  MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
5244  007166  010167  171772                  MOV     R1,$TMP1        ;PUT ADDRESS OF $TMP2 IN $TMP1
5245  007172  005011                          CLR     (R1)            ;ENSURE $TMP2 CLEAR
5246  007174  000240                  SYNC73: NOP
5247  007176                          IUT73:
5248  007176  010551                          MOV     R5,@-(R1)       ;EXECUTE INSTRUCTION UNDER TEST
5249  007200  005767  171762                  TST     $TMP2           ;DID SIGN BIT GET SET IN $TMP2?
5250  007204  100405                          BMI     TST74           ;;BRANCJ IF YES
5251  007206  020567  171752                  CMP     R5,$TMP1        ;DID MODE 4 GET EXECUTED?
5252  007212  001001                          BNE     1$
5253  007214  000000                          HALT                    ;EITHER IRCD DM357 IS NOT GOING LOW
5254                                                                  ;OR NOT GETTING THRU TO RACL E51
5255                                                                  ;FOR LOOPING CHANGE TO "BR TST73+2" (760)
5256  007216                          1$:
5257  007216  000000                          HALT                    ;INSTRUCTION FAILED
```

```
                                                        ;FOR LOOPING CHANGE TO "BR TST73+2" (757)


                                ;;*****************************************************************
                                ;*THE LOGICAL SEQUENCE WOULD NEXT TEST A DAC*DM3*DRO(0)*P/CLASS THEN A
                                ;*DAC*DM3*DRO(1)*[TST.B+BIT.B+CMP.B] THEN A DAC*DM4*DRO(1)*[ASRB+
                                ;*RORB] THEN A DAC*DM5*DRO(0)*[TST.B+BIT.B+CMP.B] THEN A DAC+DM6*
                                ;*DRO(1)*P/CLASS INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.
                                ;;*****************************************************************


                                        .SBTTL
                                ;;*****************************************************************
                                ;*TEST 74        SEVEN MICROSTATES (DAC*DM7*0/CLASS)
                                ;*
                                ;*      FORK A SHOULD NOT FAIL.
                                ;*
                                ;*      IF IRCD DM357 DOES NOT GO HIGH A DM6 WILL BE EXECUTED.
                                ;*
                                ;*      ALL OTHER LOGIC HAS BEEN TESTED.
                                ;*
                                ;*      ROM FLOW-7,251,162,231,233,311,157
                                ;;*****************************************************************
007220  005200                  TST74:  INC     R0              ;INCREMENT TEST NUMBER
007222  012705  001166                  MOV     #$TMP2,R5       ;PUT ADDRESS OF $TMP2 IN R5
007226  010567  171732                  MOV     R5,$TMP1        ;PUT ADDRESS OF $TMP2 IN $TMP1
007232  012701  001162                  MOV     #$TMP0,R1       ;PUT ADDRESS OF $TMP0 IN R1
007236  012702  100000                  MOV     #BIT15,R2       ;SET SIGN BIT IN R2
007242  005067  171720                  CLR     $TMP2           ;ENSURE $TMP2 CLEAR
007246  000240                  SYNC74: NOP
007250                          IUT74:
007250  010271  000002                  MOV     R2,@2(R1)       ;EXECUTE INSTRUCTION UNDER TEST
007254  005767  171706                  TST     $TMP2           ;DID $TMP2 GET SIGN BIT SET?
007260  100405                          BMI     TST75           ;;BRANCH IF YES
007262  020267  171676                  CMP     R2,$TMP1        ;DID DM6 GET EXECUTED?
007266  001001                          BNE     1$              ;BRANCH IF NO
007270  000000                          HALT                    ;IRCD DM357 DID NOT GO HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST74+2" (754)
007272                          1$:
007272  000000                          HALT                    ;INSTRUCTION FAILED
                                                                ;FOR LOOPING CHANGE TO "BR TST74+2" (753)


                                ;;*****************************************************************
                                ;*THE LOGICAL SEQUENCE WOULD NEXT TEST A NEG.B*DM12*DRO(1) THEN A NEG.B*DM4*DRO(0)
                                ;*THEN A JMP*DM5 INSTRUCTION, BUT NO ADDITIONAL LOGIC IS TESTED.
                                ;;*****************************************************************


                                ;;*****************************************************************
                                ;*TEST 75        SEVEN MICROSTATES (JSR*DM12)
                                ;*
                                ;*      IF FORK A FAILS EXECUTION WILL GO TO RSD.00 CAUSING A TRAP TO LOCATION 10.
                                ;*      THIS WILL ONLY OCCUR IF RACE JMP+JSR+SWAB DOES NOT GO HIGH.
                                ;*
                                ;*      IF EITHER IRCB IR(14:9)04 DOES NOT GO LOW OR E63 IS BAD
```

```
5314                                    ;*      EXECUTION WILL GO FROM D12.10 TO EXC.00.
5315                                    ;*      IF IRCB E63 IS BAD (PIN 10 OR 4&5 FLOATING) EXECUTION WILL
5316                                    ;*      GO TO RSD.00 CAUSING A TRAP TO LOCATION 10. THIS FAILURE
5317                                    ;*      WOULD INCREMENT THE DST REG. BEFORE THE TRAP.
5318                                    ;*
5319                                    ;*      IF THE INSTRUCTION FAILS THEN ONE OF THE JSR STATES FAILED.
5320                                    ;*
5321                                    ;*      ROM FLOW-2,135,34,201,274,275,32
5322                                    ;;*****************************************************************
5323  007274  005200                    TST75:  INC     R0              ;INC TST NUMBER
5324  007276  012706  001076                    MOV     #1076,SP        ;INITIALIZE SP
5325  007302  012705  007322                    MOV     #T67,R5         ;PUT ADDRESS OF T67A IN R5
5326  007306  010701                            MOV     PC,R1           ;PUT RANDOM NUMBER IN R1
5327  007310                            SYNC75:
5328  007310  000277                    T67A:   SCC                     ;ENSURE ALL CC'S SET
5329  007312                            IUT75:
5330  007312  004125                            JSR     R1,(R5)+        ;EXECUTE INSTRUCTION UNDER TEST
5331  007314  100001                    T67B:   BPL     T67C            ;BRANCH IF N CLEARED
5332  007316  000000                            HALT                    ;PCB DID NOT LOAD
5333                                                                    ;FOR LOOPING CHANGE TO "BR TST75+2" (767)
5334  007320                            T67C:
5335  007320  000000                            HALT                    ;FORK B FAILED TO EXC.00(SEE ABOVE)
5336                                                                    ;FOR LOOPING CHANGE TO "BR TST75+2" (766)
5337  007322  022716  007310           T67:    CMP     #T67A,(SP)      ;DID R1 GET STACKED?
5338  007326  001401                            BEQ     4$              ;BRANCH IF YES
5339  007330  000000                            HALT                    ;REGISTER DID NOT GET STACKED
5340                                                                    ;FOR LOOPING CHANGE TO "BR TST75+2" (762)
5341  007332  022701  007314           4$:     CMP     #T67B,R1                ;DID R1 GET LOADED?
5342  007336  001401                            BEQ     5$              ;BRANCH IF YES
5343  007340  000000                            HALT                    ;REGISTER DID NOT LOAD
5344                                                                    ;FOR LOOPING CHANGE TO "BR TST75+2" (756)
5345  007342  022706  001074           5$:     CMP     #1074,SP        ;DID SP GET DECREMENTED?
5346  007346  001401                            BEQ     TST76           ;;BRANCH IF YES
5347  007350  000000                            HALT                    ;SP DID NOT DECREMENT
5348                                                                    ;FOR LOOPING CHANGE TO "BR TST75+2" (752)
5349
5350
5351                                    ;;*****************************************************************
5352                                    ;*THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM3*DM0*-DF7*SR0(1) THEN
5353                                    ;*A BIN*SM3*DM0*DF7*SR0(0) THEN A BIN*SM3*DM0*DF7*SR0(1) INSTRUCTION,
5354                                    ;*BUT NO ADDITIONAL LOGIC IS TESTED.
5355                                    ;;*****************************************************************
5356
5357
5358                                    ;;*****************************************************************
5359                                    ;*TEST 76       SEVEN MICROSTATES (BIN*SM5*DM0*-DF7*SR0(0))
5360                                    ;*
5361                                    ;*      FORK A SHOULD NOT FAIL.
5362                                    ;*
5363                                    ;*      IF BEN14*FEN4 FAILS EXECUTION WILL GO TO D00.90
5364                                    ;*      CAUSING A SM4 INSTRUCTION TO BE EXECUTED. THIS WILL ONLY
5365                                    ;*      OCCUR IF EITHER IRCC SRCM5 DOES NOT GO LOW OR IF IRCC E28 IS BAD.
5366                                    ;*
5367                                    ;*      ROM FLOW-24,23,27,317,143,146,205
5368                                    ;;*****************************************************************
5369  007352  005200                    TST76:  INC     R0              ;INCREMENT TEST NUMBER
```

```
5370  007354  012705  001166                  MOV     #$TMP2,R5       ;PUT ADDRESS OF $TMP2 IN R5
5371  007360  010567  171600                  MOV     R5,$TMP1        ;PUT ADDRESS OF $TMP2 IN $TMP1
5372  007364  012715  100000                  MOV     #BIT15,(R5)     ;SET SIGN BIT IN $TMP2
5373  007370  005001                          CLR     R1              ;ENSURE R1 CLEAR
5374  007372  000240                  SYNC76: NOP
5375  007374                          IUT76:
5376  007374  015501                          MOV     @-(R5),R1       ;EXECUTE INSTRUCTION UNDER TEST
5377  007376  022701  100000                  CMP     #BIT15,R1       ;DID INSTRUCTION WORK?
5378  007402  001405                          BEQ     TST77           ;;BRANCH IF YES
5379  007404  020167  171554                  CMP     R1,$TMP1        ;DID MODE 4 EXECUTE?
5380  007410  001001                          BNE     1$              ;BRANCH IF NO
5381  007412  000000                          HALT                    ;EITHER IRCC SRCM5 NOT GOING LOW OR IRCC E28 BAD
5382                                                                  ;FOR LOOPING CHANGE TO "BR TST76+2" (760)
5383  007414                          1$:
5384  007414  000000                          HALT                    ;INSTRUCTION FAILED
5385                                                                  ;FOR LOOPING CHANGE TO "BR TST76+2" (757)
5386
5387                                  ;;**********************************************************************
5388                                  ;*THE LOGICAL SEQUENCE WOULD NEXT EXECUTE A BIN*SM12*DM12*SR0(0)*DR0(1)*
5389                                  ;*P/CLASS THEN A BIN*SM12*DM12*SR0(1)*DR0(1)P/CLASS INSTRUCTION, BUT
5390                                  ;*NO ADDITIONAL LOGIC IS TESTED.
5391                                  ;;**********************************************************************
5392
5393
5394                                  ;;**********************************************************************
5395                                  ;*TEST 77        SEVEN MICROSTATES (BIN*+SM12*DM3*0/CLASS)
5396                                  ;*
5397                                  ;*      IF IRCC C FORK MUX INPUT B2 IS NOT GOING LOW OR IRCC E40
5398                                  ;*      IS BAD A DM2 WILL BE EXECUTED.
5399                                  ;*
5400                                  ;*      THE ONLY OTHER  POSSIBLE FAILURE IS STATE D30.80.
5401                                  ;*
5402                                  ;*      ROM FLOW-21,27,113,221,233,311,157
5403                                  ;;**********************************************************************
5404  007416  005200                  TST77:  INC     R0              ;INCREMENT TEST NUMBER
5405  007420  005067  171542                  CLR     $TMP2           ;ENSURE $TMP2 CLEAR
5406  007424  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
5407  007430  012715  001166                  MOV     #$TMP2,(R5)     ;PUT ADDRESS OF $TMP2 IN $TMP1
5408  007434  000240                  SYNC77: NOP
5409  007436                          IUT77:
5410  007436  011535                          MOV     (R5),@(R5)+     ;EXECUTE INSTRUCTION UNDER TEST
5411  007440  024515                          CMP     -(R5),(R5)      ;DID INSTRUCTION WORK?
5412  007442  001405                          BEQ     TST100          ;;BRANCH IF YES
5413  007444  022745  001166                  CMP     #$TMP2,-(R5)    ;DID DM2 GET EXECUTED?
5414  007450  001001                          BNE     1$              ;BRANCH IF NO
5415  007452  000000                          HALT                    ;EITHER C FORK MUX INPUT B2
5416                                                                  ;NOT GOING LOW OR IRCC E40 BAD
5417                                                                  ;FOR LOOPING CHANGE TO "BR TST77+2" (762)
5418  007454                          1$:
5419  007454  000000                          HALT                    ;INSTRUCTION FAILED
5420                                                                  ;FOR LOOPING CHANGE TO "BR TST77+2" (761)
5421
5422
5423                                  ;;**********************************************************************
5424                                  ;*THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM4*SR0(0)*DR0(0)*
5425                                  ;*P/CLASS FOLLOWED BY A BIN*SM12*DM4*SR0(0)*DR0(1)*[TST.B+BIT.B+CMP.B]
```

```
                                        ;*FOLLOWED BY A BIN*SM12*DM4*SRO(1)*DRO(0)*P/CLASS FOLLOWED BY A
                                        ;*BIN*SM12*DM4*SRO(1)*DRO(1)*[TST.B+BIT.B+CMP.B] INSTRUCTION, BUT NO
                                        ;*ADDITIONAL LOGIC IS TESTED.
                                        ;;*****************************************************************


                                        ;;*****************************************************************
                                        ;*TEST 100      SEVEN MICROSTATES (BIN*SM12*DM6*O/CLASS)
                                        ;*
                                        ;*      IF FORK C FAILS EXECUTION WILL GO TO D45.90 AND A
                                        ;*      DM4 WILL BE EXECUTED. THIS WILL ONLY HAPPEN IF IRCC E39 PIN 5
                                        ;*      IS NOT GOING LOW. THIS WILL CAUSE AN RTI SINCE THE LOCATION
                                        ;*      FOLLOWING THE INSTRUCTION CONTAINS 000002.
                                        ;*
                                        ;*      THE ONLY OTHER FAILURE WOULD BE CAUSED BY STATE D67.80 BEING BAD.
                                        ;*
                                        ;*      ROM FLOW-21,27,117,6,251,122,157
                                        ;;*****************************************************************
007456  005200                  TST100: INC     R0              ;INCREMENT TEST NUMBER
007460  012706  001076                  MOV     #1076,SP        ;INITIALIZE THE SP
007464  012746  000340                  MOV     #PR7,-(SP)      ;PUT PRIORITY LEVEL 7 ON STACK
007470  012746  007526                  MOV     #T73,-(SP)      ;PUT ADDRESS OF T73 ON STACK
007474  005067  171466                  CLR     $TMP2           ;ENSURE $TMP2 CLEAR
007500  012705  001164                  MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
007504  012715  100000                  MOV     #BIT15,(R5)     ;SET SIGN BIT IN $TMP1
007510  000240                  SYN100: NOP
007512                          IUT100:
007512  011565  000002                  MOV     (R5),2(R5)      ;EXECUTE INSTRUCTION UNDER TEST
007516  005767  171444                  TST     $TMP2           ;DID INSTRUCTION WORK?
007522  100402                          BMI     TST101          ;;BRANCH IF YES
007524  000000                          HALT                    ;INSTRUCTION FAILED
                                                                ;FOR LOOPING CHANGE TO "BR TST100+2" (755)
007526                          T73:
007526  000000                          HALT                    ;IRCC E39(5) IS NOT GOING LOW
                                                                ;FOR LOOPING CHANGE TO "BR TST100+2" (754)


                                        ;;*****************************************************************
                                        ;*THE LOGICAL SEQUENCE WOULD NEXT TEST THE INSTRUCTIONS BETWEEN
                                        ;*BIN*SM4*DM12*SRO(0)*DRO(1)*[TST.B+BIT.B+CMP.B] AND BIN*SM5*DM0*DF7*SRO(1)
                                        ;*BUT NOT ADDITIONAL LOGIC IS TESTED.
                                        ;;*****************************************************************


                                                .SBTTL
                                        ;;*****************************************************************
                                        ;*TEST 101      EIGHT MICROSTATES (BIN*SM7*DM0*-DF7*SRO(0))
                                        ;*
                                        ;*      FORK A SHOULD NOT FAIL.
                                        ;*
                                        ;*      IF FEN4*BEN14 FAILS EXECUTION WILL GO TO D00.90 CAUSING
                                        ;*      A SM6 TO BE EXECUTED. THIS WILL ONLY HAPPEN IF EITHER
                                        ;*      IRCC SRCM7 DOES NOT GO LOW OR IF IRCC E28(1) IS BAD.
                                        ;*
                                        ;*      ROM FLOW-26,54,141,142,317,143,146,205
                                        ;;*****************************************************************
```

```
5482  007530  005200                    TST101: INC     R0              ;INCREMENT TEST NUMBER
5483  007532  012767  001164  171426            MOV     #$TMP1,$TMP2    ;PUT ADDRESS OF $TMP1 IN $TMP2
5484  007540  012767  100000  171416            MOV     #BIT15,$TMP1    ;SET SIGN BIT IN $TMP1
5485  007546  012705  001164                    MOV     #$TMP1,R5       ;PUT ADDRESS OF $TMP1 IN R5
5486  007552  005001                            CLR     R1              ;ENSURE R1 CLEAR
5487  007554  000240                    SYN101: NOP
5488  007556                            IUT101:
5489  007556  017501  000002                    MOV     @2(R5),R1       ;EXECUTE INSTRUCTION UNDER TEST
5490  007562  005701                            TST     R1              ;DID R1 GET SIGN BIT SET?
5491  007564  100405                            BMI     TST102          ;;BRANCH IF YES
5492  007566  022701  001164                    CMP     #$TMP1,R1       ;DID SRCM6 GET EXECUTED?
5493  007572  001001                            BNE     1$              ;BRANCH IF NO
5494  007574  000000                            HALT                    ;EITHER IRCC SRCM7 DOES NOT GO LOW OR IRCC E28 BAD
5495                                                                    ;FOR LOOPING CHANGE TO "BR TST101+2" (756)
5496  007576                            1$:
5497  007576  000000                            HALT                    ;INSTRUCTION FAILED
5498                                                                    ;FOR LOOPING CHANGE TO "BR TST101+2" (755)
5499
5500
5501                                    ;;**************************************************************
5502                                    ;*THE LOGICAL SEQUENCE WOULD NEXT TEST A BIN*SM12*DM3*SR0(0)*DR0(0)*[TST.B+
5503                                    ;*BIT.B+CMP.B] BUT NO ADDITIONAL LOGIC IS TESTED.
5504                                    ;;**************************************************************
5505
5506
5507                                    ;;**************************************************************
5508                                    ;*TEST 102      EIGHT MICROSTATES (BIN*SM12*DM3*SR0(1)*DR0(0)*CMPB)
5509                                    ;*
5510                                    ;*      THE ONLY POSSIBLE FAILURE WOULD BE IN STATE D30.90
5511                                    ;*      SINCE ALL THE OTHER LOGIC HAS BEEN TESTED.
5512                                    ;*
5513                                    ;*      ROM FLOW-21,27,112,221,233,311,177,33
5514                                    ;;**************************************************************
5515  007600  005200                    TST102: INC     R0              ;INCREMENT TEST NUMBER
5516  007602  012767  001166  171354            MOV     #$TMP2,$TMP1    ;PUT ADDRESS OF $TMP2 IN $TMP1
5517  007610  012767  000377  171350            MOV     #377,$TMP2      ;SET LOW BYTE OF $TMP2 TO ALL ONES
5518  007616  012767  177400  171336            MOV     #177400,$TMP0   ;SET HIGH BYTE OF $TMP0 TO ALL ONES
5519  007624  012705  001163                    MOV     #$TMP0+1,R5     ;PUT ADDRESS OF $TMP0 HIGH BYTE IN R5
5520  007630  012701  001164                    MOV     #$TMP1,R1       ;PUT ADDRESS OF $TMP1 IN R1
5521  007634  000240                    SYN102: NOP
5522  007636                            IUT102:
5523  007636  121531                            CMPB    (R5),@(R1)+     ;EXECUTE INSTRUCTION UNDER TEST
5524  007640  001401                            BEQ     TST103          ;;BRANCH IF TEST OK
5525  007642  000000                            HALT                    ;STATE D30.90 FAILED
5526                                                                    ;FOR LOOPING CHANGE TO "BR TST102+2" (757)
5527
5528
5529                                    ;;**************************************************************
5530                                    ;*THE LOGICAL SEQUENCE WOULD NEXT TEST INSTRUCTIONS BIN*SM12*DM4*SR0(0)*DR0(1)*
5531                                    ;*P/CLASS THRU BIN*SM12*DM6*SR0(0)*DR0(0)*[TST.B+BIT.B+CMP.B] BUT NO
5532                                    ;*ADDITIONAL LOGIC IS TESTED.
5533                                    ;;**************************************************************
5534
5535
5536                                    ;;**************************************************************
5537                                    ;*TEST 103      EIGHT MICROSTATES (BIN*SM12*DM6*SR0(1)*DR0(0)*CMPB)
```

```
5538                                            ;*
5539                                            ;*      THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D67.90.
5540                                            ;*
5541                                            ;*      ROM FLOW-21,27,116,6,251,122,177,33
5542                                            ;;*****************************************************************
5543   007644  005200                           TST103: INC     R0              ;INCREMENT TEST NUMBER
5544   007646  012767  177400  171310                   MOV     #177400,$TMP1   ;SET HIGH BYTE OF $TMP1 TO ALL ONES
5545   007654  012705  001165                           MOV     #$TMP1+1,R5     ;PUT ADDRESS OF $TMP1 HI BYTE IN R5
5546   007660  012767  000377  171300                   MOV     #377,$TMP2      ;SET LOW BYTE OF $TMP2 TO ALL ONES
5547   007666  000240                           SYN103: NOP
5548   007670                                   IUT103:
5549   007670  121567  171272                           CMPB    (R5),$TMP2      ;EXECUTE INSTRUCTION UNDER TEST
5550   007674  001401                                   BEQ     TST104          ;;BRANCH IF TEST OK
5551   007676  000000                                   HALT                    ;STATE D67.90 FAILED
5552                                                                           ;FOR LOOPING CHANGE TO "BR TST103+2" (763)
5553                                                   .SBTTL
5554                                           ;;*****************************************************************
5555                                           ;*TEST 104      NINE  MICROSTATES (BIN*SM12*DM5*SRO(0)*DRO(0)*CMP.B)
5556                                           ;*
5557                                           ;*      THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.20.
5558                                           ;*
5559                                           ;*      ROM FLOW-21,27,115,161,231,233,311,177,33
5560                                           ;;*****************************************************************
5561   007700  005200                           TST104: INC     R0                      ;INCREMENT THE TEST NUMBER
5562   007702  012705  001162                           MOV     #$TMP0,R5       ;PUT ADDRESS OF $TMP0 IN R5
5563   007706  012767  000377  171246                   MOV     #377,$TMP0      ;SET LOW BYTE OF $TMP0 TO ALL ONES
5564   007714  012701  001166                           MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
5565   007720  012767  000377  171240                   MOV     #377,$TMP2      ;SET LOW BYTE OF $TMP2 TO ALL ONES
5566   007726  012767  001166  171230                   MOV     #$TMP2,$TMP1    ;PUT ADDRESS OF $TMP2 IN $TMP1
5567   007734  000240                           SYN104: NOP
5568   007736                                   IUT104:
5569   007736  021551                                   CMP     (R5),@-(R1)     ;EXECUTE INSTRUCTION UNDER TEST
5570   007740  001401                                   BEQ     TST105          ;;GO TO NEXT TEST
5571   007742  000000                                   HALT                    ;STATE D50.20 IS BAD
5572                                                                           ;FOR LOOPING CHANGE TO "BR TST104+2" (757)
5573                                           ;;*****************************************************************
5574                                           ;*TEST 105      EIGHT MICROSTATES (BIN*SM12*DM5*SRO(1)*DRO(0)*CMPB)
5575                                           ;*
5576                                           ;*      THE ONLY POSSIBLE FAILURE IN THIS TEST IS STATE D50.30.
5577                                           ;;*****************************************************************
5578   007744  005200                           TST105: INC     R0                      ;INCREMENT THE TEST NUMBER
5579   007746  012705  001163                           MOV     #$TMP0+1,R5     ;PUT ADDRESS OF $TMP0 HIGH BYTE IN R5
5580   007752  012767  177400  171202                   MOV     #177400,$TMP0   ;SET HIGH BYTE OF $TMP0 TO ALL ONES
5581   007760  012701  001166                           MOV     #$TMP2,R1       ;PUT ADDRESS OF $TMP2 IN R1
5582   007764  012767  000377  171174                   MOV     #377,$TMP2      ;SET LOW BYTE OF $TMP2 TO ALL ONES
5583   007772  012767  001166  171164                   MOV     #$TMP2,$TMP1    ;PUT ADDRESS OF $TMP2 IN $TMP1
5584   010000  121551                                   CMPB    (R5),@-(R1)     ;EXECUTE INSTRUCTION UNDER TEST
5585   010002  001401                                   BEQ     TST106          ;;GO TO NEXT TEST
5586   010004  000000                                   HALT                    ;STATE D50.30 IS BAD
5587                                                                           ;FOR LOOPING CHANGE TO "BR TST105+2" (760)
5588
5589                                           ;;*****************************************************************
5590                                           ;*TEST 106      WRITE/READ PSW
5591                                           ;*
5592                                           ;*      THIS TEST VERIFIES THAT THE PSW CAN BE READ THRU THE DATA MUX.
5593                                           ;*      IF THE TEST FAILS ONE OF MANY THINGS COULD BE BAD WHICH CANNOT BE
```

```
                                        ;*      DETERMINED IN THIS DIAGNOSTIC.
                                        ;*      THIS TEST REQUIRES THAT SCCE PS ADRS GETS TO TMC,
                                        ;*      THAT THE TMC DMUX SELECT LINES GET TO PDR, THAT THE PSW
                                        ;*      BITS GET TO THE DMUX, THAT SCCE INTERNAL ADDRESS GETS TO TMC,
                                        ;*      AND SCCA VAOO GETS TO UBC.
                                        ;;*****************************************************************
010006  005200                  TST106: INC     R0              ;INCREMENT THE TEST NUMBER
010010  012737  000240  177776          MOV     #PR5,@#PSW      ;SET PRIORITY BITS WITH A DATO
010016  012701  000257                  MOV     #257,R1         ;PUT VALUE OF CC'S IN R1
010022  000277                          SCC                     ;SET ALL THE CC'S WITH A CCOP INSTR
010024  020137  177776                  CMP     R1,@#PSW        ;EXECUTE TEST MODE
010030  001416                          BEQ     TST107          ;;BRANCH IF READ PSW WORKS
010032  012701  177776                  MOV     #PSW,R1         ;GET ADDRESS OF PSW
010036  000277                          SCC                     ;SET ALL THE CONDITION CODES
010040  020137  177776                  CMP     R1,@#PSW        ;DID DMUX SELECT BUS REG?
010044  001001                          BNE     1$              ;BRANCH IF NO
010046  000000                          HALT                    ;EITHER TMCB E29 BAD OR SCCE PS ADDRESS
                                                                ;NOT GETTING TO TMCD AS A HIGH
                                                                ;FOR LOOPING CHANGE TO "BR TST106+2" (760)
010050  005001                  1$:     CLR     R1
010052  000277                          SCC
010054  020137  177776                  CMP     R1,@#PSW        ;DOES TMCD LOW BYTE ENABLE GO HIGH?
010060  001001                          BNE     2$              ;BRANCH IF YES
010062  000000                          HALT                    ;EITHER TMCD LO BYTE EN DOES
                                                                ;NOT GO HIGH OR IT DOES NOT GET
                                                                ;THRU TO PDRE
                                                                ;FOR LOOPING CHANGE TO "BR TST106+2" (752)
010064                          2$:
010064  000000                          HALT                    ;TEST FAILED, SEE TEST DESCRIPTION
                                                                ;FOR LOOPING CHANGE TO "BR TST106+2" (751)
                                        ;;*****************************************************************
                                        ;*TEST 107      RTI
                                        ;*
                                        ;*      IF FORK A FAILS EXECUTION WILL GO TO ONE OF THREE STATES.
                                        ;*      RSD.00 WILL CAUSE A TRAP TO LOCATION 4. THIS WOULD HAPPEN
                                        ;*      IF RACF (HALT:OP CD 7) DOES NOT GO HIGH.
                                        ;*      STATE D12.01 WOULD CAUSE AN ODD ADDRESS TRAP SINCE R0
                                        ;*      WILL CONTAIN A 1. THIS WILL HAPPEN IF RACE E7 IS BAD.
                                        ;*      HLT.00 WILL CAUSE THE PROCESSOR TO HALT ON THE INSTRUCTION
                                        ;*      UNDER TEST AND WILL  OCCUR IF RACF E17 IS BAD.
                                        ;*      IF THE INSTRUCTION DOESN'T WORK THEN ONE OF THE RTI MACHINE
                                        ;*      STATES IS BAD.
                                        ;*
                                        ;*      ROM FLOW-12,156,212,213,214,215,172
                                        ;;*****************************************************************
010066  005200                  TST107: INC     R0              ;INCREMENT TEST NUMBER
010070  012706  001100                  MOV     #STACK,SP       ;INITIALIZE THE STACK
010074  012746  000340                  MOV     #PR7,-(SP)      ;PUT PRIORITY LEVEL 7 ON STACK
010100  012746  010120                  MOV     #T71,-(SP)      ;PUT ADDRESS OF T71 ON STACK
010104  012705  000001                  MOV     #1,R5           ;PUT ODD ADDRESS IN R5.
010110  000240                  SYN107: NOP
010112                          IUT107:
010112  000002                          RTI                     ;EXECUTE INSTRUCTION UNDER TEST
010114  000240                          NOP                     ;IF THE PROCESSOR HALTS HERE
                                                                ;RACF E17 IS BAD(AFIR51(1)*(HALT:OP CD 7))
010116  000000                          HALT                    ;PCB DID NOT GET LOADED
```

```
                                                        ;FOR LOOPING CHANGE TO "BR TST107+2" (764)
010120  013705  177776          T71:    MOV     @#PSW,R5        ;GET PSW & PUT IN R5
010124  042705  177437                  BIC     #^C<PR7>,R5     ;MASK OUT THE PSW
010130  020527  000340                  CMP     R5,#PR7         ;DID PSW GET LOADED?
010134  001401                          BEQ     TST110          ;;BRANCH IF YES
010136  000000                          HALT                    ;EITHER PDRD E70(12) DOES NOT GO
                                                        ;HIGH ON LOAD PS AND KERNEL MODE
                                                        ;OR THE PRIORITY MUX ON PDRD IS BAD
                                                        ;FOR LOOPING CHANGE TO "BR TST107+2" (754)


                                ;;*****************************************************************
                                ;*THE RTT WILL NOT BE TESTED HERE SINCE THE ONLY POSSIBLE FORK A
                                ;*FAILURE WOULD CAUSE AN RTI TO BE EXECUTED. THE T BIT FUNCTIONS OF
                                ;*THE RTI & RTT ARE TESTED IN PART 2.
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 110       EMT AND TRAP
                                ;*
                                ;*      FORK A SHOULD NOT FAIL.
                                ;*      THE INSTRUCTIONS ARE EXECUTED AND THE STACK IS CHECKED TO
                                ;*      VERIFY THAT EVERYTHING WORKED OK.
                                ;*
                                ;*      ROM FLOW-0,345,354,SVC.00-SVC.90
                                ;;*****************************************************************
010140  005200                  TST110: INC     R0              ;INCREMENT TEST NUMBER
010142  012737  000340  177776          MOV     #PR7,@#PSW      ;SET PRIORITY LEVEL AT  7
010150  012706  001100                  MOV     #STACK,SP       ;INITIALIZE THE STACK
010154  012737  010250  000030          MOV     #1$,@#EMTVEC    ;PUT ADDRESS OF 1$ IN EMT VECTOR
010162  012737  000240  000032          MOV     #PR5,@#EMTVEC+2 ;PUT PRIORITY LEVEL 5 IN EMTVEC +2
010170  012737  010304  000010          MOV     #4$,@#RESVEC    ;SETUP RESVEC
010176  012737  010306  000020          MOV     #5$,@#20        ;SETUP LOCATION 20
010204  012737  010310  000034          MOV     #6$,@#34        ;SETUP LOCATION 34
010212  012737  010312  000070          MOV     #7$,@#70        ;SETUP LOCATION 70
010220  012737  010314  000130          MOV     #8$,@#130       ;SETUP LOCATION 130
010226  012737  010316  000430          MOV     #9$,@#430       ;SETUP LOCATION 430
010234  012737  010320  000630          MOV     #10$,@#630      ;SETUP LOCATION 630
010242  000277                          SCC                     ;PUT PSW IN KNOWN CONFIGURATION
010244  104377                          EMT     377             ;EXECUTE INSTRUCTION UNDER TEST
010246  000000                          HALT                    ;NEW PC FAILED TO GET LOADED
                                                        ;FOR LOOPING CHANGE TO "BR TST110+2" (735)
010250  022726  010246          1$:     CMP     #1$-2,(SP)+     ;DID CORRECT PC GET STACKED?
010254  001401                          BEQ     2$              ;BRANCH IF YES
010256  000000                          HALT                    ;OLD PC DID NOT STACK CORRECTLY
                                                        ;FOR LOOPING CHANGE TO "BR TST110+2" (731)
010260  022716  000357          2$:     CMP     #357,(SP)       ;DID PSW GET STACKED PROPERLY?
010264  001401                          BEQ     3$              ;BRANCH IF YES
010266  000000                          HALT                    ;EITHER PSW DID NOT GET STACKED
                                                        ;PROPERLY OR PSW <7:5> BITS
                                                        ;DO NOT WORK
                                                        ;FOR LOOPING CHANGE TO "BR TST110+2" (725)
010270  000257                  3$:     CCC
010272  022737  000240  177776          CMP     #240,@#PSW      ;DID NEW PSW LOAD PROPERLY?
010300  001427                          BEQ     TST111          ;;BRANCH IF YES
010302  000000                          HALT                    ;EITHER PSW DID NOT GET LOADED
                                                        ;PROPERLY OR PSW <7:5> BITS
```

```
5706
5707
5708  010304                              4$:
5709  010304  000000                              HALT                    ;DO NOT WORK
5710                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (717)
5711                                                                      ;EITHER IRCD EMT IS NOT GOING HIGH
5712                                                                      ;OR DAPE TVD3 IS NOT GOING HIGH
5713  010306                              5$:                             ;OR NOT GETTING TO THE ALU
5714  010306  000000                              HALT                    ;FOR LOOPING CHANGE TO "BR TST110+2" (716)
5715                                                                      ;EITHER DAPE TVD2 IS NOT GOING HIGH
5716                                                                      ;OR IT IS NOT GETTING TO THE ALU
5717  010310                              6$:                             ;FOR LOOPING CHANGE TO "BR TST110+2" (715)
5718  010310  000000                              HALT                    ;EITHER DAPE TVD1 IS STUCK HIGH
5719                                                                      ;OR THE LOW IS NOT GETTING TO THE ALU
5720                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (714)
5721  010312                              7$:
5722  010312  000000                              HALT                    ;EITHER DAPE TVD4 IS STUCK HIGH OR
5723                                                                      ;THE LOW IS NOT GETTING TO THE ALU
5724                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (713)
5725  010314                              8$:
5726  010314  000000                              HALT                    ;DAPE E24(BD) STUCK HIGH (CHIP FAILURE)
5727                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (712)
5728  010316                              9$:
5729  010316  000000                              HALT                    ;DAPE E25(BD) STUCK HIGH(CHIP FAILURE)
5730                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (711)
5731  010320                              10$:
5732  010320  000000                              HALT                    ;DAPE TVD5*07 STUCK HIGH
5733                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (710)
5734                                      ;NOTE:  THE ONLY WAY THE TRAP INSTRUCTION SHOULD FAIL IS THAT IT GETS
5735                                      ;THE WRONG VECTOR. IF THIS HAPPENS A HALT IN LOW CORE SHOULD OCCUR.
5736  010322  012737  010352  000034              MOV     #11$,@#TRAPVEC  ;PUT ADDRESS OF 11$ IN TRAP VECTOR
5737  010330  012737  010346  000010              MOV     #12$,@#RESVEC   ;SETUP RESVEC
5738  010336  012737  010350  000014              MOV     #13$,@#BPTVEC   ;SETUP LOCATION 14
5739  010344  104777                              TRAP    377             ;EXECUTE INSTRUCTION UNDER TEST
5740  010346                              12$:
5741  010346  000000                              HALT                    ;EITHER IRCD TRAP DOES NOT GO LOW
5742                                                                      ;OR IT IS NOT GETTING THRU DAPE
5743                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (675)
5744  010350                              13$:
5745  010350  000000                              HALT                    ;DAPE E7 IS BAD
5746                                                                      ;FOR LOOPING CHANGE TO "BR TST110+2" (674)
5747  010352  012737  000036  000034      11$:    MOV     #36,@#TRAPVEC   ;RESTORE .+2 TO TRAP VECTOR
5748                                      ;;*****************************************************************
5749                                      ;*TEST 111       IOT
5750                                      ;*
5751                                      ;*      FORK A SHOULD NOT FAIL.
5752                                      ;*
5753                                      ;*      IF THE TRAP VECTOR LOGIC FAILS THE TRAP VECTOR COULD COME
5754                                      ;*      OUT TO BE 0, 4, OR 24.
5755                                      ;*
5756                                      ;*      THE ONLY OTHER POSSIBLE FAILURE WOULD BE STATE TRP.01.
5757                                      ;*      IF THIS STATE FAILS TO LOAD THE DR THE TRAP VECTOR WILL
5758                                      ;*      BE WHATEVER IS IN R4.  IF IT FAILS TO LOAD THE BR THE OLD
5759                                      ;*      PS WILL FAIL TO BE STACKED.
5760                                      ;*
5761                                      ;*      ROM FLOW-14,354,(SVC.00-SVC.90) 355,65,357,360,367,37,25,41,222,300
```

```
5762                                    ;;*****************************************************************
5763  010360  005200                TST111: INC     R0              ;INCREMENT TEST NUMBER
5764  010362  012737  010440  000020        MOV     #1$,@#20        ;SETUP THE IOT VECTOR
5765  010370  012706  001100                MOV     #STACK,SP       ;SETUP THE SP
5766  010374  012737  010460  000000        MOV     #2$,@#0         ;SETUP LOCATION ZERO
5767  010402  012737  010462  000014        MOV     #3$,@#14        ;SETUP LOCATION 14
5768  010410  012704  000014                MOV     #14,R4          ;SETUP R4
5769  010414  012737  010464  000024        MOV     #4$,@#24        ;SETUP LOCATION 24
5770  010422  012737  010466  000004        MOV     #5$,@#ERRVEC    ;SET UP LOCATION 4
5771  010430  012737  000300  177776        MOV     #PR6,@#PSW      ;SETUP THE PSW
5772  010436  000004                        IOT                     ;EXECUTE INSTRUCTION UNDER TEST
5773  010440  042766  177437  000002  1$:   BIC     #177437,2(SP)   ;MASK OF THE PRIORITY BITS ON THE OLD PSW
5774  010446  022766  000300  000002        CMP     #300,2(SP)      ;DID OLD SP GET STACKED?
5775  010454  001405                        BEQ     SEQENC          ;;BRANCH IF YES
5776  010456  000000                        HALT                    ;STATE TRP.01 FAILED TO LOAD BR
5777                                                                ;FOR LOOPING CHANGE TO "BR TST111+2" (741)
5778  010460                          2$:
5779  010460  000000                        HALT                    ;EITHER DAPE TV04 DOES NOT GO HIGH
5780                                                                ;OR IT DOES NOT GET TO THE ALU.
5781                                                                ;FOR LOOPING CHANGE TO "BR TST111+2" (740)
5782  010462                          3$:
5783  010462  000000                        HALT                    ;STATE TRP.01 FILED TO LOAD DR
5784                                                                ;FOR LOOPING CHANGE TO "BR TST111+2" (737)
5785  010464                          4$:
5786  010464  000000                        HALT                    ;EITHER IRCD IOT DOES NOT GET TO
5787                                                                ;DAPE E7(4) AS A LOW OR E' IS BAD
5788                                                                ;FOR LOOPING CHANGE TO "BR TST111+2" (736)
5789  010466                          5$:
5790  010466  000000                        HALT                    ;EITHER IRCD IOT DOES NOT GO LOW
5791                                                                ;OR IT DOES NOT GET TO DAPE
5792                                                                ;FOR LOOPING CHANGE TO "BR TST111+2" (735)
5793  010470  016737  170404  177570 SEQENC: MOV     $PASS,@#SWR     ;DISPLAY PASS COUNT IN LIGHTS
5794  010476  005737  000042                TST     @#42            ;ACT 11 OR XXDP LOAD?
5795  010502  001017                        BNE     1$              ;BRANCH IF YES
5796  010504  005767  170374                TST     $ICNT           ;CHANGED PASS COUNT YET?
5797  010510  001407                        BEQ     2$              ;BRANCH IF NO
5798  010512  022767  000001  000164        CMP     #1,$EOPCT       ;EOP COUNT AT 1 YET?
5799  010520  001010                        BNE     1$              ;BRANCH IF NO
5800  010522  005067  170356                CLR     $ICNT           ;CLEAR CHANGED FLAG
5801  010526  000405                        BR      1$
5802  010530  005267  170350          2$:   INC     $ICNT           ;SET CHANGED FLAG
5803  010534  012767  010000  000142        MOV     #10000,$EOPCT   ;CHANGE PASS COUNT
5804  010542  022700  000111          1$:   CMP     #111,R0         ;IS TEST NUMBER OK?
5805  010546  001443                        BEQ     3$              ;BRANCH IF YES
5806  010550  012737  011652  000034        MOV     #$TRAP,@#TRAPVEC ;SETUP TRAP VECTOR
5807  010556  104400  010564                TYPE    ,65$            ;;TYPE ASCIZ STRING
5808  010562  000414                        BR      64$             ;;GET OVER THE ASCIZ
5809                                  ;;65$: .ASCIZ  <15><12>/TEST NUMBER(R0) IS  /
5810  010614                          64$:
5811  010614  010067  170334                MOV     R0,$REGO
5812  010620  016746  170330                MOV     $REGO,-(SP)     ;;SAVE $REGO FOR TYPEOUT
5813  010624  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
5814  010626  104400  010634                TYPE    ,67$            ;;TYPE ASCIZ STRING
5815  010632  000411                        BR      66$             ;;GET OVER THE ASCIZ
5816                                  ;;67$: .ASCIZ  /  SHOULD BE 111/<15><12>
5817  010656                          66$:
```

```
 5818  010656  005037  177746          3$:     CLR     @#CONTRL        ;ENABLE CACHE
 5819                                  ;;*****************************************************************
 5820                                          .MCALL  .$TYPOCT,.$TYPDEC,.$TYPE,.$TRAP
 5821                                  ;;*****************************************************************
 5822
 5823                                  .SBTTL  END OF PASS ROUTINE
 5824
 5825                                  ;*INCREMENT THE PASS NUMBER ($PASS)
 5826                                  ;*INDICATE END-OF-PROGRAM AFTER 144 PASSES THRU THE PROGRAM
 5827                                  ;*TYPE "END PASS"
 5828                                  ;*IF THERES A MONITOR GO TO IT
 5829                                  ;*IF THERE ISN'T JUMP TO TST1
 5830                                  ;*IF IT IS DESIRED TO HAVE A BELL INDICATE THE "END OF PASS" LOCATION
 5831                                  ;*$ENDMG CAN BE CHANGED TO 7.
 5832
 5833  010662                          $EOP:
 5834  010662  012737  011652  000034          MOV     #$TRAP,@#TRAPVEC        ;SETUP TRAP VECTOR
 5835  010670  005267  170204                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
 5836  010674  042767  100000  170176          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 5837  010702  005327                          DEC     (PC)+           ;;LOOP?
 5838  010704  000144                  $EOPCT: .WORD   144
 5839  010706  003015                          BGT     $DOAGN          ;;YES
 5840  010710  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 5841  010712  000144                  $ENDCT: .WORD   144
 5842  010714  010704                          $EOPCT
 5843  010716  104400  010746                  TYPE    ,$ENDMG         ;;TYPE "END PASS"
 5844  010722  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 5845  010726  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 5846  010730  000005                          RESET                   ;;CLEAR THE WORLD
 5847  010732  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 5848  010734  000240                          NOP                     ;;SAVE ROOM
 5849  010736  000240                          NOP                     ;;FOR
 5850  010740  000240                          NOP                     ;;ACT11
 5851  010742                          $DOAGN:
 5852  010742  000137  001210                  JMP     @#TST1          ;;RETURN
 5853  010746  005015  047105  020104  $ENDMG: .ASCII  <15><12>/END PASS/
 5854  010754  040520  051523
 5855  010760     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 5856          010764                          .EVEN
 5857                                  ;;*****************************************************************
 5858
 5859                                  .SBTTL  TYPE ROUTINE
 5860
 5861                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 5862                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 5863                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 5864                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 5865                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 5866                                  ;*
 5867                                  ;*CALL:
 5868                                  ;*1) USING A TRAP INSTRUCTION
 5869                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 5870                                  ;*OR
 5871                                  ;*      TYPE
 5872                                  ;*      MESADR
 5873                                  ;*
```

```
5874                                    ;*2) USING A JSR INSTRUCTION
5875                                    ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
5876                                    ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
5877                                    ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
5878
5879  010764  105767  170161          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
5880  010770  100002                          BPL     1$              ;;BR IF YES
5881  010772  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
5882  010774  000407                          BR      3$              ;;LEAVE
5883  010776  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
5884  011000  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
5885  011004  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
5886  011006  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
5887  011010  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
5888  011012  012600                          MOV     (SP)+,R0        ;;RESTORE R0
5889  011014  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
5890  011020  000002                          RTI                     ;;RETURN
5891  011022  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
5892  011026  001426                          BEQ     8$
5893  011030  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT
5894  011034  001004                          BNE     5$
5895  011036  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
5896  011040  104400  001173                  TYPE    ,$CRLF
5897  011044  000757                          BR      2$              ;;GET NEXT CHARACTER
5898  011046  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
5899  011052  126726  170072          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
5900  011056  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
5901  011060  016746  170062                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
5902                                                                  ;;AND THE NULL CHAR.
5903  011064  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
5904  011070  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
5905  011072  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
5906  011076  105367  000072                  DECB    $CHARCNT        ;;DON'T COUNT THE NULL AS A CHARACTER
5907  011102  000770                          BR      7$              ;;LOOP
5908
5909                                    ;;HORIZONTAL TAB PROCESSOR
5910
5911  011104  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
5912  011110  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
5913  011114  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
5914  011122  001372                          BNE     9$              ;;TAB STOP
5915  011124  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
5916  011126  000726                          BR      2$              ;;GET NEXT CHARACTER
5917  011130  105777  170006          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
5918  011134  100375                          BPL     $TYPEC
5919  011136  116677  000002  170000          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
5920  011144  122766  000015  000002          CMPB    #CR,2(SP)       ;;BRANCH IF
5921  011152  001003                          BNE     1$              ;;NOT <CR>
5922  011154  105067  000014                  CLRB    $CHARCNT        ;;
5923  011160  000406                          BR      $TYPEX          ;;EXIT
5924  011162  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;BRANCH IF
5925  011170  001402                          BEQ     $TYPEX          ;;<LF>
5926  011172  105227                          INCB    (PC)+           ;;INC SPACE
5927  011174  000000          $CHARCNT:.WORD  0                       ;;COUNT
5928  011176  000207          $TYPEX: RTS     PC
5929
```

```
                                ;;**********************************************************************

                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON--- ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

011200  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
011204  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
011212  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
011216  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
011222  000406                          BR      $TYPON
011224  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
011232  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
011240  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
011246  010346                          MOV     R3,-(SP)        ;;SAVE R3
011250  010446                          MOV     R4,-(SP)        ;;SAVE R4
011252  010546                          MOV     R5,-(SP)        ;;SAVE R5
011254  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
011260  005404                          NEG     R4
011262  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
011266  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
011272  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
011276  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
011302  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
011304  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
011306  000404                          BR      3$              ;;GO DO MSB
011310  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
011312  006105                          ROL     R5
011314  006105                          ROL     R5
011316  010503                          MOV     R5,R3
011320  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
011322  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
011326  100016                          BPL     7$              ;;BR IF NO
011330  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
011334  001002                          BNE     4$              ;;TEST FOR 0
011336  005704                          TST     R4              ;;SUPPRESS THIS 0?
```

```
5986   011340   001403                          BEQ     5$              ;;BR IF YES
5987   011342   005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5988   011344   052703   000060                 BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5989   011350   052703   000040         5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5990   011354   110367   000040                 MOVB    R3,8$           ;;SAVE FOR TYPING
5991   011360   104400   011420                 TYPE    ,8$             ;;GO TYPE THIS DIGIT
5992   011364   105367   000032         7$:     DECB    $OCNT           ;;COUNT BY 1
5993   011370   003347                         BGT     2$              ;;BR IF MORE TO DO
5994   011372   002402                         BLT     6$              ;;BR IF DONE
5995   011374   005204                         INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5996   011376   000744                         BR      2$              ;;GO DO THE LAST DIGIT
5997   011400   012605                 6$:     MOV     (SP)+,R5        ;;RESTORE R5
5998   011402   012604                         MOV     (SP)+,R4        ;;RESTORE R4
5999   011404   012603                         MOV     (SP)+,R3        ;;RESTORE R3
6000   011406   016666   000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
6001   011414   012616                         MOV     (SP)+,(SP)
6002   011416   000002                         RTI                     ;;RETURN
6003   011420      000                 8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
6004   011421      000                         .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
6005   011422      000                 $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
6006   011423      000                 $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
6007   011424   000000                 $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
6008                                   ;;**************************************************************
6009
6010                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6011
6012                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
6013                                   ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
6014                                   ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
6015                                   ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
6016                                   ;*REPLACED WITH SPACES.
6017                                   ;*CALL:
6018                                   ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
6019                                   ;*      TYPDS                   ;;GO TO THE ROUTINE
6020
6021   011426                          $TYPDS:
6022   011426   010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6023   011430   010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6024   011432   010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6025   011434   010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
6026   011436   010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
6027   011440   012746   020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
6028   011444   016605   000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
6029   011450   100004                         BPL     1$              ;;BR IF INPUT IS POS.
6030   011452   005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
6031   011454   112766   000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
6032   011462   005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
6033   011464   012703   011642                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
6034   011470   112723   000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
6035   011474   005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
6036   011476   016001   011632                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
6037   011502   160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
6038   011504   002402                         BLT     4$              ;;BR IF DONE
6039   011506   005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
6040   011510   000774                         BR      3$
6041   011512   060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
```

```
6042  011514  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
6043  011516  001002                          BNE     5$              ;;FALL THROUGH IF 0
6044  011520  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
6045  011522  100407                          BMI     7$              ;;BR IF YES
6046  011524  106316                  5$:     ASLB    (SP)            ;;MSD?
6047  011526  103003                          BCC     6$              ;;BR IF NO
6048  011530  116663  000001 177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
6049  011536  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
6050  011542  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
6051  011546  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
6052  011550  005720                          TST     (R0)+           ;;JUST INCREMENTING
6053  011552  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
6054  011556  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
6055  011560  003002                          BGT     8$              ;;GO TO EXIT
6056  011562  010502                          MOV     R5,R2           ;;GET THE LSD
6057  011564  000764                          BR      6$              ;;GO CHANGE TO ASCII
6058  011566  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
6059  011570  100003                          BPL     9$              ;;BR IF NO
6060  011572  116663  177777 177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
6061  011600  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
6062  011602  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
6063  011604  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
6064  011606  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
6065  011610  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
6066  011612  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
6067  011614  104400  011642                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
6068  011620  016666  000002 000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
6069  011626  012616                          MOV     (SP)+,(SP)
6070  011630  000002                          RTI                     ;;RETURN TO USER
6071  011632  023420                  $DTBL:  10000.
6072  011634  001750                          1000.
6073  011636  000144                          100.
6074  011640  000012                          10.
6075  011642  000004                  $DBLK:  .BLKW   4
6076                                  ;;*****************************************************************
6077
6078                                  .SBTTL  TRAP DECODER
6079
6080                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
6081                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
6082                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
6083                                  ;*GO TO THAT ROUTINE.
6084
6085  011652  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
6086  011654  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
6087  011660  005740                          TST     -(R0)           ;;BACKUP BY 2
6088  011662  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
6089  011664  016000  011672                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
6090  011670  000200                          RTS     R0              ;;GO TO ROUTINE
6091
6092
6093                                  .SBTTL  TRAP TABLE
6094
6095                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
6096                                  ;*BY THE "TRAP" INSTRUCTION.
6097
```

```
                                        ;       ROUTINE
                                        ;       -------
                                        $TRPAD:
                                                $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
                                                $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
                                                $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
                                                $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
                                                $TYPDS  ;;CALL=TYPDS    TRAP+10(104410) TYPE DECIMAL NUMBER (WITH SIGN)
                                        SUBTAB: .WORD   .
                                                .END
```

```
A      = 000107          3540#  3542#  3590#  3592#  3657#  3659#  3725#  3726#  3912#  3970#  3971#  4014#  4015#
                         4083#  4084#  4119#  4120#  4137#  4139#  4160#  4161#  4206#  4207#  4231#  4232#  4250#
                         4251#  4285#  4286#  4303#  4304#  4344#  4345#  4384#  4385#  4409#  4410#  4459#  4460#
                         4493#  4494#  4540#  4541#  4596#  4597#  4633#  4634#  4683#  4684#  4711#  4712#  4736#
                         4737#  4765#  4766#  4804#  4805#  4870#  4872#  4928#  4929#  4949#  4950#  5001#  5002#
                         5052#  5053#  5084#  5085#  5118#  5119#  5164#  5165#  5208#  5209#  5246#  5247#  5287#
                         5288#  5327#  5329#  5374#  5375#  5408#  5409#  5451#  5452#  5487#  5488#  5521#  5522#
                         5547#  5548#  5567#  5568#  5644#  5645#
ADCRO    001544          2907   2908   2909   2911#  2926
ADDRO    002250          3118   3119   3120   3122#  3136
AFTER    006572          4994   5004#  5010   5018   5026
ASLRO    001656          2956   2957   2958   2960#  2973
ASRRO    001730          2983   2984   2985   2987#  3000
BAD      005140          4283   4296#  4305   4308
BAD1     005432          4454   4467#
BICRO    002176          3091   3092   3093   3095#  3108
BISRO    002220          3103   3104   3105   3107#  3123
BITRO    002152          3078   3079   3080   3082#  3096
BIT0   = 000001          2145#  4903
BIT00  = 000001          2135#  2145
BIT01  = 000002          2134#  2144
BIT02  = 000004          2133#  2143
BIT03  = 000010          2132#  2142
BIT04  = 000020          2131#  2141
BIT05  = 000040          2130#  2140
BIT06  = 000100          2129#  2139   4859
BIT07  = 000200          2128#  2138
BIT08  = 000400          2127#  2137
BIT09  = 001000          2126#  2136
BIT1   = 000002          2144#
BIT10  = 002000          2125#
BIT11  = 004000          2124#
BIT12  = 010000          2123#
BIT13  = 020000          2122#
BIT14  = 040000          2121#  4708
BIT15  = 100000          2120#  3950   4118   4381   4405   4458   4489   4559   4628   4660   4735   4847   4948
                         5201   5242   5285   5372   5377   5450   5484
BIT2   = 000004          2143#
BIT3   = 000010          2142#
BIT4   = 000020          2141#  3548   3692
BIT5   = 000040          2140#
BIT6   = 000100          2139#
BIT7   = 000200          2138#  5205
BIT8   = 000400          2137#
BIT9   = 001000          2136#
BPTVEC= 000014           2152#  5738*
CACHVE= 000114           2159#
CLRRO    001460          2872   2873   2874   2876#  2889
CMPRO    002316          3143   3144   3145   3147#
COMRO    001524          2896   2897   2898   2900#  2912
CONTRL= 177746           2169#  2628*  5818*
CPUERR= 177766           2182#
CR     = 000015          2057#  5920   5930
CRLF   = 000200          2058#  5893   5930
DECRO    001614          2933   2934   2935   2937#  2949
DISPLA= 177570           2052#
```

```
DUM    = 000007          3540#  3590#  3657#  3725#  3912#  3970#  4014#  4083#  4119#  4137#  4160#  4206#  4231#
                         4250#  4285#  4303#  4344#  4384#  4409#  4459#  4493#  4540#  4596#  4633#  4683#  4711#
                         4736#  4765#  4804#  4870#  4928#  4949#  5001#  5052#  5084#  5118#  5164#  5208#  5246#
                         5287#  5327#  5374#  5408#  5451#  5487#  5521#  5547#  5567#  5644#
EMTVEC= 000030            2155#  5678*  5679*
ERRVEC= 000004            2148#  5770*
ER25    003656            3728#
FORKA   003366            3545   3555#
GNS    = ****** U         2435   5809   5816   6101   6102   6103   6104   6105
GOOD    005120            4283#  4287
GRA1T   002354            3185   3189#  3205
GRA2T   002410            3204   3208#  3224
HIADRS= 177742            2167#
HITMIS= 177752            2171#
HT     = 000011           2055#  5891   5930
INCRD   001634            2944   2945   2946   2948#  2961
IOTVEC= 000020            2153#
IT      006420            4897   4900#  4902
IUT     004304            3913#
IUT100  007512            5452#
IUT101  007556            5488#
IUT102  007636            5522#
IUT103  007670            5548#
IUT104  007736            5568#
IUT107  010112            5645#
IUT24   003334            3542#
IUT25   003430            3592#
IUT26   003524            3659#
IUT27   003654            3726#
IUT32   004426            3971#
IUT33   004504            4015#
IUT34   004630            4084#
IUT35   004710            4120#
IUT36   004732            4139#
IUT37   004754            4161#
IUT40   005020            4207#
IUT41   005054            4232#
IUT42   005076            4251#
IUT43   005126            4286#  4304#
IUT44   005220            4345#
IUT45   005302            4385#
IUT46   005330            4410#
IUT47   005414            4460#
IUT50   005460            4494#
IUT51   005536            4541#
IUT52   005632            4597#
IUT53   005706            4634#
IUT55   005776            4684#
IUT56   006024            4712#
IUT57   006060            4737#
IUT60   006116            4766#
IUT61   006144            4805#
IUT62   006344            4872#
IUT63   006464            4929#
IUT64   006512            4950#
IUT65   006570            5002#
```

```
IUT66   006722          5053#
IUT67   006750          5085#
IUT70   007004          5119#
IUT71   007062          5165#
IUT72   007136          5209#
IUT73   007176          5247#
IUT74   007250          5288#
IUT75   007312          5329#
IUT76   007374          5375#
IUT77   007436          5409#
JMPTO   006002          4682    4686#
JMP1AD  005032          4204    4216#
JMP2AD  005056          4230    4234#
KDPAR0= 172360          2325#
KDPAR1= 172362          2326#
KDPAR2= 172364          2327#
KDPAR3= 172366          2328#
KDPAR4= 172370          2329#
KDPAR5= 172372          2330#
KDPAR6= 172374          2331#
KDPAR7= 172376          2332#
KDPDR0= 172320          2303#
KDPDR1= 172322          2304#
KDPDR2= 172324          2305#
KDPDR3= 172326          2306#
KDPDR4= 172330          2307#
KDPDR5= 172332          2308#
KDPDR6= 172334          2309#
KDPDR7= 172336          2310#
KERSTK= 001100          2042#
KIPAR0= 172340          2314#
KIPAR1= 172342          2315#
KIPAR2= 172344          2316#
KIPAR3= 172346          2317#
KIPAR4= 172350          2318#
KIPAR5= 172352          2319#
KIPAR6= 172354          2320#
KIPAR7= 172356          2321#
KIPDR0= 172300          2292#
KIPDR1= 172302          2293#
KIPDR2= 172304          2294#
KIPDR3= 172306          2295#
KIPDR4= 172310          2296#
KIPDR5= 172312          2297#
KIPDR6= 172314          2298#
KIPDR7= 172316          2299#
KSP    =%000006         2076#
LF     = 000012         2056#   5924    5930
LOADRS= 177740          2166#
MAINT  = 177750         2170#
MAPH0  = 170202         2409#
MAPH00= 170202          2345#   2409
MAPH01= 170206          2347#   2411
MAPH02= 170212          2349#   2413
MAPH03= 170216          2351#   2415
MAPH04= 170222          2353#   2417
```

```
MAPH05= 170226           2355#   2419
MAPH06= 170232           2357#   2421
MAPH07= 170236           2359#   2423
MAPH1 = 170206           2411#
MAPH10= 170242           2361#
MAPH11= 170246           2363#
MAPH12= 170252           2365#
MAPH13= 170256           2367#
MAPH14= 170262           2369#
MAPH15= 170266           2371#
MAPH16= 170272           2373#
MAPH17= 170276           2375#
MAPH2 = 170212           2413#
MAPH20= 170302           2377#
MAPH21= 170306           2379#
MAPH22= 170312           2381#
MAPH23= 170316           2383#
MAPH24= 170320           2385#
MAPH25= 170326           2387#
MAPH26= 170332           2389#
MAPH27= 170336           2391#
MAPH3 = 170216           2415#
MAPH30= 170342           2393#
MAPH31= 170346           2395#
MAPH32= 170352           2397#
MAPH33= 170356           2399#
MAPH34= 170362           2401#
MAPH35= 170366           2403#
MAPH36= 170372           2405#
MAPH37= 170376           2407#
MAPH4 = 170222           2417#
MAPH5 = 170226           2419#
MAPH6 = 170232           2421#
MAPH7 = 170236           2423#
MAPL0 = 170200           2408#
MAPL00= 170200           2344#   2408
MAPL01= 170204           2346#   2410
MAPL02= 170210           2348#   2412
MAPL03= 170214           2350#   2414
MAPL04= 170220           2352#   2416
MAPL05= 170224           2354#   2418
MAPL06= 170230           2356#   2420
MAPL07= 170234           2358#   2422
MAPL1 = 170204           2410#
MAPL10= 170240           2360#
MAPL11= 170244           2362#
MAPL12= 170250           2364#
MAPL13= 170254           2366#
MAPL14= 170260           2368#
MAPL15= 170264           2370#
MAPL16= 170270           2372#
MAPL17= 170274           2374#
MAPL2 = 170210           2412#
MAPL20= 170300           2376#
MAPL21= 170304           2378#
MAPL22= 170310           2380#
```

DEKBAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
MAPL23= 170314          2382#
MAPL24= 170320          2384#
MAPL25= 170324          2386#
MAPL26= 170330          2388#
MAPL27= 170334          2390#
MAPL3 = 170214          2414#
MAPL30= 170340          2392#
MAPL31= 170344          2394#
MAPL32= 170350          2396#
MAPL33= 170354          2398#
MAPL34= 170360          2400#
MAPL35= 170364          2402#
MAPL36= 170370          2404#
MAPL37= 170374          2406#
MAPL4 = 170220          2416#
MAPL5 = 170224          2418#
MAPL6 = 170230          2420#
MAPL7 = 170234          2422#
MEMERR= 177744          2168#
MMR0  = 177572          2193#   2197
MMR1  = 177574          2194#   2198
MMR2  = 177576          2195#   2199
MMR3  = 172516          2196#   2200
MMVEC = 000250          2161#
MOVBRO  002126          3058    3059    3060    3061    3067#
MOVRO   002130          3065    3070#   3083
N     = 000735          2647#   2667#   2673#   2693#   2699#   2718#   2724#   2743#   2749#   2771#   2788#   2794#   2812#
                        2818#   2835#   2841#   2857#   2877#   2889#   2901#   2912#   2926#   2938#   2949#   2961#   2973#
                        2988#   3000#   3014#   3027#   3035#   3048#   3068#   3071#   3083#   3096#   3108#   3123#   3136#
                        3148#   3178#   3181#   3186#   3197#   3200#   3205#   3216#   3219#   3224#   3232#   3236#   3240#
                        3244#   3251#   3255#   3267#   3271#   3277#   3281#   3293#   3297#   3303#   3307#   3319#   3323#
                        3329#   3333#   3345#   3349#   3355#   3359#   3371#   3375#   3381#   3385#   3397#   3401#   3407#
                        3411#   3446#   3450#   3455#   3459#   3464#   3468#   3473#   3477#   3482#   3486#   3491#   3495#
                        3559#   3562#   3567#   3595#   3610#   3613#   3617#   3620#   3624#   3673#   3677#   3684#   3688#
                        3695#   3698#   3702#   3707#   3738#   3741#   3756#   3759#   3784#   3790#   3798#   3803#   3812#
                        3819#   3828#   3835#   3844#   3851#   3859#   3862#   3871#   3880#   3889#   3925#   3928#   3932#
                        3940#   3948#   3974#   3983#   4024#   4027#   4036#   4041#   4044#   4055#   4058#   4062#   4093#
                        4097#   4100#   4123#   4142#   4167#   4170#   4211#   4214#   4219#   4236#   4290#   4293#   4297#
                        4308#   4319#   4351#   4355#   4360#   4365#   4388#   4416#   4419#   4465#   4468#   4500#   4505#
                        4553#   4557#   4563#   4568#   4571#   4603#   4606#   4610#   4639#   4643#   4646#   4664#   4668#
                        4687#   4715#   4740#   4744#   4748#   4770#   4811#   4814#   4879#   4883#   4888#   4892#   4899#
                        4905#   4932#   4954#   4958#   5008#   5016#   5020#   5024#   5035#   5057#   5089#   5125#   5131#
                        5134#   5139#   5171#   5175#   5214#   5217#   5253#   5257#   5294#   5297#   5332#   5335#   5339#
                        5343#   5347#   5381#   5384#   5415#   5419#   5456#   5459#   5494#   5497#   5525#   5551#   5571#
                        5586#   5610#   5617#   5622#   5649#   5655#   5689#   5693#   5697#   5704#   5709#   5714#   5718#
                        5722#   5726#   5729#   5732#   5741#   5745#   5776#   5779#   5783#   5786#   5790#
NEGR5   003446          3597    3598    3599    3601#
NEXTT   005102          4249    4254#
OVER    002344          3173    3183#
OVER2   002400          3192    3202#
OVER3   002434          3211    3221#
PC    =%000007          2079#   3543#   3660    3727    3743    3750    3910    3914*   3943*   3968    3972*   3980*   4542*
                        4546#   4561#   4598    4601    4608    4860*   4862    4866*   4868*   4875*   4900    5326    5837*
                        5840#   5847#   5898*   5905*   5912*   5926*   5928*
PIRQ  = 177772          2050#
PIRQVE= 000240          2160#
```

DEKBAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
POINT   005634          4592    4599#
POINT1  006350          4840*   4853    4875#
POINT2  007064          5163    5167#   5169
POSERR  003526          3662#   3666    3682    3686
PR0     = 000000        2082#
PR1     = 000040        2083#
PR2     = 000100        2084#
PR3     = 000140        2085#
PR4     = 000200        2086#
PR5     = 000240        2087#   5601    5679
PR6     = 000300        2088#   5771
PR7     = 000340        2089#   5446    5641    5652    5653    5676
PS      = 177776        2047#   2048
PSW     = 177776        2048#   3665    4169    4855    4999    5601*   5604    5606    5608    5615    5651    5676*   5702
                        5771*
PWRVEC= 000024          2154#
RESVEC= 000010          2149#   5680*   5737*
ROLR0   001700          2968    2969    2970    2972#
RORR0   001572          2921    2922    2923    2925#   2938    2988
ROUT0   002330          3174#   3178    3181
ROUT1   002364          3193#   3197    3200
ROUT2   002420          3212#   3216    3219
R0      =%000000        2061#   2069    2871*   2880*   2883    2892*   2895*   2904*   2906*   2915*   2917*   2920*   2923*
                        2932*   2941*   2943*   2952*   2955*   2964*   2966*   2967*   2976*   2978*   2979*   2982*   2991*
                        2994*   3003*   3005*   3008*   3017*   3018*   3021*   3029*   3031*   3032*   3038*   3039*   3042*
                        3051*   3053*   3054*   3057*   3064    3074*   3077    3086*   3087*   3090*   3099*   3100*   3102*
                        3111*   3113*   3114*   3117*   3126*   3127*   3130*   3139*   3142    3170*   3172    3174*   3176
                        3184    3189*   3191    3193*   3195    3203    3208*   3210    3212*   3214    3222    3263*   3265
                        3269    3273*   3275    3279    3289*   3291    3295    3299*   3301    3305    3315*   3317    3321
                        3325*   3327    3331    3341*   3343    3347    3351*   3353    3357    3367*   3369    3373    3377*
                        3379    3383    3393*   3395    3399    3403*   3405    3409    3420*   3421*   3422*   3423*   3424*
                        3425*   3426*   3427*   3428*   3429*   3430*   3431*   3432*   3433*   3434*   3435*   3436*   3437*
                        3438*   3439*   3440*   3441*   3442*   3443    3444    3448    3452    3453    3457    3461    3462
                        3466    3470    3471    3475    3479    3480    3484    3488    3489    3493    3510*   3511*   3512*
                        3513*   3514*   3515*   3516*   3517*   3530*   3585*   3655*   3723*   3778*   3909*   3967*   4005*
                        4079*   4114*   4133*   4156*   4202*   4229*   4248*   4279*   4339*   4379*   4404*   4447*   4487*
                        4529*   4590*   4626*   4658*   4681*   4706*   4729*   4761*   4800*   4837*   4922*   4944*   4991*
                        5048*   5080*   5114*   5161*   5200*   5241*   5281*   5323*   5369*   5404*   5444*   5482*   5515*
                        5543*   5561*   5578*   5600*   5639*   5675*   5763*   5804    5811    5844*   5847    5883    5884*
                        5885    5888*   6022    6032*   6036    6052    6053    6066*   6085    6086*   6087    6088*   6089*
                        6090*
R1      =%000001        2062#   2070    3171*   3175*   3183*   3184*   3195*   3214*   3226*   3229*   3264*   3265    3269
                        3274*   3275    3279    3443*   3444    3448    3602*   3603*   3604    3670*   3732*   3734*   3735*
                        3736    3744*   3746*   3747    3751*   3753*   3754    3780*   3781*   3782*   3786*   3787*   3788*
                        3795*   3796*   3800*   3801*   3807*   3808*   3809*   3810*   3814*   3815*   3816*   3817*   3823*
                        3824*   3825*   3826*   3830*   3831*   3832*   3833*   3839*   3840*   3841*   3842*   3846*   3847*
                        3848*   3849*   3855*   3857*   3866*   3867*   3868*   3869*   3875*   3876*   3877*   3878*   3884*
                        3885*   3886*   3887*   3919    3922    3945*   3946    3951*   3977*   3978*   4008*   4020*   4021
                        4022    4038*   4039    4052    4116*   4117*   4121    4135*   4136*   4140    4157*   4159    4203*
                        4216    4249*   4252    4311*   4312*   4316    4340*   4343*   4346*   4347    4349    4357*   4358
                        4362*   4363*   4381*   4382    4405*   4407    4458*   4461    4489*   4490    4492*   4495    4498
                        4531*   4532    4533*   4534    4538*   4539    4592*   4593*   4594    4628*   4629    4631*   4632*
                        4635*   4708*   4709    4731*   4732    4734*   4742    4746    4803*   4806*   4807    4846*   4849*
                        4873*   4877    4881    4885    4925*   4926*   4927*   4930    4946*   4956    4994*   4995    4999*
                        5004    5010*   5011*   5012*   5014    5026*   5027    5028*   5029*   5031*   5033    5166*   5167
                        5169    5203*   5204    5205*   5206*   5210    5212    5243*   5244    5245*   5248*   5284*   5289*
```

```
                    5326*  5330*  5341   5373*  5376*  5377   5379   5486*  5489*  5490   5492   5520*  5523
                    5564*  5569   5581*  5584   5602*  5604   5606*  5608   5613*  5615   6023   6036*  6037
                    6041   6065*
R10     =%000000    2069#
R11     =%000001    2070#
R12     =%000002    2071#
R13     =%000003    2072#
R14     =%000004    2073#
R15     =%000005    2074#
R2      =%000002    2063#  2071   3176*  3190*  3194*  3202*  3203*  3246*  3248*  3290*  3291   3295   3300*
                    3301   3305   3452*  3453   3457   3733*  3745*  3752*  3794*  3800   3809   3816   3825
                    3832   3841   3848   3946*  3950*  3951   4006*  4016   4030   4034   4039   4080*  4082*
                    4085   4205*  4209   4497   4847*  4848   4850*  4851*  4852*  4854*  4855*  4874*  4903
                    4997*  5004*  5022   5285*  5289   5292   6024   6035*  6039*  6042   6049*  6050*  6051
                    6056*  6064*
R3      =%000003    2064#  2072   3227*  3230   3234*  3316*  3317   3321   3326*  3327   3331   3461*  3462
                    3466   4011*  4012*  4017   4733*  4734   4735*  4738   4858*  4860   4866   4998*  5006
                    5013*  5014   5964   5973*  5979*  5980*  5983*  5988*  5989*  5990   5999*  6025   6033*
                    6034*  6048*  6051*  6060*  6061*  6063*
R4      =%000004    2065#  2073   3209*  3213*  3221*  3222*  3342*  3343   3347   3352*  3353   3357   3470*
                    3471   3475   3586*  3601*  3606   3622   4284*  4287*  4853*  4854   4859*  4868   4875
                    4885*  4886   4895*  4896   5081*  5082*  5083   5086   5768*  5965   5967*  5968*  5969*
                    5970   5971*  5985   5987*  5995*  5998*
R5      =%000005    2066#  2074   3228*  3238   3242*  3368*  3369   3373   3378*  3379   3383   3479*  3480
                    3484   3531*  3532*  3533*  3534*  3535*  3536*  3537*  3538*  3539*  3543   3546*  3552*
                    3553*  3557*  3587*  3588*  3593*  3604   3608   3615   3656*  3660*  3669*  3670   3671
                    3681*  3682   3705*  3724*  3727*  3731*  3732   3733   3736   3743*  3744   3745   3747
                    3750*  3751   3752   3754   3910*  3914   3918*  3919*  3920   3922*  3923   3934*  3935*
                    3936   3938*  3939   3943   3968*  3972   3976*  3978   3980   4007*  4008   4012   4016*
                    4017   4019*  4021*  4022   4029*  4030   4032*  4034*  4052*  4053   4064*  4081*  4082
                    4090*  4091*  4115*  4117   4118*  4121   4134*  4136   4140   4158*  4159*  4162   4164
                    4166   4204*  4205   4208   4209   4230*  4233   4234   4281*  4288   4301*  4302*  4305*
                    4306   4310*  4313*  4314   4341*  4342*  4343   4346   4362   4363   4380*  4382*  4383*
                    4386   4406*  4407*  4408*  4411   4450*  4451   4454*  4455   4456*  4457*  4461*  4463
                    4488*  4490*  4491*  4495   4497*  4498   4537*  4539*  4542   4545*  4546   4559*  4561
                    4591*  4594*  4595*  4598   4600*  4601   4608   4627*  4629*  4630*  4635   4637   4641
                    4660*  4661   4662*  4663*  4666   4682*  4685   4707*  4709*  4710*  4713*  4730*  4732*
                    4738*  4746   4762*  4763*  4767*  4768*  4801*  4802*  4803   4806   4845*  4848*  4873
                    4886   4923*  4924*  4930   4945*  4946   4947*  4948*  4951*  4952   4993*  4995*  4996*
                    5003*  5005   5027*  5030*  5031   5049*  5050*  5051*  5054*  5055   5117*  5120*  5121
                    5123   5129   5137*  5162*  5163*  5166   5201*  5202   5204*  5207*  5210   5242*  5248
                    5251   5282*  5283   5325*  5330   5370*  5371   5372*  5376   5406*  5407*  5410*  5411
                    5413   5449*  5450*  5453*  5485*  5489   5519*  5523   5545*  5549   5562*  5569   5579*
                    5584   5643*  5651*  5652*  5653   5966   5972*  5974*  5976*  5977*  5978*  5979   5997*
                    6026   6028*  6030*  6037*  6041*  6056   6062*
R50K.     003510    3605   3622#
R6      =%000006    2067#  2075   3247*  3249   3253*  3394*  3395   3399   3404*  3405   3409   3488*  3489
                    3493
R7      =%000007    2068#  2079
SBCR0     001752    2995   2996   2997   2999#  3014
SDPAR0=   172260    2281#
SDPAR1=   172262    2282#
SDPAR2=   172264    2283#
SDPAR3=   172266    2284#
SDPAR4=   172270    2285#
SDPAR5=   172272    2286#
```

```
SDPAR6= 172274          2287#
SDPAR7= 172276          2288#
SDPDR0= 172220          2259#
SDPDR1= 172222          2260#
SDPDR2= 172224          2261#
SDPDR3= 172226          2262#
SDPDR4= 172230          2263#
SDPDR5= 172232          2264#
SDPDR6= 172234          2265#
SDPDR7= 172236          2266#
SEQENC  010470          5775    5793#
SIPAR0= 172240          2270#
SIPAR1= 172242          2271#
SIPAR2= 172244          2272#
SIPAR3= 172246          2273#
SIPAR4= 172250          2274#
SIPAR5= 172252          2275#
SIPAR6= 172254          2276#
SIPAR7= 172256          2277#
SIPDR0= 172200          2248#
SIPDR1= 172202          2249#
SIPDR2= 172204          2250#
SIPDR3= 172206          2251#
SIPDR4= 172210          2252#
SIPDR5= 172212          2253#
SIPDR6= 172214          2254#
SIPDR7= 172216          2255#
SIZEHI= 177762          2179#
SIZELO= 177760          2177#
SKIP    005122          4282    4284#
SP     =%000006         2075#   2076    2077    2078    3497*   3498*   3499*   3500*   3501*   3502*   3503*   3504*   3505*
                        3506*   3507*   3508*   3548*   3549*   3556*   3686    4218*   4280*   4413    4415*   4448*   4449*
                        4451*   4452*   4453*   4455*   4462*   4530*   4532*   4534*   4535*   4536*   4659*   4661*   4844*
                        4890    4992*   5018    5324*   5337    5345    5445*   5446*   5447*   5640*   5641*   5642*   5677*
                        5691    5695    5765*   5773*   5774    5812*   5883*   5884    5885*   5887    5888    5889*   5891
                        5893    5895    5899    5901*   5903*   5911*   5915    5919    5920    5924    5956*   5957    5958
                        5959*   5964*   5965*   5966*   5972    5997    5998    5999    6000*   6001*   6022*   6023*   6024*
                        6025*   6026*   6027*   6028    6031*   6044    6046*   6048    6058    6060    6062    6063    6064
                        6065    6066    6068*   6069*   6085*   6086
SR0    = 177572         2197#
SR1    = 177574         2198#
SR2    = 177576         2199#
SR3    = 172516         2200#
SSP    =%000006         2077#
STACK  = 001100         2041#   2042    2043    2044    4280    4415    4462    4530    4659    4844    5640    5677    5765
START   001202          2439    2628#
STKLMT= 177774          2049#
SUBR0   002274          3131    3132    3133    3135#   3148
SUBTAB  011704          3731    6106#
SUPSTK= 000700          2043#
SWABR0  002000          3009    3010    3011    3013#   3027
SWR    = 177570         2051#   2052    5793*
SW0    = 000001         2117#
SW00   = 000001         2107#   2117
SW01   = 000002         2106#   2116
SW02   = 000004         2105#   2115
```

```
SW03   = 000010            2104#   2114
SW04   = 000020            2103#   2113
SW05   = 000040            2102#   2112
SW06   = 000100            2101#   2111
SW07   = 000200            2100#   2110
SW08   = 000400            2099#   2109
SW09   = 001000            2098#   2108
SW1    = 000002            2116#
SW10   = 002000            2097#
SW11   = 004000            2096#
SW12   = 010000            2095#
SW13   = 020000            2094#
SW14   = 040000            2093#
SW15   = 100000            2092#
SW2    = 000004            2115#
SW3    = 000010            2114#
SW4    = 000020            2113#
SW5    = 000040            2112#
SW6    = 000100            2111#
SW7    = 000200            2110#
SW8    = 000400            2109#
SW9    = 001000            2108#
SXTR0    002024            3022    3023    3024    3026#   3035
SXT2     002040            3034#   3048
SYNC24   003332            3540#
SYNC25   003426            3590#
SYNC26   003522            3657#
SYNC27   003652            3725#
SYNC31   004302            3912#
SYNC32   004424            3970#
SYNC33   004502            4014#
SYNC34   004626            4083#
SYNC35   004706            4119#
SYNC36   004730            4137#
SYNC37   004752            4160#
SYNC40   005016            4206#
SYNC41   005052            4231#
SYNC42   005074            4250#
SYNC43   005124            4285#   4303#
SYNC44   005216            4344#
SYNC45   005300            4384#
SYNC46   005326            4409#
SYNC47   005412            4459#
SYNC50   005456            4493#
SYNC51   005534            4540#
SYNC52   005630            4596#
SYNC53   005704            4633#
SYNC55   005774            4683#
SYNC56   006022            4711#
SYNC57   006056            4736#
SYNC60   006114            4765#
SYNC61   006142            4804#
SYNC62   006342            4857    4870#
SYNC63   006462            4928#
SYNC64   006510            4949#
SYNC65   006570            5001#
```

```
SYNC66  006720          5052#
SYNC67  006746          5084#
SYNC70  007002          5118#
SYNC71  007060          5164#
SYNC72  007134          5208#
SYNC73  007174          5246#
SYNC74  007246          5287#
SYNC75  007310          5327#
SYNC76  007372          5374#
SYNC77  007434          5408#
SYN100  007510          5451#
SYN101  007554          5487#
SYN102  007634          5521#
SYN103  007666          5547#
SYN104  007734          5567#
SYN107  010110          5644#
SYSTID= 177764          2181#
TBITVE= 000014          2150#
TESTCC  003410          3565#
TKVEC = 000060          2157#
TPVEC = 000064          2158#
TRAPVE= 000034          2156#   5736#   5747#   5806#   5834#
TRTVEC= 000014          2151#
TSTRO   001502          2884    2885    2886    2888#   2901
TST1    001210          2639#   2647    5852
TST10   001370          2792    2806#   2812
TST100  007456          5412    5444#   5456    5459
TST101  007530          5455    5482#   5494    5497
TST102  007600          5491    5515#   5525
TST103  007644          5524    5543#   5551
TST104  007700          5550    5561#   5571
TST105  007744          5570    5578#   5586
TST106  010006          5585    5600#   5610    5617    5622
TST107  010066          5605    5639#   5649    5655
TST11   001410          2816    2829#   2835
TST110  010140          5654    5675#   5689    5693    5697    5704    5709    5714    5718    5722    5726    5729    5732
                        5741    5745
TST111  010360          5703    5763#   5776    5779    5783    5786    5790
TST12   001430          2839    2851#   2857
TST13   001442          2855    2869#   2877
TST14   002320          3170#   3186
TST15   002526          3254    3263#   3267    3271    3277    3281    3446    3450    3455    3459    3464    3468    3473
                        3477    3482    3486    3491    3495
TST16   002566          3280    3289#   3293    3297    3303    3307
TST17   002626          3306    3315#   3319    3323    3329    3333
TST2    001226          2645    2661#   2667
TST20   002666          3332    3341#   3345    3349    3355    3359
TST21   002726          3358    3367#   3371    3375    3381    3385
TST22   002766          3384    3393#   3397    3401    3407    3411
TST23   003026          3410    3420#
TST24   003306          3530#   3559    3562    3567
TST25   003414          3566    3585#   3595    3610    3613    3617    3620    3624
TST26   003516          3623    3655#   3673    3677    3684    3688    3695    3698    3702    3707
TST27   003646          3706    3723#   3738    3741    3756    3759
TST3    001246          2671    2686#   2693
TST30   003742          3748    3778#   3784
```

```
TST31   004274          3888   3909#  3925   3928   3932   3940   3948
TST32   004416          3967#  3974   3983
TST33   004460          3982   4005#  4024   4027   4036   4041   4044   4055   4058   4062
TST34   004612          4035   4079#  4093   4097   4100
TST35   004670          4087   4114#  4123
TST36   004716          4122   4133#  4142
TST37   004740          4141   4156#  4167   4170
TST4    001270          2697   2711#  2718
TST40   005004          4165   4202#  4211   4214   4219
TST41   005044          4217   4229#  4236
TST42   005066          4235   4248#
TST43   005104          4279#  4290   4293   4297
TST44   005202          4317   4339#  4351   4355   4360   4365
TST45   005262          4364   4379#  4388
TST46   005310          4387   4404#  4416   4419
TST47   005352          4412   4447#  4465   4468
TST5    001312          2722   2736#  2743
TST50   005434          4464   4487#  4500   4505
TST51   005476          4496   4529#  4553   4557   4563   4568   4571
TST52   005612          4544   4590#  4603   4606   4610
TST53   005660          4609   4626#  4639   4643   4646
TST54   005734          4636   4658#  4664   4668
TST55   005766          4667   4681#  4687
TST56   006004          4706#  4715
TST57   006032          4714   4729#  4740   4744   4748
TST6    001334          2747   2765#  2771
TST60   006102          4747   4761#  4770
TST61   006126          4769   4800#  4811   4814
TST62   006166          4808   4837#  4879   4883   4888   4892   4898   4905
TST63   006436          4904   4922#  4932
TST64   006472          4931   4944#  4954   4958
TST65   006532          4957   4991#  5008   5016   5020   5024   5035
TST66   006702          5034   5048#  5057
TST67   006734          5056   5080#  5089
TST7    001350          2769   2782#  2788
TST70   006764          5088   5114#  5125   5131   5134   5139
TST71   007046          5138   5161#  5171   5175
TST72   007104          5168   5200#  5214   5217
TST73   007154          5211   5241#  5253   5257
TST74   007220          5250   5281#  5294   5297
TST75   007274          5291   5323#  5332   5335   5339   5343   5347
TST76   007352          5346   5369#  5381   5384
TST77   007416          5378   5404#  5415   5419
TYPDS = 104410          6105#
TYPE  = 104400          5807   5814   5843   5896   5991   6067   6101#
TYPOC = 104402          5813   6102#
TYPON = 104406          6104#
TYPOS = 104404          6103#
T67     007322          5325   5337#
T67A    007310          5328#  5337
T67B    007314          5331#  5341
T67C    007320          5331   5334#
T71     010120          5642   5651#
T73     007526          5447   5458#
UDPAR0= 177660          2237#
UDPAR1= 177662          2238#
```

```
UDPAR2= 177664          2239#
UDPAR3= 177666          2240#
UDPAR4= 177670          2241#
UDPAR5= 177672          2242#
UDPAR6= 177674          2243#
UDPAR7= 177676          2244#
UDPDR0= 177620          2215#
UDPDR1= 177622          2216#
UDPDR2= 177624          2217#
UDPDR3= 177626          2218#
UDPDR4= 177630          2219#
UDPDR5= 177632          2220#
UDPDR6= 177634          2221#
UDPDR7= 177636          2222#
UIPAR0= 177640          2226#
UIPAR1= 177642          2227#
UIPAR2= 177644          2228#
UIPAR3= 177646          2229#
UIPAR4= 177650          2230#
UIPAR5= 177652          2231#
UIPAR6= 177654          2232#
UIPAR7= 177656          2233#
UIPDR0= 177600          2204#
UIPDR1= 177602          2205#
UIPDR2= 177604          2206#
UIPDR3= 177606          2207#
UIPDR4= 177610          2208#
UIPDR5= 177612          2209#
UIPDR6= 177614          2210#
UIPDR7= 177616          2211#
USESTK= 000600          2044#
USP    =%000006         2078#
XORR0   002064          3043    3044    3045    3047#   3068    3071
Y      = 000767         2650#   2668#   2675#   2694#   2700#   2719#   2725#   2744#   2750#   2772#   2789#   2796#   2813#
                        2819#   2836#   2842#   2858#   2878#   3179#   3182#   3187#   3198#   3201#   3206#   3217#   3220#
                        3225#   3233#   3237#   3241#   3245#   3252#   3256#   3268#   3272#   3278#   3282#   3294#   3298#
                        3304#   3308#   3320#   3324#   3330#   3334#   3346#   3350#   3356#   3360#   3372#   3376#   3382#
                        3386#   3398#   3402#   3408#   3412#   3447#   3451#   3456#   3460#   3465#   3469#   3474#   3478#
                        3483#   3487#   3492#   3496#   3596#   3791#   3799#   3804#   3813#   3820#   3829#   3836#   3845#
                        3852#   3860#   3863#   3872#   3881#   3890#   4320#

$BDADR  001122          2487#
$BDDAT  001126          2489#
$CHARC  011174          5906*   5913    5922*   5927#
$CMTAG  001100          2475#
$CM1   = 000003         2501#   2502#   2503#   2504#
$CM2   = 000006         2501#   2502#   2503#   2504#
$CM3   = 000003         2499#   2501
$CM4   = 000004         2504#   2505#   2506#   2507#   2508#
$CRLF   001173          2509#   5896    5930
$DBLK   011642          6033    6067    6075#
$DOAGN  010742          5839    5845    5851#
$DTBL   011632          6036    6071#
$ENDAD  010732          2462    5847#
$ENDCT  010712          5841#
$ENDMG  010746          5843    5853#
$ENULL  010760          5855#
```

```
$EOP     010662            5833#
$EOPCT   010704            5798    5803*   5838#   5842
$ERFLG   001103            2478#
$ERMAX   001115            2484#
$ERPSW   001176            2511#   3665*   3690    3692    4169*
$ERRPC   001116            2485#
$ERRTB   001202            2530#
$ERTTL   001112            2482#
$FILLC   001150            2497#   5899    5930
$FILLS   001147            2496#   5930
$GDADR   001120            2486#
$GDDAT   001124            2488#
$GET42   010722            5844#
$HD    = 000002            2036
$ICNT    001104            2479#   5796    5800*   5802*
$ITEMB   001114            2483#
$LF      001174            2510#   5930
$LPADR   001106            2480#
$LPERR   001110            2481#
$NULL    001146            2495#   5901    5930
$NWTST= 000001             2629#   2631    2651#   2653    2676#   2678    2701#   2703    2726#   2728    2756#   2758    2773#
                           2775    2797#   2799    2820#   2822    2843#   2845    2859#   2861    3152#   3154    3257#   3259
                           3283#   3285    3309#   3311    3335#   3337    3361#   3363    3387#   3389    3413#   3415    3520#
                           3522    3570#   3572    3627#   3629    3709#   3711    3761#   3763    3891#   3893    3952#   3954
                           3985#   3987    4046#   4048    4102#   4104    4125#   4127    4144#   4146    4181#   4183    4221#
                           4223    4238#   4240    4255#   4257    4322#   4324    4367#   4369    4391#   4393    4429#   4431
                           4470#   4472    4508#   4510    4576#   4578    4613#   4615    4648#   4650    4670#   4672    4692#
                           4694    4717#   4719    4750#   4752    4790#   4792    4816#   4818    4915#   4917    4934#   4936
                           4961#   4963    5037#   5039    5070#   5072    5102#   5104    5150#   5152    5189#   5191    5229#
                           5231    5270#   5272    5307#   5309    5358#   5360    5394#   5396    5432#   5434    5471#   5473
                           5507#   5509    5536#   5538    5554#   5556    5573#   5575    5589#   5591    5624#   5626    5666#
                           5668    5748#   5750
$OCNT    011422            5963*   5992*   6005#
$OMODE   011424            5958*   5962*   5967    5970*   5981*   6007#
$PASS    001100            2476#   4856    5793    5835*   5836*   5853
$QUES    001172            2508#
$RDCHR= ****** U           6106
$RDDEC= ****** U           6106
$RDLIN= ****** U           6106
$RDOCT= ****** U           6106
$REGAD   001152            2499#
$REGO    001154            2501#   4166*   5005*   5811*   5812
$REG1    001156            2502#
$REG2    001160            2503#
$RTI     005610            4533    4570#
$SAVRE= ****** U           6106
$SETUP= 000024             2428#   5820#   5835
$STUP  = 177777            2428#   5820#
$SVPC  = 000204            2460#   2465
$SWR   = 160000            2036#   2508    2640    2662    2687    2712    2737    2766    2783    2807    2830    2852    2870
                           3171    3264    3290    3316    3342    3368    3394    3421    3531    3586    3656    3724    3779
                           3910    3968    4006    4080    4115    4134    4157    4203    4230    4249    4280    4340    4380
                           4405    4448    4488    4530    4591    4627    4659    4682    4707    4730    4762    4801    4838
                           4923    4945    4992    5049    5081    5115    5162    5201    5242    5282    5324    5370    5405
                           5445    5483    5516    5544    5562    5579    5601    5640    5676    5764    5828    5835    5846
                           5852    5853
```

DEKBAB.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
$TKB    001140          2492#
$TKS    001136          2491#
$TMP0   001162          2504#   5284    5518*   5519    5562    5563*   5579    5580*
$TMP1   001164          2505#   3918    3920    3923    4011    4019    4020    4053    4080    4089*   4090    4095    4116
                        4135    4158    4164    4340    4347    4357    4380    4406    4413    4456    4488    4627    4707
                        4730    4762    4801    4845    4923    4945    4993    5013    5033    5049    5083*   5086*   5087
                        5115*   5116*   5117    5162    5202*   5244*   5251    5283*   5292    5371*   5379    5406    5449
                        5483    5484*   5485    5492    5516*   5520    5544*   5545    5566*   5583*
$TMP2   001166          2506#   4007    4641    4731    4809    4846    4925    4952    5203    5212    5243    5249    5282
                        5286*   5290    5370    5405*   5407    5413    5448*   5454    5483*   5516    5517*   5546*   5549
                        5564    5565*   5566    5581    5582*   5583
$TMP3   001170          2507#
$TN   = 000112          2011#   2036    2629    2640#   2645    2647    2651    2662#   2667    2671    2676    2687#   2693
                        2697    2701    2712#   2718    2722    2726    2737#   2743    2747    2756    2766#   2769    2771
                        2773    2783#   2788    2792    2797    2807#   2812    2816    2820    2830#   2835    2839    2843
                        2852#   2855    2857    2859    2870#   2877    3152    3171#   3186    3254    3257    3264#   3267
                        3271    3277    3280    3281    3283    3290#   3293    3297    3303    3306    3307    3309    3316#
                        3319    3323    3329    3332    3333    3335    3342#   3345    3349    3355    3358    3359    3361
                        3368#   3371    3375    3381    3384    3385    3387    3394#   3397    3401    3407    3410    3411
                        3413    3421#   3520    3531#   3540    3542    3559    3562    3566    3567    3570    3586#   3590
                        3592    3595    3610    3613    3617    3620    3623    3624    3627    3656#   3657    3659    3673
                        3677    3684    3688    3695    3698    3702    3706    3707    3709    3724#   3725    3726    3738
                        3741    3748    3756    3759    3761    3779#   3784    3888    3891    3910#   3912    3925    3928
                        3932    3940    3948    3952    3968#   3970    3971    3974    3982    3983    3985    4006#   4014
                        4015    4024    4027    4035    4036    4041    4044    4046    4055    4058    4062    4080#   4083
                        4084    4087    4093    4097    4100    4102    4115#   4119    4120    4122    4123    4125    4134#
                        4137    4139    4141    4142    4144    4157#   4160    4161    4165    4167    4170    4181    4203#
                        4206    4207    4211    4214    4217    4219    4221    4230#   4231    4232    4235    4236    4238
                        4249#   4250    4251    4255    4280#   4285    4286    4290    4293    4297    4303    4304    4317
                        4322    4340#   4344    4345    4351    4355    4360    4364    4365    4367    4380#   4384    4385
                        4387    4388    4391    4405#   4409    4410    4412    4416    4419    4429    4448#   4459.   4460
                        4464    4465    4468    4470    4488#   4493    4494    4496    4500    4505    4508    4530#   4540
                        4541    4544    4553    4557    4563    4568    4571    4576    4591#   4596    4597    4603    4606
                        4609    4610    4613    4627#   4633    4634    4636    4639    4643    4646    4648    4659#   4664
                        4667    4668    4670    4682#   4683    4684    4687    4692    4707#   4711    4712    4714    4715
                        4717    4730#   4736    4737    4740    4744    4747    4748    4750    4762#   4765    4766    4769
                        4770    4790    4801#   4804    4805    4808    4811    4814    4816    4838#   4870    4872    4879
                        4883    4888    4892    4898    4904    4905    4915    4923#   4928    4929    4931    4932    4934
                        4945#   4949    4950    4954    4957    4958    4961    4992#   5001    5002    5008    5016    5020
                        5024    5034    5035    5037    5049#   5052    5053    5056    5057    5070    5081#   5084    5085
                        5088    5089    5102    5115#   5118    5119    5125    5131    5134    5138    5139    5150    5162#
                        5164    5165    5168    5171    5175    5189    5201#   5208    5209    5211    5214    5217    5229
                        5242#   5246    5247    5250    5253    5257    5270    5282#   5287    5288    5291    5294    5297
                        5307    5324#   5327    5329    5332    5335    5339    5343    5346    5347    5358    5370#   5374
                        5375    5378    5381    5384    5394    5405#   5408    5409    5412    5415    5419    5432    5445#
                        5451    5452    5455    5456    5459    5471    5483#   5487    5488    5491    5494    5497    5507
                        5516#   5521    5522    5524    5525    5536    5544#   5547    5548    5550    5551    5554    5562#
                        5567    5568    5570    5571    5573    5579#   5585    5586    5589    5601#   5605    5610    5617
                        5622    5624    5640#   5644    5645    5649    5654    5655    5666    5676#   5689    5693    5697
                        5703    5704    5709    5714    5718    5722    5726    5729    5732    5741    5745    5748    5764#
                        5776    5779    5783    5786    5790
$TPB    001144          2494#   4842    4843    5919*   5930
$TPFLG  001151          2498#   5879    5930
$TPS    001142          2493#   4838    4839    4840    4841    5917    5930
$TRAP   011652          5806    5834    6085#
$TRP  = 000012          6092#   6102#   6103#   6104#   6105#   6106#
```

```
$TRPAD  011672          6089    6100#
$TSTNM  001102          2477#
$TYPBN= ****** U        6106
$TYPDS  011426          6021#   6105
$TYPE   010764          5879#   6092    6101
$TYPEC  011130          5898    5905    5912    5917#   5918
$TYPEX  011176          5923    5925    5928#
$TYPOC  011224          5961#   6102
$TYPON  011240          5960    5963#   6104
$TYPOS  011200          5956#   6103
$SGET4= 000000          5846#
$STN  = 000105          2531#   2538#   2539#   2540#   2542#   2543#   2545#   2547#   2549#   2551#   2553#   2556#   2557#
                        2559#   2561#   2562#   2563#   2564#   2566#   2567#   2569#   2570#   2572#   2574#   2575#   2576#
                        2578#   2580#   2582#   2584#   2588#   2589#   2591#   2593#   2595#   2597#   2599#   2601#   2604#
                        2605#   2606#   2608#   2610#   2612#   2614#   2616#   2617#   2619#   2620#
$STPS   006422          4841*   4901#
$STRP = 000002          6091#   6102    6103    6104    6105    6106
$OFILL  011423          5957*   5961*   5971    6006#
.     = 011706          2431#   2435    2437#   2460    2461#   2463#   2465#   2473#   2511    2647    2667    2673    2693
                        2699    2718    2724    2743    2749    2771    2788    2794    2812    2818    2835    2841    2857
                        2875    2877    2887    2889    2899    2901    2910    2912    2924    2926    2936    2938    2947
                        2949    2959    2961    2971    2973    2986    2988    2998    3000    3012    3014    3025    3027
                        3033    3035    3046    3048    3066    3068    3071    3081    3083    3094    3096    3106    3108
                        3121    3123    3134    3136    3146    3148    3178    3181    3186    3197    3200    3205    3216
                        3219    3224    3232    3236    3240    3244    3251    3255    3267    3271    3277    3281    3293
                        3297    3303    3307    3319    3323    3329    3333    3345    3349    3355    3359    3371    3375
                        3381    3385    3397    3401    3407    3411    3446    3450    3455    3459    3464    3468    3473
                        3477    3482    3486    3491    3495    3559    3562    3567    3595    3600    3610    3613    3617
                        3620    3624    3673    3677    3684    3688    3695    3698    3702    3707    3738    3741    3756
                        3759    3784    3790    3798    3803    3812    3819    3828    3835    3844    3851    3859    3862
                        3871    3880    3889    3925    3928    3932    3940    3948    3974    3983    4024    4027    4036
                        4041    4044    4055    4058    4062    4093    4097    4100    4123    4142    4167    4170    4211
                        4214    4219    4236    4290    4293    4297    4308    4319    4351    4355    4360    4365    4388
                        4416    4419    4465    4468    4500    4505    4553    4557    4563    4568    4571    4603    4606
                        4610    4639    4643    4646    4664    4668    4687    4715    4740    4744    4748    4770    4811
                        4814    4879    4883    4888    4892    4898    4905    4932    4954    4958    5008    5016    5020
                        5024    5035    5057    5089    5125    5131    5134    5139    5171    5175    5214    5217    5253
                        5257    5294    5297    5332    5335    5339    5343    5347    5381    5384    5415    5419    5456
                        5459    5494    5497    5525    5551    5571    5586    5610    5617    5622    5649    5655    5689
                        5693    5697    5704    5709    5714    5718    5722    5726    5729    5732    5741    5745    5776
                        5779    5783    5786    5790    5810#   5853    5856#   5930    6075#   6106
```

```
COMA      3673#
COMAA     4388#
COMB      3676#   3677
COMBB     4500#
COMC      3697#   3698
COMCC     5125#
COMD      3940#
COMDD     5785#   5786
COME      3927#   3928    4057#   4058
COMEE     5778#   5779
COMENT    4570#   4571
COMFF     5789#   5790
COMG      4292#   4293
COMGG     5708#   5709
COMH      4292#   4297
COMHH     5713#   5714
COMI      4351#
COMII     5717#   5718
COMJ      4504#   4505
COMJJ     5721#   5722
COMK      4563#
COMKK     5725#   5726
COML      4686#   4687
COMLL     5728#   5729
COMMEN       1#   2427#
COMMM     5731#   5732
COMN      5171#
COMNN     5740#   5741
COMO      5253#
COMP      5415#   5697#
COMQ      5704#
COMX      5617#
COMY      5610#
COMZ      5655#
DOC       2011#   2538    2539    2540    2542    2543    2545    2547    2549    2551    2553    2556    2557    2559    2561
          2562    2563    2564    2566    2567    2569    2570    2572    2574    2575    2576    2578    2580    2582    2584
          2588    2589    2591    2593    2595    2597    2599    2601    2604    2605    2606    2608    2610    2612    2614
          2616    2617    2619    2620
DOCEXP    2011#   2538    2539    2540    2542    2543    2545    2547    2549    2551    2553    2556    2557    2559    2561
          2562    2563    2564    2566    2567    2569    2570    2572    2574    2575    2576    2578    2580    2582    2584
          2588    2589    2591    2593    2595    2597    2599    2601    2604    2605    2606    2608    2610    2612    2614
          2616    2617    2619    2620
ENDCOM       1#   2427#
ERROR     2045#
ERRORD    2011#   2646    2666    2672    2692    2698    2717    2723    2742    2748    2770    2787    2793    2811    2817
          2834    2840    2856    2876    2888    2900    2911    2925    2937    2948    2960    2972    2987    2999    3013
          3026    3034    3047    3067    3070    3082    3095    3107    3122    3135    3147    3178    3180    3186    3197
          3199    3205    3216    3218    3224    3232    3236    3240    3244    3251    3255    3267    3271    3277    3281
          3293    3297    3303    3307    3319    3323    3329    3333    3345    3349    3355    3359    3371    3375    3381
          3385    3397    3401    3407    3411    3446    3450    3455    3459    3464    3468    3473    3477    3482    3486
          3491    3495    3559    3561    3567    3595    3610    3612    3617    3619    3624    3673    3676    3684    3688
          3694    3697    3701    3707    3738    3740    3756    3758    3784    3790    3798    3803    3812    3819    3828
          3835    3844    3851    3859    3862    3871    3880    3889    3925    3927    3931    3940    3948    3974    3983
          4024    4026    4036    4041    4043    4055    4057    4061    4093    4097    4099    4123    4142    4167    4170
          4211    4213    4219    4236    4290    4292    4296    4308    4318    4351    4354    4360    4365    4388    4416
          4418    4465    4467    4500    4504    4553    4557    4563    4567    4570    4603    4605    4610    4639    4643
```

```
                4645    4664    4668    4686    4715    4740    4744    4748    4770    4811    4813    4879    4883    4888    4892
                4898    4905    4932    4954    4958    5008    5016    5020    5024    5035    5057    5089    5125    5131    5133
                5139    5171    5174    5214    5216    5253    5256    5294    5296    5332    5334    5339    5343    5347    5381
                5383    5415    5418    5456    5458    5494    5496    5525    5551    5571    5586    5610    5617    5621    5649
                5655    5689    5693    5697    5704    5708    5713    5717    5721    5725    5728    5731    5740    5744    5776
                5778    5782    5785    5789
ESCAPE             1#  2427#
HLT             2011#   2647    2667    2673    2693    2699    2718    2724    2743    2749    2771    2788    2794    2812    2818
                2835    2841    2857    2877    2889    2901    2912    2926    2938    2949    2961    2973    2988    3000    3014
                3027    3035    3048    3068    3071    3083    3096    3108    3123    3136    3148    3178    3181    3186    3197
                3200    3205    3216    3219    3224    3232    3236    3240    3244    3251    3255    3267    3271    3277    3281
                3293    3297    3303    3307    3319    3323    3329    3333    3345    3349    3355    3359    3371    3375    3381
                3385    3397    3401    3407    3411    3446    3450    3455    3459    3464    3468    3473    3477    3482    3486
                3491    3495    3559    3562    3567    3595    3610    3613    3617    3620    3624    3673    3677    3684    3688
                3695    3698    3702    3707    3738    3741    3756    3759    3784    3790    3798    3803    3812    3819    3828
                3835    3844    3851    3859    3862    3871    3880    3889    3925    3928    3932    3940    3948    3974    3983
                4024    4027    4036    4041    4044    4055    4058    4062    4093    4097    4100    4123    4142    4167    4170
                4211    4214    4219    4236    4290    4293    4297    4308    4319    4351    4355    4360    4365    4388    4416
                4419    4465    4468    4500    4505    4553    4557    4563    4568    4571    4603    4606    4610    4639    4643
                4646    4664    4668    4687    4715    4740    4744    4748    4770    4811    4814    4879    4883    4888    4892
                4898    4905    4932    4954    4958    5008    5016    5020    5024    5035    5057    5089    5125    5131    5134
                5139    5171    5175    5214    5217    5253    5257    5294    5297    5332    5335    5339    5343    5347    5381
                5384    5415    5419    5456    5459    5494    5497    5525    5551    5571    5586    5610    5617    5622    5649
                5655    5689    5693    5697    5704    5709    5714    5718    5722    5726    5729    5732    5741    5745    5776
                5779    5783    5786    5790
IDN             4892#
LABEL           2011#   3540    3590    3657    3725    3912    3970    4014    4083    4119    4137    4160    4206    4231    4250
                4285    4303    4344    4384    4409    4459    4493    4540    4596    4633    4683    4711    4736    4765    4804
                4870    4928    4949    5001    5052    5084    5118    5164    5208    5246    5287    5327    5374    5408    5451
                5487    5521    5547    5567    5644
LABEL1          2011#   3542    3592    3659    3726    3971    4015    4084    4120    4139    4161    4207    4232    4251    4286
                4304    4345    4385    4410    4460    4494    4541    4597    4634    4684    4712    4737    4766    4805    4872
                4929    4950    5002    5053    5085    5119    5165    5209    5247    5288    5329    5375    5409    5452    5488
                5522    5548    5568    5645
MSGD1           2672#   2673
MSGD10          3335#   3337
MSGD11          3361#   3363
MSGD12          3387#   3389
MSGD13          3413#   3415
MSGD2           2793#   2794
MSGD5           2859#   2861
MSGD6           3151#   3154
MSGD7           3257#   3259
MSGD8           3283#   3285
MSGD9           3309#   3311
MSG1            2629#   2631
MSG10           2843#   2845
MSG120          5748#   5750
MSG16           3520#   3522
MSG17           3570#   3572
MSG2            2651#   2653
MSG20           3627#   3629
MSG21           3709#   3711
MSG21A          3761#   3763
MSG22           3891#   3893
MSG23           3952#   3954
```

```
MSG24     3985#   3987
MSG25     4046#   4048
MSG26     4102#   4104
MSG27     4125#   4127
MSG3      2676#   2678
MSG30     4144#   4146
MSG31     4181#   4183
MSG32     4221#   4223
MSG33     4255#   4257
MSG34     4322#   4324
MSG35     4367#   4369
MSG36     4391#   4393
MSG37     4429#   4431
MSG4      2701#   2703
MSG40     4238#   4240
MSG41     4470#   4472
MSG42     4508#   4510
MSG43     4576#   4578
MSG44     4613#   4615
MSG45     4648#   4650
MSG46     4692#   4694
MSG47     4717#   4719
MSG5      2726#   2728
MSG50     4750#   4752
MSG51     4670#   4672
MSG53     4790#   4792
MSG54     4816#   4818
MSG55     4915#   4917
MSG56     4934#   4936
MSG57     4961#   4963
MSG6      2756#   2758
MSG60     5037#   5039
MSG61     5070#   5072
MSG62     5102#   5104
MSG63     5150#   5152
MSG64     5189#   5191
MSG65     5229#   5231
MSG66     5270#   5272
MSG67     5307#   5309
MSG7      2773#   2775
MSG70     5358#   5360
MSG71     5624#   5626
MSG72     5394#   5396
MSG73     5432#   5434
MSG74     5471#   5473
MSG75     5507#   5509
MSG76     5536#   5538
MSG77     5666#   5668
MSG77A    5554#   5556
MSG77B    5573#   5575
MSG77C    5589#   5591
MSG8      2797#   2799
MSG9      2820#   2822
MULT         1#   2427#
NEWMAC    2646#   2647
NEWTST       1#   2427#  2629  2651  2676  2701  2726  2756  2773  2797  2820  2843  2859  3152  3257
```

```
                3283    3309    3335    3361    3387    3413    3520    3570    3627    3709    3761    3891    3952    3985    4046
                4102    4125    4144    4181    4221    4238    4255    4322    4367    4391    4429    4470    4508    4576    4613
                4648    4670    4692    4717    4750    4790    4816    4915    4934    4961    5037    5070    5102    5150    5189
                5229    5270    5307    5358    5394    5432    5471    5507    5536    5554    5573    5589    5624    5666    5748
POP                1#  2427#   6062
PUSH               1#  2427#   6021
SAVEAD          2011#
SCOPE           2046#
SETTRA          6092#   6102    6103    6104    6105
SETUP              1#  2427#
SKIP               1#  2427#   2645    2671    2697    2722    2747    2769    2792    2816    2839    2855    3223    3254    3280
                3306    3332    3358    3384    3410    3565    3623    3706    3748    3888    3982    4035    4086    4122    4141
                4165    4217    4235    4317    4364    4387    4412    4464    4496    4544    4609    4636    4667    4714    4747
                4769    4808    4904    4931    4957    5034    5056    5088    5138    5168    5211    5250    5291    5346    5378
                5412    5455    5491    5524    5550    5570    5585    5605    5654    5703    5775
SLASH              1#  2427#
SPACE           2427#
STARS              1#  2427#   2440    2466    2514    2531    2622    2629    2638    2651    2660    2676    2685    2701    2710
                2726    2735    2751    2756    2764    2773    2781    2797    2805    2820    2828    2843    2850    2859    2868
                3152    3169    3257    3262    3283    3288    3309    3314    3335    3340    3361    3366    3387    3392    3413
                3419    3509    3520    3529    3547    3551    3570    3584    3627    3654    3709    3722    3761    3777    3792
                3805    3821    3837    3853    3864    3873    3882    3891    3908    3952    3966    3985    4004    4009    4013
                4046    4078    4102    4113    4125    4132    4144    4155    4174    4178    4181    4201    4221    4228    4238
                4247    4255    4278    4322    4338    4367    4378    4391    4403    4423    4426    4429    4446    4470    4486
                4508    4528    4576    4589    4613    4625    4648    4657    4670    4680    4692    4705    4717    4728    4750
                4760    4774    4778    4783    4787    4790    4799    4816    4836    4909    4912    4915    4921    4934    4943
                4961    4990    5037    5047    5062    5066    5070    5079    5095    5098    5102    5113    5144    5147    5150
                5160    5179    5186    5189    5199    5221    5226    5229    5240    5261    5266    5270    5280    5301    5304
                5307    5322    5351    5355    5358    5368    5387    5391    5394    5403    5423    5429    5432    5443    5463
                5467    5471    5481    5501    5504    5507    5514    5529    5533    5536    5542    5554    5560    5573    5577
                5589    5599    5624    5638    5661    5665    5666    5674    5748    5762    5819    5821    5857    5930    6008
                6076
TRMTRP          6092#
TYPBIN             1#  2427#
TYPDEC             1#  2427#
TYPNAM             1#  2427#
TYPNUM             1#  2427#
TYPOCS             1#  2427#
TYPOCT             1#  2427#   5812
TYPTXT             1#  2427#   5807    5814
$IUT            2011#   3542    3592    3659    3726    3971    4015    4084    4120    4139    4161    4207    4232    4251    4286
                4304    4345    4385    4410    4460    4494    4541    4597    4634    4684    4712    4737    4766    4805    4872
                4929    4950    5002    5053    5085    5119    5165    5209    5247    5288    5329    5375    5409    5452    5488
                5522    5548    5568    5645
$SAVEA          2011#
$SYNC           2011#   3540    3590    3657    3725    3912    3970    4014    4083    4119    4137    4160    4206    4231    4250
                4285    4303    4344    4384    4409    4459    4493    4540    4596    4633    4683    4711    4736    4765    4804
                4870    4928    4949    5001    5052    5084    5118    5164    5208    5246    5287    5327    5374    5408    5451
                5487    5521    5547    5567    5644
$$CMRE          2466#   2501    2502    2503
$$CMTM          2466#   2504    2505    2506    2507
$$ESCA             1#  2427#
$$NEWT             1#  2427#   2629    2651    2676    2701    2726    2756    2773    2797    2820    2843    2859    3152    3257
                3283    3309    3335    3361    3387    3413    3520    3570    3627    3709    3761    3891    3952    3985    4046
                4102    4125    4144    4181    4221    4238    4255    4322    4367    4391    4429    4470    4508    4576    4613
                4648    4670    4692    4717    4750    4790    4816    4915    4934    4961    5037    5070    5102    5150    5189
```

```
            5229    5270    5307    5358    5394    5432    5471    5507    5536    5554    5573    5589    5624    5666    5748
$$SET       6092#   6102    6103    6104    6105
$$SKIP         1#   2427#   2645    2671    2697    2722    2747    2769    2792    2816    2839    2855    3254    3280    3306
            3332    3358    3384    3410    3566    3623    3706    3748    3888    3982    4035    4087    4122    4141    4165
            4217    4235    4317    4364    4387    4412    4464    4496    4544    4609    4636    4667    4714    4747    4769
            4808    4904    4931    4957    5034    5056    5088    5138    5168    5211    5250    5291    5346    5378    5412
            5455    5491    5524    5550    5570    5585    5605    5654    5703
.EQUAT         1#
.HEADE         1#   2011#   2026
.KT11          1#
.SETUP         1#   2011#   2428    5820
.SWRHI         1#   2011#
.SWRLO      2011#
.$ACTI         1#   2011#   2440
.$CATC         1#   2011#   2428
.$CMTA         1#   2011#   2466
.$DB2D         1#
.$DB20         1#
.$DIV          1#
.$EOP          1#   2011#   5821
.$ERRO         1#
.$ERRT         1#
.$MULT         1#
.$POWE         1#
.$RAND         1#
.$RDDE         1#
.$RDOC         1#
.$READ         1#
.$SAVE         1#
.$SB2D         1#
.$SB20         1#
.$SCOP         1#
.$SIZE         1#
.$SUPR         1#
.$TRAP         1#   5820#   6076
.$TYPB         1#
.$TYPD         1#   5820#   6008
.$TYPE         1#   2011#   5820#   5857
.$TYPO         1#   5820#   5930
.1170          1#   2011#   2037
```

```
ADC     2906  2917  3857
ADD     3117  3543  3556  3781  3787  3796  3808  3815  3824  3831  3840  3847  3867  3876  3885
        4218  4542  4546  4561  5889  5959  5969  6041
ASL     2955
ASLB    6046
ASR     2982
ASRB    5003
BCC     2645  2670  2910  2959  3012  3025  3061  3121  3134  3146  6047
BCS     2641  2671  2689  2714  2872  2884  2899  2924  2936  2947  2971  2986  2998  3046  3081
        3094  3106  3558  3600  3706  3797  3858
BEQ     2768  2785  2816  2832  2853  2875  2887  2897  2908  2934  2945  2984  2996  3010  3033
        3059  3104  3119  3173  3185  3192  3204  3211  3223  3231  3235  3239  3243  3250  3254
        3266  3270  3276  3280  3292  3296  3302  3306  3318  3322  3328  3332  3344  3348  3354
        3358  3370  3374  3380  3384  3396  3400  3406  3410  3445  3449  3454  3458  3463  3467
        3472  3476  3481  3485  3490  3494  3598  3605  3607  3623  3672  3691  3693  3737  3748
        3783  3789  3802  3811  3818  3827  3834  3843  3850  3861  3870  3879  3888  4018  4023
        4092  4096  4122  4141  4165  4217  4235  4307  4317  4348  4359  4364  4599  4602  4747
        4769  4808  4931  5023  5034  5056  5088  5122  5168  5211  5338  5342  5346  5378  5412
        5524  5550  5570  5585  5605  5654  5692  5696  5703  5775  5797  5805  5845  5892  5925
        5986
BGE     2810  2838
BGT     2769  2791  5839  5993  6055
BHI     2721  2741
BIC     3090  4346  4363  4806  4875  5029  5030  5652  5773  5836  5983
BIS     3102  3800  3809  3816  3825  3832  3841  3848  3868  3877  3886  4635  5031  5988  5989
        6049  6050
BIT     3077  3690  3692  4121  4903  5022
BITB    4495  4498  5913
BLE     2766  2792
BLOS    2644  2722  2739
BLT     2767  2809  2839  2854  5904  5994  6038  6054
BMI     2642  2663  2690  2715  2747  2874  2886  2907  2921  2944  2956  2968  3022  3043  3078
        3091  3103  3131  3143  3662  3663  4087  4387  4412  4496  4499  4562  4636  4667  4714
        4739  4743  4882  4957  5250  5291  5455  5491  6045
BNE     2786  2815  2833  2855  2922  2957  2969  3023  3044  3066  3079  3092  3132  3144  3177
        3196  3215  3566  3609  3616  3667  3683  3687  3755  3921  3924  3937  3982  4031  4035
        4040  4051  4054  4163  4210  4289  4315  4350  4414  4555  4609  4638  4642  4810  4857
        4887  4891  4904  4953  5007  5015  5019  5124  5130  5138  5170  5213  5252  5293  5380
        5414  5493  5609  5616  5795  5799  5886  5894  5900  5914  5921  5984  6043
BPL     2665  2691  2746  2896  2933  2983  2995  3009  3058  3065  3118  3597  3947  4464  4865
        4878  4897  4902  5331  5880  5918  5982  6029  6059
BR      3594  3728  3729  3915  3916  4282  4283  4543  4544  4552  5801  5808  5815  5882  5897
        5907  5916  5923  5960  5975  5996  6040  6057
BVC     2696  2716  2923  2970  3120
BVS     2643  2664  2697  2740  2873  2885  2898  2909  2935  2946  2958  2985  2997  3011  3024
        3045  3060  3080  3093  3105  3133  3145  3599
CCC     2639  2686  2711  2736  2765  2782  2806  2829  2893  2918  3019  3589  5701
CLC     2981  3704
CLN     2931  2981  2993  3007  3056  3062  3116  4560
CLR     2871  2880  2892  2904  2915  2929  2941  2952  2964  2976  2991  3003  3017  3029  3038
        3051  3074  3086  3099  3111  3126  3139  3170  3174  3189  3193  3208  3212  3226  3227
        3228  3246  3247  3263  3264  3289  3290  3315  3316  3341  3342  3367  3368  3393  3394
        3403  3404  3420  3497  3510  3531  3546  3586  3587  3602  3656  3669  3724  4115  4134
        4157  4203  4284  4301  4310  4311  4341  4408  4457  4537  4591  4630  4632  4733  4763
        4849  4850  4851  4895  4947  4997  5081  5115  5245  5286  5373  5405  5448  5486  5613
        5800  5818  5973  6032  6035
```

DEKBAB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
CLRB    5206  5922  6061
CLV     3116
CLZ     2870  2882  2954  3041  3076  3089  3129  3141  3658  4138
CMP     3142  3265  3269  3275  3279  3291  3295  3301  3305  3317  3321  3327  3331  3343  3347
        3353  3357  3369  3373  3379  3383  3395  3399  3405  3409  3444  3448  3453  3457  3462
        3466  3471  3475  3480  3484  3489  3493  3604  3608  3615  3666  3692  3686  3736  3747
        3754  3920  3923  3936  4017  4022  4030  4039  4053  4095  4140  4164  4209  4234  4347
        4413  4637  4641  4746  4886  4890  4952  5014  5018  5033  5055  5121  5123  5129  5167
        5169  5212  5251  5292  5337  5341  5345  5377  5379  5411  5413  5492  5569  5604  5608
        5615  5653  5691  5695  5702  5774  5798  5804  6053
CMPB    4598  4601  4608  4930  5210  5523  5549  5584  5891  5893  5899  5920  5924
COM     2895  3273  3274  3299  3300  3325  3326  3351  3352  3377  3378  3782  3788  3801  3810
        3817  3826  3833  3842  3849  3869  3878  3887  5082
DEC     2932  3018  3039  3603  3753  4448  4449  4452  4453  4535  4536  5207  5837
DECB    5903  5906  5981  5992
EMT     2045  5688
HALT    2435  2647  2667  2673  2693  2699  2718  2724  2743  2749  2771  2788  2794  2812  2818
        2835  2841  2857  2877  2889  2901  2912  2926  2938  2949  2961  2973  2988  3000  3014
        3027  3035  3048  3068  3071  3083  3096  3108  3123  3136  3148  3178  3181  3186  3197
        3200  3205  3216  3219  3224  3232  3236  3240  3244  3251  3255  3267  3271  3277  3281
        3293  3297  3303  3307  3319  3323  3329  3333  3345  3349  3355  3359  3371  3375  3381
        3385  3397  3401  3407  3411  3446  3450  3455  3459  3464  3468  3473  3477  3482  3486
        3491  3495  3559  3562  3567  3595  3610  3613  3617  3620  3624  3673  3677  3684  3688
        3695  3698  3702  3707  3738  3741  3756  3759  3784  3790  3798  3803  3812  3819  3828
        3835  3844  3851  3859  3862  3871  3880  3889  3925  3928  3932  3940  3948  3974  3983
        4024  4027  4036  4041  4044  4055  4058  4062  4093  4097  4100  4123  4142  4167  4170
        4211  4214  4219  4236  4290  4293  4297  4308  4319  4351  4355  4360  4365  4388  4416
        4419  4465  4468  4500  4505  4553  4557  4563  4568  4571  4603  4606  4610  4639  4643
        4646  4664  4668  4687  4715  4740  4744  4748  4770  4811  4814  4879  4883  4888  4892
        4898  4905  4932  4954  4958  5008  5016  5020  5024  5035  5057  5089  5125  5131  5134
        5139  5171  5175  5214  5217  5253  5257  5294  5297  5332  5335  5339  5343  5347  5381
        5384  5415  5419  5456  5459  5494  5497  5525  5551  5571  5586  5610  5617  5622  5649
        5655  5689  5693  5697  5704  5709  5714  5718  5722  5726  5729  5732  5741  5745  5776
        5779  5783  5786  5790  5881
INC     2943  2979  3032  3087  3100  3127  3171  3175  3183  3190  3194  3202  3209  3213  3221
        3229  3248  3421  3424  3427  3430  3433  3436  3439  3442  3498  3502  3511  3515  3517
        3530  3532  3535  3538  3552  3553  3585  3588  3601  3655  3723  3734  3735  3746  3778
        3909  3967  4005  4032  4079  4091  4114  4133  4156  4202  4229  4248  4279  4302  4312
        4339  4342  4379  4383  4404  4447  4487  4491  4529  4590  4595  4626  4658  4681  4706
        4710  4729  4761  4768  4800  4837  4922  4927  4944  4991  4996  5048  5080  5114  5116
        5161  5200  5241  5281  5323  5369  5404  5444  5482  5515  5543  5561  5578  5600  5639
        5675  5763  5802  5835  5987  5995  6039
INCB    5926
IOT     2046  5772
JMP     2439  4208  4233  4252  4685  5852
JSR     5330  5847  5898  5905  5912
MOV     2628  3176  3184  3195  3203  3214  3222  3234  3242  3253  3443  3452  3461  3470  3479
        3488  3518  3545  3548  3549  3555  3660  3665  3670  3727  3731  3732  3733  3743  3744
        3750  3751  3780  3786  3794  3795  3807  3814  3823  3830  3839  3846  3855  3866  3875
        3884  3910  3914  3918  3919  3922  3934  3935  3938  3939  3943  3945  3946  3950  3951
        3968  3972  3976  3977  3978  3980  4006  4007  4008  4011  4012  4016  4019  4020  4021
        4029  4034  4038  4052  4064  4080  4081  4082  4089  4090  4116  4117  4118  4135  4136
        4158  4159  4166  4169  4204  4205  4230  4249  4280  4281  4340  4343  4357  4362  4380
        4381  4382  4405  4406  4407  4415  4450  4451  4454  4455  4456  4458  4461  4462  4488
        4489  4490  4492  4497  4530  4531  4532  4533  4534  4538  4539  4545  4556  4559  4592
        4594  4600  4627  4628  4629  4631  4659  4660  4661  4662  4682  4707  4708  4709  4730
```

```
                4731    4732    4734    4735    4738    4762    4801    4802    4803    4838    4839    4840    4841    4842    4843
                4844    4845    4846    4847    4848    4852    4853    4854    4855    4858    4859    4860    4866    4868    4873
                4874    4885    4923    4925    4926    4945    4946    4948    4951    4992    4993    4994    4995    4998    4999
                5004    5005    5010    5013    5026    5027    5049    5083    5117    5162    5163    5166    5201    5202    5203
                5204    5242    5243    5244    5248    5282    5283    5284    5285    5289    5324    5325    5326    5370    5371
                5372    5376    5406    5407    5410    5445    5446    5447    5449    5450    5453    5483    5484    5485    5489
                5516    5517    5518    5519    5520    5544    5545    5546    5562    5563    5564    5565    5566    5579    5580
                5581    5582    5583    5601    5602    5606    5640    5641    5642    5643    5651    5676    5677    5678    5679
                5680    5681    5682    5683    5684    5685    5686    5736    5737    5738    5747    5764    5765    5766    5767
                5768    5769    5770    5771    5793    5803    5806    5811    5812    5834    5840    5844    5883    5884    5888
                5901    5956    5964    5965    5966    5972    5979    5997    5998    5999    6000    6001    6022    6023    6024
                6025    6026    6027    6028    6033    6036    6056    6062    6063    6064    6065    6066    6068    6069    6085
                6086    6089
MOVB            3057    3745    3752    4924    5050    5051    5205    5885    5911    5919    5957    5958    5961    5962    5963
                5967    5970    5971    5990    6031    6034    6048    6051    6060    6088
NEG             3593    5120    5137    5968    6030
NOP             3554    3725    3912    3970    4014    4083    4119    4160    4206    4231    4250    4253    4254    4285    4303
                4344    4384    4409    4459    4493    4540    4596    4633    4683    4711    4736    4765    4804    4928    4949
                5052    5084    5118    5164    5208    5246    5287    5374    5408    5451    5487    5521    5547    5567    5644
                5647    5848    5849    5850
RESET           5846
ROL             2967    3422    3423    3425    3426    3428    3429    3431    3432    3434    3435    3437    3438    3440    3441
                3499    3500    3501    3503    3504    3505    3506    3507    3508    3512    3513    3514    3516    3533    3534
                3536    3537    3539    3705    5974    5976    5977    5978    5980
ROLB            4713
ROR             2920    2966    2978    3005    3053    3113    3114    3557    5028
RORB            5012    5054
RTI             5646    5890    6002    6070
RTS             4663    5928    6090
RTT             3550
SBC             2994
SCC             2869    2881    2905    2930    2942    2953    2980    2992    3006    3040    3055    3075    3088    3101    3115
                3128    3140    5328    5603    5607    5614    5687
SEC             2661    2916    2965    2977    3004    3052    3112    3856    4547    4871
SEN             2738    2831    2919    3030    4764
SEV             2688    2784    2851    2894    3020    3591
SEZ             2713    2808    2894    3063    3541    3591    3979    4033    5000    5136
SOB             4287    4305    4313
SUB             3130    6037
SWAB            3008    3054    3681    4593    5011
SXT             3021    3031    4767
TRAP            5739    6092    6102    6103    6104    6105
TST             2883    3064    3172    3191    3210    3230    3238    3249    3606    3622    3671    4085    4162    4216    4288
                4306    4314    4316    4349    4358    4411    4463    4666    4742    4807    4809    4856    4877    4881    4896
                4956    5006    5087    5249    5290    5454    5490    5794    5796    5887    5895    5915    5985    6042    6052
                6087
TSTB            4386    4862    4900    5879    5917    6044    6058
XOR             3042    5086
.ASCII          2508    2509    5853
.ASCIZ          2510    5810    5817
.BLKW           6075
.BYTE           2477    2478    2483    2484    2495    2496    2497    2498    5855    6003    6004    6005    6006
.DSABL          2011
.ENABL             1    2011
.END            6107
.ENDC           2031    2037    2426    2428    2440    2441    2463    2465    2467    2474    2499    2504    2508    2514    2515
```

2532	2623	2630	2631	2638	2639	2640	2646	2650	2651	2652	2653	2660	2661	2662
2668	2669	2672	2675	2676	2677	2678	2685	2686	2687	2694	2695	2698	2700	2701
2702	2703	2710	2711	2712	2719	2720	2723	2725	2726	2727	2728	2735	2736	2737
2744	2745	2748	2750	2751	2752	2757	2758	2764	2765	2766	2770	2772	2773	2774
2775	2781	2782	2783	2789	2790	2793	2796	2797	2798	2799	2805	2806	2807	2813
2814	2817	2819	2820	2821	2822	2828	2829	2830	2836	2837	2840	2842	2843	2844
2845	2850	2851	2852	2856	2858	2859	2860	2861	2868	2869	2870	2878	2879	2890
2891	2902	2903	2913	2914	2927	2928	2939	2940	2950	2951	2962	2963	2974	2975
2989	2990	3001	3002	3015	3016	3028	3029	3036	3037	3049	3050	3069	3070	3072
3073	3084	3085	3097	3098	3109	3110	3124	3125	3137	3138	3149	3150	3153	3154
3169	3170	3171	3179	3180	3182	3183	3187	3188	3198	3199	3201	3202	3206	3207
3217	3218	3220	3221	3224	3225	3226	3233	3234	3237	3238	3241	3242	3245	3246
3252	3253	3255	3256	3257	3258	3259	3262	3263	3264	3268	3269	3272	3273	3278
3279	3281	3282	3283	3284	3285	3288	3289	3290	3294	3295	3298	3299	3304	3305
3307	3308	3309	3310	3311	3314	3315	3316	3320	3321	3324	3325	3330	3331	3333
3334	3335	3336	3337	3340	3341	3342	3346	3347	3350	3351	3356	3357	3359	3360
3361	3362	3363	3366	3367	3368	3372	3373	3376	3377	3382	3383	3385	3386	3387
3388	3389	3392	3393	3394	3398	3399	3402	3403	3408	3409	3411	3412	3413	3414
3415	3419	3420	3421	3447	3448	3451	3452	3456	3457	3460	3461	3465	3466	3469
3470	3474	3475	3478	3479	3483	3484	3487	3488	3492	3493	3496	3497	3510	3521
3522	3529	3530	3531	3541	3548	3552	3560	3561	3563	3564	3567	3568	3569	3571
3572	3584	3585	3586	3591	3596	3597	3611	3612	3614	3615	3618	3619	3621	3622
3624	3625	3626	3628	3629	3654	3655	3656	3658	3675	3676	3680	3681	3685	3686
3689	3690	3696	3697	3700	3701	3703	3704	3707	3708	3709	3710	3711	3722	3723
3724	3725	3726	3739	3740	3742	3743	3749	3757	3758	3760	3761	3762	3763	3777
3778	3779	3785	3786	3791	3792	3793	3799	3800	3804	3805	3806	3813	3814	3820
3821	3822	3829	3830	3836	3837	3838	3845	3846	3852	3853	3854	3860	3861	3863
3864	3865	3872	3873	3874	3881	3882	3883	3889	3890	3891	3892	3893	3908	3909
3910	3912	3913	3926	3927	3930	3931	3933	3934	3942	3943	3949	3950	3953	3954
3966	3967	3968	3970	3971	3975	3976	3983	3984	3985	3986	3987	4004	4005	4006
4010	4014	4015	4025	4026	4028	4029	4036	4037	4038	4042	4043	4045	4046	4047
4048	4056	4057	4060	4061	4063	4064	4078	4079	4080	4083	4084	4088	4094	4095
4098	4099	4101	4102	4103	4104	4113	4114	4115	4119	4120	4123	4124	4125	4126
4127	4132	4133	4134	4138	4142	4143	4144	4145	4146	4155	4156	4157	4160	4161
4166	4168	4169	4171	4172	4175	4179	4182	4183	4201	4202	4203	4206	4207	4212
4213	4215	4216	4218	4220	4221	4222	4223	4228	4229	4230	4231	4232	4236	4237
4238	4239	4240	4247	4248	4249	4250	4251	4256	4257	4278	4279	4280	4285	4286
4291	4292	4295	4296	4300	4301	4303	4304	4309	4310	4318	4320	4321	4323	4324
4338	4339	4340	4344	4345	4353	4354	4356	4357	4361	4362	4365	4366	4367	4368
4369	4378	4379	4380	4384	4385	4388	4390	4391	4392	4393	4403	4404	4405	4409
4410	4413	4417	4418	4420	4421	4424	4427	4430	4431	4446	4447	4448	4459	4460
4465	4466	4467	4469	4470	4471	4472	4486	4487	4488	4493	4494	4497	4503	4504
4507	4508	4509	4510	4528	4529	4530	4540	4541	4545	4554	4555	4558	4559	4566
4567	4569	4570	4575	4576	4577	4578	4589	4590	4591	4596	4597	4604	4605	4607
4608	4610	4611	4612	4614	4615	4625	4626	4627	4633	4634	4637	4640	4641	4644
4645	4647	4648	4649	4650	4657	4658	4659	4665	4666	4668	4669	4670	4671	4672
4680	4681	4682	4683	4684	4690	4691	4693	4694	4705	4706	4707	4711	4712	4715
4716	4717	4718	4719	4728	4729	4730	4736	4737	4741	4742	4745	4746	4748	4749
4750	4751	4752	4760	4761	4762	4765	4766	4770	4771	4772	4775	4779	4784	4788
4791	4792	4799	4800	4801	4804	4805	4809	4812	4813	4815	4816	4817	4818	4836
4837	4838	4871	4880	4881	4884	4885	4889	4890	4894	4895	4899	4900	4905	4906
4907	4910	4913	4916	4917	4921	4922	4923	4928	4929	4932	4933	4934	4935	4936
4943	4944	4945	4949	4950	4955	4956	4958	4959	4960	4962	4963	4990	4991	4992
5002	5009	5010	5017	5018	5021	5022	5025	5026	5035	5036	5037	5038	5039	5047
5048	5049	5052	5053	5057	5058	5059	5063	5067	5071	5072	5079	5080	5081	5084
5085	5089	5090	5091	5096	5099	5103	5104	5113	5114	5115	5118	5119	5128	5129

```
                5132    5133    5135    5136    5139    5140    5141    5145    5148    5151    5152    5160    5161    5162    5164
                5165    5169    5173    5174    5176    5177    5180    5187    5190    5191    5199    5200    5201    5208    5209
                5212    5215    5216    5218    5219    5222    5227    5230    5231    5240    5241    5242    5246    5247    5251
                5255    5256    5258    5259    5262    5267    5271    5272    5280    5281    5282    5287    5288    5292    5295
                5296    5298    5299    5302    5305    5308    5309    5322    5323    5324    5328    5333    5334    5336    5337
                5340    5341    5344    5345    5347    5348    5349    5352    5356    5359    5360    5368    5369    5370    5374
                5375    5379    5382    5383    5385    5386    5388    5392    5395    5396    5403    5404    5405    5408    5409
                5413    5417    5418    5420    5421    5424    5430    5433    5434    5443    5444    5445    5451    5452    5456
                5457    5458    5460    5461    5464    5468    5472    5473    5481    5482    5483    5487    5488    5492    5495
                5496    5498    5499    5502    5505    5508    5509    5514    5515    5516    5521    5522    5525    5526    5527
                5530    5534    5537    5538    5542    5543    5544    5547    5548    5551    5552    5553    5555    5556    5560
                5561    5562    5567    5568    5571    5572    5573    5574    5575    5577    5578    5579    5586    5587    5588
                5590    5591    5599    5600    5601    5606    5612    5613    5620    5621    5623    5624    5625    5626    5638
                5639    5640    5644    5645    5650    5651    5655    5658    5659    5662    5666    5667    5668    5674    5675
                5676    5690    5691    5694    5695    5700    5701    5704    5707    5708    5712    5713    5716    5717    5720
                5721    5724    5725    5727    5728    5730    5731    5733    5734    5743    5744    5746    5747    5749    5750
                5762    5763    5764    5776    5777    5778    5781    5782    5784    5785    5788    5789    5792    5793    5810
                5817    5820    5822    5825    5827    5828    5830    5832    5835    5839    5842    5843    5844    5846    5852
                5853    5855    5856    5858    5931    6009    6077    6086    6089    6091    6101    6102    6103    6104    6105
                6106
.EQUIV          2045    2046    2048    2069    2070    2071    2072    2073    2074    2075    2076    2077    2078    2079    2108
                2109    2110    2111    2112    2113    2114    2115    2116    2117    2136    2137    2138    2139    2140    2141
                2142    2143    2144    2145    2197    2198    2199    2200    2408    2409    2410    2411    2412    2413    2414
                2415    2416    2417    2418    2419    2420    2421    2422    2423
.EVEN           5810    5817    5856
.IF             2027    2037    2426    2428    2435    2440    2462    2464    2466    2473    2499    2504    2508    2511    2514
                2531    2622    2629    2631    2638    2640    2645    2647    2650    2651    2653    2660    2662    2667    2668
                2671    2673    2675    2676    2678    2685    2687    2693    2694    2697    2699    2700    2701    2703    2710
                2712    2718    2719    2722    2724    2725    2726    2728    2735    2737    2743    2744    2747    2749    2750
                2751    2756    2758    2764    2766    2769    2771    2772    2773    2775    2781    2783    2788    2789    2792
                2794    2796    2797    2799    2805    2807    2812    2813    2816    2818    2819    2820    2822    2828    2830
                2835    2836    2839    2841    2842    2843    2845    2850    2852    2855    2857    2858    2859    2861    2868
                2870    2877    2878    2889    2890    2901    2902    2912    2913    2926    2927    2938    2939    2949    2950
                2961    2962    2973    2974    2988    2989    3000    3001    3014    3015    3027    3028    3035    3036    3048
                3049    3068    3069    3071    3072    3083    3084    3096    3097    3108    3109    3123    3124    3136    3137
                3148    3149    3152    3154    3169    3171    3178    3179    3181    3182    3186    3187    3197    3198    3200
                3201    3205    3206    3216    3217    3219    3220    3223    3224    3225    3232    3233    3236    3237    3240
                3241    3244    3245    3251    3252    3254    3255    3256    3257    3259    3262    3264    3267    3268    3271
                3272    3277    3278    3280    3281    3282    3283    3285    3288    3290    3293    3294    3297    3298    3303
                3304    3306    3307    3308    3309    3311    3314    3316    3319    3320    3323    3324    3329    3330    3332
                3333    3334    3335    3337    3340    3342    3345    3346    3349    3350    3355    3356    3358    3359    3360
                3361    3363    3366    3368    3371    3372    3375    3376    3381    3382    3384    3385    3386    3387    3389
                3392    3394    3397    3398    3401    3402    3407    3408    3410    3411    3412    3413    3415    3419    3421
                3446    3447    3450    3451    3455    3456    3459    3460    3464    3465    3468    3469    3473    3474    3477
                3478    3482    3483    3486    3487    3491    3492    3495    3496    3509    3520    3522    3529    3531    3540
                3541    3547    3551    3559    3560    3562    3563    3566    3567    3568    3570    3572    3584    3586    3590
                3591    3595    3596    3610    3611    3613    3614    3617    3618    3620    3621    3623    3624    3625    3627
                3629    3654    3656    3657    3658    3673    3675    3677    3680    3684    3685    3688    3689    3695    3696
                3698    3700    3702    3703    3706    3707    3708    3709    3711    3722    3724    3725    3738    3739    3741
                3742    3748    3756    3757    3759    3760    3761    3763    3777    3779    3784    3785    3790    3791    3792
                3798    3799    3803    3804    3805    3812    3813    3819    3820    3821    3828    3829    3835    3836    3837
                3844    3845    3851    3852    3853    3859    3860    3862    3863    3864    3871    3872    3873    3880    3881
                3882    3888    3889    3890    3891    3893    3908    3910    3912    3925    3926    3928    3930    3932    3933
                3940    3942    3948    3949    3952    3954    3966    3968    3970    3974    3975    3982    3983    3984    3985
                3987    4004    4006    4009    4013    4014    4024    4025    4027    4028    4035    4036    4037    4041    4042
                4044    4045    4046    4048    4055    4056    4058    4060    4062    4063    4078    4080    4083    4087    4093
```

4094	4097	4098	4100	4101	4102	4104	4113	4115	4119	4122	4123	4124	4125	4127
4132	4134	4137	4138	4141	4142	4143	4144	4146	4155	4157	4160	4165	4167	4168
4170	4171	4174	4178	4181	4183	4201	4203	4206	4211	4212	4214	4215	4217	4219
4220	4221	4223	4228	4230	4231	4235	4236	4237	4238	4240	4247	4249	4250	4255
4257	4278	4280	4285	4290	4291	4293	4295	4297	4300	4303	4308	4309	4317	4319
4320	4322	4324	4338	4340	4344	4351	4353	4355	4356	4360	4361	4364	4365	4366
4367	4369	4378	4380	4384	4387	4388	4390	4391	4393	4403	4405	4409	4412	4416
4417	4419	4420	4423	4426	4429	4431	4446	4448	4459	4464	4465	4466	4468	4469
4470	4472	4486	4488	4493	4496	4500	4503	4505	4507	4508	4510	4528	4530	4540
4544	4553	4554	4557	4558	4563	4566	4568	4569	4571	4575	4576	4578	4589	4591
4596	4603	4604	4606	4607	4609	4610	4611	4613	4615	4625	4627	4633	4636	4639
4640	4643	4644	4646	4647	4648	4650	4657	4659	4664	4665	4667	4668	4669	4670
4672	4680	4682	4683	4687	4690	4692	4694	4705	4707	4711	4714	4715	4716	4717
4719	4728	4730	4736	4740	4741	4744	4745	4747	4748	4749	4750	4752	4760	4762
4765	4769	4770	4771	4774	4778	4783	4787	4790	4792	4799	4801	4804	4808	4811
4812	4814	4815	4816	4818	4836	4838	4870	4871	4879	4880	4883	4884	4888	4889
4892	4894	4898	4899	4904	4905	4906	4909	4912	4915	4917	4921	4923	4928	4931
4932	4933	4934	4936	4943	4945	4949	4954	4955	4957	4958	4959	4961	4963	4990
4992	5001	5002	5008	5009	5016	5017	5020	5021	5024	5025	5034	5035	5036	5037
5039	5047	5049	5052	5056	5057	5058	5062	5066	5070	5072	5079	5081	5084	5088
5089	5090	5095	5098	5102	5104	5113	5115	5118	5125	5128	5131	5132	5134	5135
5138	5139	5140	5144	5147	5150	5152	5160	5162	5164	5168	5171	5173	5175	5176
5179	5186	5189	5191	5199	5201	5208	5211	5214	5215	5217	5218	5221	5226	5229
5231	5240	5242	5246	5250	5253	5255	5257	5258	5261	5266	5270	5272	5280	5282
5287	5291	5294	5295	5297	5298	5301	5304	5307	5309	5322	5324	5327	5328	5332
5333	5335	5336	5339	5340	5343	5344	5346	5347	5348	5351	5355	5358	5360	5368
5370	5374	5378	5381	5382	5384	5385	5387	5391	5394	5396	5403	5405	5408	5412
5415	5417	5419	5420	5423	5429	5432	5434	5443	5445	5451	5455	5456	5457	5459
5460	5463	5467	5471	5473	5481	5483	5487	5491	5494	5495	5497	5498	5501	5504
5507	5509	5514	5516	5521	5524	5525	5526	5529	5533	5536	5538	5542	5544	5547
5550	5551	5552	5554	5556	5560	5562	5567	5570	5571	5572	5573	5575	5577	5579
5585	5586	5587	5589	5591	5599	5601	5605	5610	5612	5617	5620	5622	5623	5624
5626	5638	5640	5644	5649	5650	5654	5655	5658	5661	5665	5666	5668	5674	5676
5689	5690	5693	5694	5697	5700	5703	5704	5707	5709	5712	5714	5716	5718	5720
5722	5724	5726	5727	5729	5730	5732	5733	5741	5743	5745	5746	5748	5750	5762
5764	5775	5776	5777	5779	5781	5783	5784	5786	5788	5790	5792	5809	5816	5819
5820	5821	5825	5826	5827	5828	5829	5830	5834	5838	5841	5843	5844	5846	5852
5853	5855	5857	5930	6008	6076	6085	6089	6091	6092	6102	6103	6104	6105	6106
.IFF 2441	2463	2465	2467	2473	2499	2515	2532	2623	2629	2630	2631	2639	2640	2646
2647	2650	2651	2652	2653	2661	2662	2667	2668	2672	2673	2675	2676	2677	2678
2686	2687	2693	2694	2698	2700	2701	2702	2703	2711	2712	2718	2719	2723	2725
2726	2727	2728	2736	2737	2743	2744	2748	2750	2751	2752	2756	2757	2758	2765
2766	2770	2771	2772	2773	2774	2775	2782	2783	2788	2789	2793	2794	2796	2797
2798	2799	2806	2807	2812	2813	2817	2819	2820	2821	2822	2829	2830	2835	2836
2840	2842	2843	2844	2845	2851	2852	2856	2857	2858	2859	2860	2861	2869	2870
2877	2878	2890	2891	2902	2903	2913	2914	2927	2928	2939	2940	2950	2951	2962
2963	2974	2975	2989	2990	3001	3002	3015	3016	3028	3029	3036	3037	3049	3050
3069	3070	3072	3073	3084	3085	3097	3098	3109	3110	3124	3125	3137	3138	3149
3150	3152	3153	3154	3170	3171	3179	3180	3182	3183	3186	3187	3198	3199	3201
3202	3206	3207	3217	3218	3220	3221	3223	3225	3226	3233	3234	3237	3238	3241
3242	3245	3246	3252	3253	3255	3256	3257	3258	3259	3263	3264	3267	3268	3271
3272	3277	3278	3281	3282	3283	3284	3285	3289	3290	3293	3294	3297	3298	3303
3304	3307	3308	3309	3310	3311	3315	3316	3319	3320	3323	3324	3329	3330	3333
3334	3335	3336	3337	3341	3342	3345	3346	3349	3350	3355	3356	3359	3360	3361
3362	3363	3367	3368	3371	3372	3375	3376	3381	3382	3385	3386	3387	3388	3389
3393	3394	3397	3398	3401	3402	3407	3408	3411	3412	3413	3414	3415	3420	3421

3447	3448	3451	3452	3456	3457	3460	3461	3465	3466	3469	3470	3474	3475	3478
3479	3483	3484	3487	3488	3492	3493	3496	3497	3510	3520	3521	3522	3530	3531
3541	3548	3552	3559	3560	3561	3562	3563	3564	3567	3568	3569	3570	3571	3572
3585	3586	3591	3595	3596	3610	3611	3612	3613	3614	3615	3617	3618	3619	3620
3621	3622	3624	3625	3626	3627	3628	3629	3655	3656	3658	3673	3676	3677	3681
3684	3685	3686	3688	3689	3690	3695	3696	3697	3698	3701	3702	3703	3704	3707
3708	3709	3710	3711	3723	3724	3725	3726	3738	3739	3740	3741	3742	3743	3749
3756	3757	3758	3759	3760	3761	3762	3763	3778	3779	3784	3785	3786	3791	3792
3793	3799	3800	3804	3805	3806	3813	3814	3820	3821	3822	3829	3830	3836	3837
3838	3845	3846	3852	3853	3854	3860	3861	3863	3864	3865	3872	3873	3874	3881
3882	3883	3889	3890	3891	3892	3893	3909	3910	3912	3913	3925	3926	3927	3928
3931	3932	3933	3934	3940	3943	3948	3949	3950	3952	3953	3954	3967	3968	3970
3971	3974	3975	3976	3983	3984	3985	3986	3987	4005	4006	4010	4014	4015	4024
4025	4026	4027	4028	4029	4036	4037	4038	4041	4042	4043	4044	4045	4046	4047
4048	4055	4056	4057	4058	4061	4062	4063	4064	4079	4080	4083	4084	4088	4093
4094	4095	4097	4098	4099	4100	4101	4102	4103	4104	4114	4115	4119	4120	4123
4124	4125	4126	4127	4133	4134	4138	4142	4143	4144	4145	4146	4156	4157	4160
4161	4166	4167	4168	4169	4170	4171	4172	4175	4179	4181	4182	4183	4202	4203
4206	4207	4211	4212	4213	4214	4215	4216	4218	4219	4220	4221	4222	4223	4229
4230	4231	4232	4236	4237	4238	4239	4240	4248	4249	4250	4251	4255	4256	4257
4279	4280	4285	4286	4290	4291	4292	4293	4296	4297	4301	4303	4304	4309	4310
4318	4320	4321	4322	4323	4324	4339	4340	4344	4345	4351	4354	4355	4356	4357
4360	4361	4362	4365	4366	4367	4368	4369	4379	4380	4384	4385	4388	4391	4392
4393	4404	4405	4409	4410	4413	4416	4417	4418	4419	4420	4421	4424	4427	4429
4430	4431	4447	4448	4459	4460	4465	4466	4467	4468	4469	4470	4471	4472	4487
4488	4493	4494	4497	4500	4504	4505	4508	4509	4510	4529	4530	4540	4541	4545
4553	4554	4555	4557	4558	4559	4563	4567	4568	4569	4570	4571	4576	4577	4578
4590	4591	4596	4597	4603	4604	4605	4606	4607	4608	4610	4611	4612	4613	4614
4615	4626	4627	4633	4634	4637	4639	4640	4641	4643	4644	4645	4646	4647	4648
4649	4650	4658	4659	4664	4665	4666	4668	4669	4670	4671	4672	4681	4682	4683
4684	4687	4691	4692	4693	4694	4706	4707	4711	4712	4715	4716	4717	4718	4719
4729	4730	4736	4737	4740	4741	4742	4744	4745	4746	4748	4749	4750	4751	4752
4761	4762	4765	4766	4770	4771	4772	4775	4779	4784	4788	4790	4791	4792	4800
4801	4804	4805	4809	4811	4812	4813	4814	4815	4816	4817	4818	4837	4838	4871
4879	4880	4881	4883	4884	4885	4888	4889	4890	4892	4895	4898	4899	4900	4905
4906	4907	4910	4913	4915	4916	4917	4922	4923	4928	4929	4932	4933	4934	4935
4936	4944	4945	4949	4950	4954	4955	4956	4958	4959	4960	4961	4962	4963	4991
4992	5002	5008	5009	5010	5016	5017	5018	5020	5021	5022	5024	5025	5026	5035
5036	5037	5038	5039	5048	5049	5052	5053	5057	5058	5059	5063	5067	5070	5071
5072	5080	5081	5084	5085	5089	5090	5091	5096	5099	5102	5103	5104	5114	5115
5118	5119	5125	5129	5131	5132	5133	5134	5135	5136	5139	5140	5141	5145	5148
5150	5151	5152	5161	5162	5164	5165	5169	5171	5174	5175	5176	5177	5180	5187
5189	5190	5191	5200	5201	5208	5209	5212	5214	5215	5216	5217	5218	5219	5222
5227	5229	5230	5231	5241	5242	5246	5247	5251	5253	5256	5257	5258	5259	5262
5267	5270	5271	5272	5281	5282	5287	5288	5292	5294	5295	5296	5297	5298	5299
5302	5305	5307	5308	5309	5323	5324	5328	5332	5333	5334	5335	5336	5337	5339
5340	5341	5343	5344	5345	5347	5348	5349	5352	5356	5358	5359	5360	5369	5370
5374	5375	5379	5381	5382	5383	5384	5385	5386	5388	5392	5394	5395	5396	5404
5405	5408	5409	5413	5415	5418	5419	5420	5421	5424	5430	5432	5433	5434	5444
5445	5451	5456	5457	5458	5459	5460	5461	5464	5468	5471	5472	5473	5482	5483
5487	5492	5494	5495	5496	5497	5498	5499	5502	5505	5507	5508	5509	5515	5516
5521	5525	5526	5527	5530	5534	5536	5537	5538	5543	5544	5547	5551	5552	5553
5554	5555	5556	5561	5562	5567	5571	5572	5573	5574	5575	5578	5579	5586	5587
5588	5589	5590	5591	5600	5601	5606	5610	5613	5617	5621	5622	5623	5624	5625
5626	5639	5640	5644	5649	5650	5651	5655	5659	5662	5666	5667	5668	5675	5676
5689	5690	5691	5693	5694	5695	5697	5701	5704	5708	5709	5713	5714	5717	5718

```
                5721    5722    5725    5726    5728    5729    5731    5732    5734    5741    5744    5745    5746    5747    5748
                5749    5750    5763    5764    5775    5776    5777    5778    5779    5782    5783    5784    5785    5786    5789
                5790    5793    5820    5822    5829    5834    5838    5841    5853    5858    5931    6009    6077    6086
.IFT            5810    5817
.IFTF           5810    5817
.IIF            2026    2031    2036    2435    2511    5813    5827    5828    5835    5853    5856    5930    6101    6102    6103
                6104    6105
.IRP            2428    2511    2629    2647    2650    2651    2667    2668    2675    2676    2693    2694    2700    2701    2718
                2719    2725    2726    2743    2744    2750    2756    2771    2772    2773    2788    2789    2796    2797    2812
                2813    2819    2820    2835    2836    2842    2843    2857    2858    2859    2877    2878    3152    3179    3182
                3186    3187    3198    3201    3206    3217    3220    3225    3233    3237    3241    3245    3252    3256    3257
                3267    3268    3271    3272    3277    3278    3281    3282    3283    3293    3294    3297    3298    3303    3304
                3307    3308    3309    3319    3320    3323    3324    3329    3330    3333    3334    3335    3345    3346    3349
                3350    3355    3356    3359    3360    3361    3371    3372    3375    3376    3381    3382    3385    3386    3387
                3397    3398    3401    3402    3407    3408    3411    3412    3413    3447    3451    3456    3460    3465    3469
                3474    3478    3483    3487    3492    3496    3520    3559    3562    3567    3570    3595    3596    3610    3613
                3617    3620    3624    3627    3673    3677    3684    3688    3695    3698    3702    3707    3709    3738    3741
                3756    3759    3761    3784    3791    3799    3804    3813    3820    3829    3836    3845    3852    3860    3863
                3872    3881    3890    3891    3925    3928    3932    3940    3948    3952    3974    3983    3985    4024    4027
                4036    4041    4044    4046    4055    4058    4062    4093    4097    4100    4102    4123    4125    4142    4144
                4167    4170    4181    4211    4214    4219    4221    4236    4238    4255    4290    4293    4297    4320    4322
                4351    4355    4360    4365    4367    4388    4391    4416    4419    4429    4465    4468    4470    4500    4505
                4508    4553    4557    4563    4568    4571    4576    4603    4606    4610    4613    4639    4643    4646    4648
                4664    4668    4670    4687    4692    4715    4717    4740    4744    4748    4750    4770    4790    4811    4814
                4816    4879    4883    4888    4892    4898    4905    4915    4932    4934    4954    4958    4961    5008    5016
                5020    5024    5035    5037    5057    5070    5089    5102    5125    5131    5134    5139    5150    5171    5175
                5189    5214    5217    5229    5253    5257    5270    5294    5297    5307    5332    5335    5339    5343    5347
                5358    5381    5384    5394    5415    5419    5432    5456    5459    5471    5494    5497    5507    5525    5536
                5551    5554    5571    5573    5586    5589    5610    5617    5622    5624    5649    5655    5666    5689    5693
                5697    5704    5709    5714    5718    5722    5726    5729    5732    5741    5745    5748    5776    5779    5783
                5786    5790    5820    5834    6022    6062
.LIST              1    2011    2427    2428    2435    2499    2501    2502    2503    2504    2505    2506    2507    2508    2531
                2538    2539    2540    2542    2543    2545    2547    2549    2551    2553    2556    2557    2559    2561    2562
                2563    2564    2566    2567    2569    2570    2572    2574    2575    2576    2578    2580    2582    2584    2588
                2589    2591    2593    2595    2597    2599    2601    2604    2605    2606    2608    2610    2612    2614    2616
                2617    2619    2620    2629    2640    2647    2650    2651    2662    2667    2668    2673    2675    2676    2687
                2693    2694    2699    2700    2701    2712    2718    2719    2724    2725    2726    2737    2743    2744    2749
                2750    2756    2766    2771    2772    2773    2783    2788    2789    2794    2796    2797    2807    2812    2813
                2818    2819    2820    2830    2835    2836    2841    2842    2843    2852    2857    2858    2859    2870    2877
                2878    2889    2901    2912    2926    2938    2949    2961    2973    2988    3000    3014    3027    3035    3048
                3068    3071    3083    3096    3108    3123    3136    3148    3152    3171    3178    3179    3181    3182    3186
                3187    3197    3198    3200    3201    3205    3206    3216    3217    3219    3220    3224    3225    3232    3233
                3236    3237    3240    3241    3244    3245    3251    3252    3255    3256    3257    3264    3267    3268    3271
                3272    3277    3278    3281    3282    3283    3290    3293    3294    3297    3298    3303    3304    3307    3308
                3309    3316    3319    3320    3323    3324    3329    3330    3333    3334    3335    3342    3345    3346    3349
                3350    3355    3356    3359    3360    3361    3368    3371    3372    3375    3376    3381    3382    3385    3386
                3387    3394    3397    3398    3401    3402    3407    3408    3411    3412    3413    3421    3446    3447    3450
                3451    3455    3456    3459    3460    3464    3465    3468    3469    3473    3474    3477    3478    3482    3483
                3486    3487    3491    3492    3495    3496    3520    3531    3540    3542    3559    3562    3567    3570    3586
                3590    3592    3595    3596    3610    3613    3617    3620    3624    3627    3656    3657    3659    3673    3677
                3684    3688    3695    3698    3702    3707    3709    3724    3725    3726    3738    3741    3756    3759    3761
                3779    3784    3790    3791    3798    3799    3803    3804    3812    3813    3819    3820    3828    3829    3835
                3836    3844    3845    3851    3852    3859    3860    3862    3863    3871    3872    3880    3881    3889    3890
                3891    3910    3912    3925    3928    3932    3940    3948    3952    3968    3970    3971    3974    3983    3985
                4006    4014    4015    4024    4027    4036    4041    4044    4046    4055    4058    4062    4080    4083    4084
                4093    4097    4100    4102    4115    4119    4120    4123    4125    4134    4137    4139    4142    4144    4157
```

```
                4160    4161    4167    4170    4181    4203    4206    4207    4211    4214    4219    4221    4230    4231    4232
                4236    4238    4249    4250    4251    4255    4280    4285    4286    4290    4293    4297    4303    4304    4308
                4319    4320    4322    4340    4344    4345    4351    4355    4360    4365    4367    4380    4384    4385    4388
                4391    4405    4409    4410    4416    4419    4429    4448    4459    4460    4465    4468    4470    4488    4493
                4494    4500    4505    4508    4530    4540    4541    4553    4557    4563    4568    4571    4576    4591    4596
                4597    4603    4606    4610    4613    4627    4633    4634    4639    4643    4646    4648    4659    4664    4668
                4670    4682    4683    4684    4687    4692    4707    4711    4712    4715    4717    4730    4736    4737    4740
                4744    4748    4750    4762    4765    4766    4770    4790    4801    4804    4805    4811    4814    4816    4838
                4870    4872    4879    4883    4888    4892    4898    4905    4915    4923    4928    4929    4932    4934    4945
                4949    4950    4954    4958    4961    4992    5001    5002    5008    5016    5020    5024    5035    5037    5049
                5052    5053    5057    5070    5081    5084    5085    5089    5102    5115    5118    5119    5125    5131    5134
                5139    5150    5162    5164    5165    5171    5175    5189    5201    5208    5209    5214    5217    5229    5242
                5246    5247    5253    5257    5270    5282    5287    5288    5294    5297    5307    5324    5327    5329    5332
                5335    5339    5343    5347    5358    5370    5374    5375    5381    5384    5394    5405    5408    5409    5415
                5419    5432    5445    5451    5452    5456    5459    5471    5483    5487    5488    5494    5497    5507    5516
                5521    5522    5525    5536    5544    5547    5548    5551    5554    5562    5567    5568    5571    5573    5579
                5586    5589    5601    5610    5617    5622    5624    5640    5644    5645    5649    5655    5666    5676    5689
                5693    5697    5704    5709    5714    5718    5722    5726    5729    5732    5741    5745    5748    5764    5776
                5779    5783    5786    5790    5810    5817    5820    5835    5846    6091    6092    6101    6102    6103    6104
                6105    6106
.MACRO             1    2011    2466    2629    2646    2651    2672    2676    2701    2726    2756    2773    2793    2797    2820
                2843    2859    3151    3257    3283    3309    3335    3361    3387    3413    3520    3570    3627    3673    3676
                3697    3709    3761    3891    3927    3940    3952    3985    4046    4057    4102    4125    4144    4181    4221
                4238    4255    4292    4322    4351    4367    4388    4391    4429    4470    4500    4504    4508    4563    4570
                4576    4613    4648    4670    4686    4692    4717    4750    4790    4816    4892    4915    4934    4961    5037
                5070    5102    5125    5150    5171    5189    5229    5253    5270    5307    5358    5394    5415    5432    5471
                5507    5536    5554    5573    5589    5610    5617    5624    5655    5666    5697    5704    5708    5713    5717
                5721    5725    5728    5731    5740    5748    5778    5785    5789    6092
.MCALL          2011    2427    5820
.NLIST             1    2011    2427    2428    2435    2499    2501    2502    2503    2504    2505    2506    2507    2508    2531
                2538    2539    2540    2542    2543    2545    2547    2549    2551    2553    2556    2557    2559    2561    2562
                2563    2564    2566    2567    2569    2570    2572    2574    2575    2576    2578    2580    2582    2584    2588
                2589    2591    2593    2595    2597    2599    2601    2604    2605    2606    2608    2610    2612    2614    2616
                2617    2619    2620    2629    2640    2647    2650    2651    2662    2667    2668    2673    2675    2676    2687
                2693    2694    2699    2700    2701    2712    2718    2719    2724    2725    2726    2737    2743    2744    2749
                2750    2756    2766    2771    2772    2773    2783    2788    2789    2794    2796    2797    2807    2812    2813
                2818    2819    2820    2830    2835    2836    2841    2842    2843    2852    2857    2858    2859    2870    2877
                2878    2889    2901    2912    2926    2938    2949    2961    2973    2988    3000    3014    3027    3035    3048
                3068    3071    3083    3096    3108    3123    3136    3148    3152    3171    3178    3179    3181    3182    3186
                3187    3197    3198    3200    3201    3205    3206    3216    3217    3219    3220    3224    3225    3232    3233
                3236    3237    3240    3241    3244    3245    3251    3252    3255    3256    3257    3264    3267    3268    3271
                3272    3277    3278    3281    3282    3283    3290    3293    3294    3297    3298    3303    3304    3307    3308
                3309    3316    3319    3320    3323    3324    3329    3330    3333    3334    3335    3342    3345    3346    3349
                3350    3355    3356    3359    3360    3361    3368    3371    3372    3375    3376    3381    3382    3385    3386
                3387    3394    3397    3398    3401    3402    3407    3408    3411    3412    3413    3421    3446    3447    3450
                3451    3455    3456    3459    3460    3464    3465    3468    3469    3473    3474    3477    3478    3482    3483
                3486    3487    3491    3492    3495    3496    3520    3531    3540    3542    3559    3562    3567    3570    3586
                3590    3592    3595    3596    3610    3613    3617    3620    3624    3627    3656    3657    3659    3673    3677
                3684    3688    3695    3698    3702    3707    3709    3724    3725    3726    3738    3741    3756    3759    3761
                3779    3784    3790    3791    3798    3799    3803    3804    3812    3813    3819    3820    3828    3829    3835
                3836    3844    3845    3851    3852    3859    3860    3862    3863    3871    3872    3880    3881    3889    3890
                3891    3910    3912    3925    3928    3932    3940    3948    3952    3968    3970    3971    3974    3983    3985
                4006    4014    4015    4024    4027    4036    4041    4044    4046    4055    4058    4062    4080    4083    4084
                4093    4097    4100    4102    4115    4119    4120    4123    4125    4134    4137    4139    4142    4144    4157
                4160    4161    4167    4170    4181    4203    4206    4207    4211    4214    4219    4221    4230    4231    4232
                4236    4238    4249    4250    4251    4255    4280    4285    4286    4290    4293    4297    4303    4304    4308
```

```
                4319    4320    4322    4340    4344    4345    4351    4355    4360    4365    4367    4380    4384    4385    4388
                4391    4405    4409    4410    4416    4419    4429    4448    4459    4460    4465    4468    4470    4488    4493
                4494    4500    4505    4508    4530    4540    4541    4553    4557    4563    4568    4571    4576    4591    4596
                4597    4603    4606    4610    4613    4627    4633    4634    4639    4643    4646    4648    4659    4664    4668
                4670    4682    4683    4684    4687    4692    4707    4711    4712    4715    4717    4730    4736    4737    4740
                4744    4748    4750    4762    4765    4766    4770    4790    4801    4804    4805    4811    4814    4816    4838
                4870    4872    4879    4883    4888    4892    4898    4905    4915    4923    4928    4929    4932    4934    4945
                4949    4950    4954    4958    4961    4992    5001    5002    5008    5016    5020    5024    5035    5037    5049
                5052    5053    5057    5070    5081    5084    5085    5089    5102    5115    5118    5119    5125    5131    5134
                5139    5150    5162    5164    5165    5171    5175    5189    5201    5208    5209    5214    5217    5229    5242
                5246    5247    5253    5257    5270    5282    5287    5288    5294    5297    5307    5324    5327    5329    5332
                5335    5339    5343    5347    5358    5370    5374    5375    5381    5384    5394    5405    5408    5409    5415
                5419    5432    5445    5451    5452    5456    5459    5471    5483    5487    5488    5494    5497    5507    5516
                5521    5522    5525    5536    5544    5547    5548    5551    5554    5562    5567    5568    5571    5573    5579
                5586    5589    5601    5610    5617    5622    5624    5640    5644    5645    5649    5655    5666    5676    5689
                5693    5697    5704    5709    5714    5718    5722    5726    5729    5732    5741    5745    5748    5764    5776
                5779    5783    5786    5790    5810    5817    5820    5835    5846    6091    6092    6101    6102    6103    6104
                6105    6106
.PAGE           2037    2466    2514
.REM               1
.REPT           2435    2501    2504
.SBTTL          2038    2163    2174    2188    2337    2429    2436    2442    2468    2516    2629    2651    2676    2701    2726
                2756    2773    2797    2820    2843    2859    3152    3257    3283    3309    3335    3361    3387    3413    3519
                3520    3569    3570    3626    3627    3709    3761    3891    3952    3985    4046    4102    4125    4144    4181
                4221    4238    4255    4321    4322    4367    4391    4429    4470    4508    4576    4613    4648    4670    4691
                4692    4717    4750    4790    4816    4915    4934    4960    4961    5037    5070    5102    5150    5189    5229
                5269    5270    5307    5358    5394    5432    5470    5471    5507    5536    5553    5554    5573    5589    5624
                5666    5748    5823    5859    5932    6010    6078    6093
.TITLE          2026
.WORD           2435    2462    2464    2476    2479    2480    2481    2482    2485    2486    2487    2488    2489    2490    2499
                2501    2502    2503    2504    2505    2506    2507    2512    3911    3944    3969    3973    3981    4861    4864
                4867    4869    4876    4901    5838    5841    5927    6007    6106
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

```
*,DEKBAB.SEQ/SOL/CRF/PAGNUM/NL:TOC=DEKBAB.SML,DEKBAB.P11
RUN-TIME: 78 100 15 SECONDS
RUN-TIME RATIO: 280/194=1.4
CORE USED: 36K (71 PAGES)
```

Spooler runtime 21 Seconds, 92 KCS, 632 disk reads, 4 disk writes, 156 pages