RH70

CONTROLLER DIAGNOSTIC

MD-11-DERHA-B

EP-DERHA-B-DL-A

COPYRIGHT© 1976

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA

RH70

CONTROLLER DIAGNOSTIC

MD-11-DERHA-B

EP-DERHA-B-DL-A

COPYRIGHT©1976

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DERHA-B-D
PRODUCT NAME:	RH70 FUNCTIONAL CONTROLLER TEST
DATE RELEASED:	AUGUST, 1976
MAINTAINER:	DIAGNOSTIC GROUP

COPYRIGHT (C) 1975, 1976, DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES
NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS
DOCUMENT.
THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A
LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH
THE TERMS OF SUCH LICENSE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT
THAT IS NOT SUPPLIED BY DIGITAL.

1. ABSTRACT

THIS PROGRAM TESTS THE RH70 IN THE PDP11/70 SUBSYSTEM.
IT USES ANY WORKING RH70 PERIPHERAL CONNECTED TO THE RH70
UNDER TEST. IT CAN TEST UP TO FOUR RH70S CONNECTED ON THE
SUBSYSTEM. ALTHOUGH THE PERIPHERAL CONNECTED IS USED, THE
PERIPHERAL IS NOT TESTED BY THIS PROGRAM.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11/70 COMPUTER WITH CONSOLE TELETYPE, AND ANY ONE OF THE
RH70 PERIPHERALS, (FOR EXAMPLE RP04,RS03,RS04,TU16).
THE PERIPHERAL MUST BE COMPLETE AND WORKING. FOR EXAMPLE
THE RP04,RS03,RS04 MUST HAVE A SCRATCH PACK ON IT.
THE TU MUST HAVE A SCRATCH TAPE.

2.2 STORAGE

THIS PROGRAM HAS 16K WORDS OF MEMORY.

2.3 PRELIMINARY PROGRAMS

ALL PDP11/70 CPU DIAGNOSTICS MUST BE RUN ERROR FREE BEFORE
THIS DIAGNOSTIC IS RUN.

3. LOADING PROCEDURE

USE STANDARD LOADING PROCEDURES FOR .BIN TAPES.

4. STARTING PROCEDURE

4.1 OPERATIONAL SWITCH SETTINGS

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
11	INHIBIT ITERATIONS
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SWR<7:0>

4.2 STARTING ADDRESS

START AT ADDRESS 200 --- FOR STANDARD RH70.
START AT ADDRESS 210 --- FOR NONSTANDARD RH70.
STARTING ADDRESS 210 SHOULD BE USED TO TEST
ANY ONE RH INSTEAD OF ALL THE RH GIVEN BY 200 START

4.3 PROGRAM AND/OR OPERATOR ACTION

BEFORE STARTING THE PROGRAM, MAKE SURE THAT THE MAGNETIC MEDIUM IS PROPERLY POSITIONED:

1.) TU16 - MAGTAPE MUST BE AT LOAD POINT.
2.) RP - HEADS MUST BE IN HOME POSITION.

ON STARTING FROM 210 OPERATOR MUST CHOOSE BETWEEN THE RH70 PERIPHERALS AND TYPE THE BASE ADDRESS OF ANY OR ALL OF THE PERIPHERALS CONNECTED TO THE RH70.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

SEE SECTION 4.1

5.2 SUB-ROUTINE ABSTRACTS

SEE SECTION 9 "SUBROUTINES"

6. ERRORS

6.1 ERROR HALTS AND DESCRIPTION

ON DETECTION OF AN ERROR THE PROGRAM WILL PRINT ALL RELEVENT INFORMATION AND THEN PROCEDE ACCORDING TO THE SWITCH SETTINGS GIVEN IN 4.1. IF ALL SWITCHES ARE DOWN THE PROGRAM WILL CONTINUE.

```
                        DOCUMENT
                        **********
                        MAINDEC-11-DERHAB-A
                        **********
```

COPYRIGHT 1975, 1976
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

```
                                2       COPYRIGHT (C) 1975, 1976
                                        DIGITAL EQUIPMENT CORP.
                                        MAYNARD, MASS. 01754

                                        THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                        PACKAGE (MAINDEC-11-DZQAC-A4).

                                ;*****************************************************************
                        13      ;OPERATIONAL SWITCH SETTINGS
                                ;*****************************************************************

                                14
                                                SWITCH          USE
                                                ------          --------------------
                                                  15            HALT ON ERROR
                                                  14            LOOP ON TEST
                                                  13            INHIBIT ERROR TYPEOUTS
                                                  11            INHIBIT ITERATIONS
                                                  10            BELL ON ERROR
                                                   9            LOOP ON ERROR
                                                   8            LOOP ON TEST IN SWR<7:0>
                                                   7            STOP FURTHER COMPARES IF SW08 IS LOW
                                                   6            TYPE ALL REG. WITH ERROR IF SW8 LOW

                                ;*****************************************************************
                        27      ;BASIC DEFINITIONS
                                ;*****************************************************************

                                29      INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***

                                40      MISCELLANEOUS DEFINITIONS

                                46      GENERAL PURPOSE REGISTER DEFINITIONS

                                58      PRIORITY LEVEL DEFINITIONS

                                68      "SWITCH REGISTER" SWITCH DEFINITIONS

                                96      DATA BIT DEFINITIONS (BIT00 TO BIT15)

                                124     BASIC "CPU" TRAP VECTOR ADDRESSES
```

```
                        ;*****************************************************************
                        ;TRAP CATCHER
                        ;*****************************************************************
                        ;
                        ;       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
                        ;
                        ;*****************************************************************
                        ;STARTING ADDRESS(ES)
                        ;*****************************************************************
                        ;
                        ;       STARTING ADDRESS 200 FOR NORMAL STARTS
                        ;       THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
                        ;
                        ;       STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
                        ;
                        ;       STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERA
                        ;       AT THE DRIVE
                        ;
                        ;*****************************************************************
                        ;MEMORY MANAGEMENT DEFINITIONS
                        ;*****************************************************************
                        ;
                        ;       KT11 VECTOR ADDRESS
                        ;
                        ;       KT11 STATUS REGISTER ADDRESSES
                        ;
                        ;       KERNEL "I" PAGE DESCRIPTOR REGISTERS
                        ;
                        ;       KERNEL "I" PAGE ADDRESS REGISTERS
                        ;
                        ;*****************************************************************
                        ;COMMON TAGS
                        ;*****************************************************************
                        ;
                        ;       THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                        ;       USED IN THE PROGRAM.
```

```
                        MAINDEC-11-DERHAB-A

                ;*****************************************************************
        252     ;ERROR POINTER TABLE
                ;*****************************************************************

        254     ;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

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

        269     ;*****************************************************************

                ;*****************************************************************
       1587     ;REGISTER ADDRESSES
                ;*****************************************************************

                ;*****************************************************************
       1834     ;REGISTER TEST
                ;*****************************************************************

       1899     TEST 1  SIZE FOR RH DEVICES
                        THIS TEST DETERMINS WHICH RH DEVICE IS ON THE SYSTEM.
                        IF THE SYSTEM HAS MORE THAN ONE RH DEVICE THEN ONLY ONE
                        THE DEVICES WILL BE USED IN THE FOLLOWING TESTS.
                        THE ONE THAT WILL BE USED IS THE FIRST ONE THAT IS PRESE
                        IN THE FOLLOWING LIST
                        RS
                        RP
                        TM

                        THE WAY THE TEST WORKS IS AS FOLLOWS:-
                        A REFRENCE IS MADE TO THE CONTROL AND STATUS REGISTER 1
                        FOR THE RS BY A TST INSTRUCTION. IF IT RESPONDS THEN IT
                        ASSUMED PRESENT AND THE LOCATIONS FOR THE I/O REGISTERS
                        FILLED WITH THE APPROPIATE ADDRESSES.
                        THEN THE DRIVE TYPE REGISTER IS CHECKED TO HAVE GOOD
                        VALUES.
                        IF THE TST INSTRUCTION TRAPS TO A NON EXISTANT VECTOR
                        THEN A SIMILAR ATTEMPT IS MADE TO A RP CONTROL AND STATU
                        REGISTER 1.
                        THEN A TM IS TRYED.
                        THEN A MIXED SYSTEM WITH MORE THAN ONE RH DEVICS IS TRYE

       2195     TEST 2  UNIT UNDER TEST
                        THIS TYPES THE UNIT TO BE TESTED
                        AND IS THE FIRST TEST AFTER THE END OF THE PROGRAM
```

```
                                MAINDEC-11-DERHAB-A
                        2302    TEST 3  BIT BANG RHCS1

                        2304            TEST LOADING AND READING OF ALL POSSIBLE BITS
                                        IN RHCS1 REGISTER.
                                        USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                        AND WALKING 0'S (-2,-3,-5 ETC)
                                        IN THIS TEST SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO BITS WIL
                                        AND RDY!DVA BITS WILL ALWAYS BE SET
                                        AND BIT10 BITS WILL ALWAYS BE CLEARED
                                        IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                        THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                        GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                        SIGNAL.

                        2383    TEST 4  BIT BANG RHCS2

                        2385            TEST LOADING AND READING OF ALL POSSIBLE BITS
                                        IN RHCS2 REGISTER.
                                        USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                        AND WALKING 0'S (-2,-3,-5 ETC)
                                        IN THIS TEST 177740 BITS WILL NOT BE WRITTEN INTO
                                        AND IR BITS WILL ALWAYS BE SET
                                        IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                        THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                        GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                        SIGNAL.

                        2460    TEST 5  BIT BANG RHCS3

                        2462            TEST LOADING AND READING OF ALL POSSIBLE BITS
                                        IN RHCS3 REGISTER.
                                        USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                        AND WALKING 0'S (-2,-3,-5 ETC)
                                        IN THIS TEST 177660 BITS WILL NOT BE WRITTEN INTO
                                        AND 0 BITS WILL ALWAYS BE SET
                                        AND BIT9!BIT8!BIT7!BIT5!BIT4 BITS WILL ALWAYS BE CLEARED
                                        IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                        THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                        GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                        SIGNAL.

                        2541    TEST 6  BIT BANG RHWC

                        2543            TEST LOADING AND READING OF ALL POSSIBLE BITS
                                        IN RHWC REGISTER.
                                        USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                        AND WALKING 0'S (-2,-3,-5 ETC)
                                        IN THIS TEST 0 BITS WILL NOT BE WRITTEN INTO
                                        AND 0 BITS WILL ALWAYS BE SET
                                        IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                        THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                        GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                        SIGNAL.
```

```
                        MAINDEC-11-DERHAB-A
                2618    TEST 7  BIT BANG RHBA

                2620            TEST LOADING AND READING OF ALL POSSIBLE BITS
                                IN RHBA REGISTER.
                                USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                AND WALKING 0'S (-2,-3,-5 ETC)
                                IN THIS TEST 0 BITS WILL NOT BE WRITTEN INTO
                                AND 0 BITS WILL ALWAYS BE SET
                                AND BIT00 BITS WILL ALWAYS BE CLEARED
                                IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                SIGNAL.

                2699    TEST 10 BIT BANG RHBAE

                2701            TEST LOADING AND READING OF ALL POSSIBLE BITS
                                IN RHBAE REGISTER.
                                USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                AND WALKING 0'S (-2,-3,-5 ETC)
                                IN THIS TEST 177700 BITS WILL NOT BE WRITTEN INTO
                                AND 0 BITS WILL ALWAYS BE SET
                                AND 177700 BITS WILL ALWAYS BE CLEARED
                                IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                SIGNAL.

                2784    TEST 11 SILO TEST 1 (ONE WORD WRITE)
                                AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH
                                TOGETHER WITH UNIT NUMBER
                                ONE WORD OF ALL ZEROS IS WRITTEN INTO RHDB
                                BY "CLR"
                                "OR" (BIT #7) IS WAITED FOR BY A TRAP INSTRUCTION
                                CALLED "WAT" (NO TIMING IS DONE)
                                HOWEVER IF "OR" DOES NOT SET WITHIN "WAT" COUNT
                                DOWN AN ERROR IS REPORTED
                                RHDB IS READ AND CHECKED TO CONTAIN ZEROS
                                RHCS2 WILL BE CHECKED TO HAVE "IR" AND UNIT NUMBER
                                RHCS1,RHCS3,RHBA,RHBAE,RHWC, WILL BE CHECKED TO HAVE
                                APPROPIATE VALUES

                2973    TEST 12 SILO TEST 2 (ONE WORD WRITE)
                                AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH
                                TOGETHER WITH UNIT NUMBER
                                ONE WORD OF ALL ONES IS WRITTEN INTO RHDB
                                BY "CLR"
                                "OR" (BIT #7) IS WAITED FOR BY A TRAP INSTRUCTION
                                CALLED "WAT" (NO TIMING IS DONE)
                                HOWEVER IF "OR" DOES NOT SET WITHIN "WAT" COUNT
                                DOWN AN ERROR IS REPORTED
                                RHDB IS READ AND CHECKED TO CONTAIN ALL ONES
```

```
                                MAINDEC-11-DERHAB-A

                                                        RHCS2 WILL BE CHECKED TO HAVE "IR" AND UNIT NUMBER
                                                        RHCS1,RHCS3,RHBA,RHBAE,RHWC, WILL BE CHECKED TO HAVE
                                                        APPROPIATE VALUES

                                3144    TEST 13 SILO TEST 3 (TWO WORD WRITE)

                                3146            AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                                RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                                TOGETHER WITH UNIT NUMBER
                                                ONE WORD = 52525 IS WRITTEN INTO RHDB
                                                RHCS2 IS CHECKED TO HAVE "IR" AND UNIT NUMBER
                                                A SECOND WORD = 12525 IS WRITTEN INTO RHDB
                                                "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                                INSTRUCTION CALLED "WAT"
                                                RHDB IS READ AND CHECKED TO CONTAIN 52525
                                                RHCS2 IS CHECKED TO HAVE "IR", "OR" AND UNIT NUMBER
                                                RHDB IS READ A SECOND TIME AND CHECKED TO
                                                CONTAIN 125252
                                                RHCS2 IS CHECKED TO HAVE "IR" AND UNIT NUMBER
                                                THEN ALL REGISTERS RHCS1, RHCS3, RHBA, RHBAE, RHWC
                                                ARE CHECKED TO HAVE APPROPIATE VALUE


                                3416    TEST 14 SILO TEST 4 (COUNT PATTERN)

                                3418            AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                                RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH, TOGETHER
                                                WITH UNIT NUMBER
                                                EIGHT WORDS, A COUNT PATTERN 0 THRU 7 IS WRITTEN
                                                INTO RHDB
                                                "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                                INSTRUCTION CALLED "WAT"
                                                THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                                "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                                THEN RHBIS READ AND COMPARED TO HAVE THE RIGHT VALUE
                                                EIGHT TIES
                                                THEN RHCS2, RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                                CHECKED TO HAVE THE APPROPIATE VALUE

                                3808    TEST 15 SILO TEST 5 (FLOATING ONES)

                                3810            AFTER ARH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                                RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                                TOGETHER WITH UNIT NUMBER
                                                EIGHT WORDS, A PATTERN OF FLOATING ONES (1,2,4,10,20,40,
                                                IS WRITTEN INTO RHDB
                                                "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                                INSTRUCTION CALLED "WAT"
                                                THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                                "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                                THEN RHDB IS READ AND COMPARED TO HAVE THE
                                                RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                                THEN RHCS2 IS CHECKED TO HAVE "IR"
```

```
481
```

```
                        MAINDEC-11-DERHAB-A

                                        AND UNIT NUMBER
                                        THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                        CHECKED TO HAVE THE APPROPIATE VALUE

                        4227    TEST 16 SILO TEST 6 (FLOATING ONES IN UPPER BYTE)

                        4229            AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                        RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                        TOGETHER WITH UNIT NUMBER
                                        EIGHT WORDS, A PATTERN OF FLOATING ONES IN UPPER BYTE
                                        400,1000,2000,4000,10000,20000,40000,100000
                                        IS WRITTEN INTO RHDB
                                        "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                        INSTRUCTION CALLED "WAT"
                                        THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                        "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                        THEN RHDB IS READ AND COMPARED TO HAVE THE
                                        RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                        THEN RHCS2 IS CHECKED TO HAVE "IR"
                                        AND UNIT NUMBER
                                        THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                        CHECKED TO HAVE THE APPROPIATE VALUE

                        4649    TEST 17 SILO TEST 7 (FLOATING 0 IN LOWER BYTE)

                        4651            AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                        RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                        TOGETHER WITH UNIT NUMBER
                                        EIGHT WORDS OF FLOATING ZEROS 177776, 177775, 177773,
                                        177767, 177757, 177737, 177677, 177577
                                        IS WRITTEN INTO RHDB
                                        "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                        INSTRUCTION CALLED "WAT"
                                        THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                        "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                        THEN RHDB IS READ AND COMPARED TO HAVE THE
                                        RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                        THEN RHCS2 IS CHECKED TO HAVE "IR"
                                        AND UNIT NUMBER
                                        THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                        CHECKED TO HAVE THE APPROPIATE VALUE

                        5070    TEST 20 SILO TEST 8 (FLOATING ZEROS IN UPPER BYTE)

                        5072            AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                        RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,

                        5074            TOGETHER WITH UNIT NUMBER
                                        EIGHT WORDS, A PATTERN OF FLOATING ZEROS IN UPPER BYTE
                                        177377, 177677, 175777, 173777, 167777, 157777, 137777,
                                        IS WRITTEN INTO RHDB
                                        "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                        INSTRUCTION CALLED "WAT"
                                        THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
```

```
                                MAINDEC-11-DERHAB-A

                                                        "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                                        THEN RHDB IS READ AND COMPARED TO HAVE THE
                                                        RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                                        THEN RHCS2 IS CHECKED TO HAVE "IR"
                                                        AND UNIT NUMBER
                                                        THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                                        CHECKED TO HAVE THE APPROPIATE VALUE

                                5508    TEST 21 RHCS1 - MCPE BIT #13 (PARITY LINE = 0)

                                5510            AN RH CLEAR (SET BIT #5 IN RHCS2) IS GIVEN TO
                                                CLEAR ALL DEVICE REGISTERS
                                                THEN RHER1 (ERROR REGISTER 1) IS CHECKED TO HAVE
                                                ZEROS
                                                SET "PAT" (BIT #4 IN RHCS2) TO INVERT PARITY CHECKING
                                                READ ANY DEVICE REGISTER - HERE RHER1
                                                READ AND CHECK RHCS1 TO CONTAIN SC (BIT #15)
                                                AND MCPE (BIT #13)
                                                WRITE "1" INTO TRE (BIT #14 IN RHCS1)
                                                READ RHCS1 SC, MCPE, AND RDY SHOULD BE SET
                                                GIVE AN RH CLEAR (CLR - BIT #5 IN RHCS2)
                                                CHECK RHCS1 TO HAVE RDY
                                                CHECK RHCS2 TO HAVE ONLY IR AND UNIT NUMBER
                                                CHECK RHCS3, RHBA, RHBAE, RHWC TO HAVE APPROPIATE
                                                VALUES.

                                5785    TEST 22 RHCS1 - MCPE BIT #13 (PARITY LINE = 1)

                                5787            AN RH CLEAR (SET BIT #5 IN RHCS2) IS GIVEN TO CLEAR
                                                ALL DEVICE REGISTERS
                                                WRITE A ONE INTO DISK ADDRESS REGISTER (RHDA)
                                                (IN TAPE DRIVES CALLED REGISTER)
                                                SET "PAT" (RHCS2 BIT #4) THIS WILL INVERT THE PARITY CHE
                                                READ RHDA AND COMPARE IT HAS ONE IN IT
                                                THIS READING SHOULD SET SC, MCPE IN RHCS1
                                                CHECK RHCS1 TO HAVE ONLY RDY SET
                                                CHECK RHCS2, RHCS3, RHBA, RHBAE, RHWC TO HAVE APPROPIATE
                                                VALUES

                                5959    TEST 23 TEST DUPLICATED A16 (RHCS1 BIT #8)

                                5961            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                MOVE 400 (ONE INTO A16) IN RHCS1
                                                READ RHCS1 TO CONTAIN 600 (BIT #8 AND RDY)
                                                READ RHBAE TO CONTAIN "1" (BIT #0 HIGH)


                                                MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #0)
                                                READ RHBAE TO CONTAIN 0
                                                READ RHCS1 TO CONTAIN ONLY RDY (BIT #8 IN RHCS1 IS ZERO)
```

```
                                MAINDEC-11-DERHAB-A
                                6079    TEST 24 TEST DUPLICATED A17 (RHCS1 BIT #9)
                                6081            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                MOVE 400 (ONE INTO A17) IN RHCS1
                                                READ RHCS1 TO CONTAIN 1200 (BIT #9 AND RDY)
                                                READ RHBAE TO CONTAIN "2" (BIT #1 HIGH)


                                                MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #1)
                                                READ RHBAE TO CONTAIN 2
                                                READ RHCS1 TO CONTAIN ONLY RDY (BIT #9 IN RHCS1 IS ZERO)



                                6199    TEST 25 TEST DUPLICATED IE (RHCS1 BIT #6)
                                6201            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                MOVE 100 (ONE INTO IE) IN RHCS1
                                                READ RHCS1 TO CONTAIN 300 (BIT #6 AND RDY)
                                                READ RHCS3 TO CONTAIN "100" (3IT #6 HIGH)


                                                MOVE 0 INTO RHCS3 (WRITING 0 INTO RHCS3 BIT #6)
                                6208            READ RHCS3 TO CONTAIN 100
                                                READ RHCS1 TO CONTAIN ONLY RDY (BIT #6 IN RHCS1 IS ZERO)



                                6319    TEST 26 RHCS1 PROGRAM ERROR (PGE BIT #10)
                                6321            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                SET UP FOR A 10 WORD WRITE
                                                SET "GO" (RHCS1 BIT #0) TWICE IN TWO SUCCESSIVE
                                                INSTRUCTIONS
                                                THIS SHOULD SET, SC AND TRE IN RHCS1
                                                AND PGE SHOULD BE SET IN RHCS2
                                                RHCS3, ARE CHECKED
                                                THE NUMBER OF WORDS ARE LARGE ENOUGH SO THAT AFTER
                                                ONE "BIS" INSTRUCTION TO SET "GO" IN RHCS1
                                                THERE IS SUFFICIENT TIME TO GIVE ANOTHER "BIS" INSTRUCTI
                                                TO SET "GO" BEFORE THE FIRST "GO" HAS TIME TO COMPLETE


                                6415    TEST 27 RHCS2 - BUS ADDRESS INHIBIT BIT #3
                                6417            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                SET UP FOR A 2 WORD WRITE FROM A BUFFER TAGGED "WRFROM"
                                                SET BUS ADDRESS INHIBIT RHCS2 BIT #3 "BAI"
                                                AT THE END OF WRITE CHECK
                                                RHCS1 TO HAVE ONLY RDY AND COMMAND
                                                RHCS2 TO HAVE BAI
                                                RHCS3 TO HAVE 0
```

```
                                MAINDEC-11-DERHAB-A
                                                6424            RHBA TO HAVE ADDRESS OF "WRFROM"
                                                                RHBAE AND RHWC TO HAVE ZEROS

                                                                NOW SET UP READ FOR THE SAME DATA WITH
                                                                BAI SET TO READ INTO A BUFFER TAGGED "REINTO"
                                                                AFTER READ CHECK
                                                                RHCS1 TO HAVE ONLY RDY AND COMMAND
                                                                RHCS2 TO HAVE BAI
                                                                RHCS3 TO HAVE 0
                                                                RHBA TO HAVE ADDRESS OF "REINTO"
                                                                RHBAE AND RHWC TO HAVE ZEROS
                                                                DATA IN REINTO BUFFER IS CHECKED WITH DATA IN
                                                                WRFROM BUFFER

                                        6571    TEST 30 RHSC2 MDPE BIT #8 AND RHCS3 IPCK0 BIT #0
                                                        CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                        SET IPCK0 (RHCS3 BIT #0)
                                                        MOVE ALL ZEROS INTO RHDB ONCE
                                                        THIS SHOULD SET TRE SC IN RHCS1
                                                        READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
                                                        CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15
                                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                                        ALL ERRORS ARE CLEARED
                                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                        6807    TEST 31 RHSC2 MDPE BIT #8 AND RHCS3 IPCK1 BIT #1
                                                        CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                        SET IPCK1 (RHCS3 BIT #1)
                                                        MOVE ALL ZEROS INTO RHDB ONCE
                                                        THIS SHOULD SET TRE SC IN RHCS1
                                                        READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
                                                        CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15
                                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                                        ALL ERRORS ARE CLEARED
                                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                        7043    TEST 32 RHSC2 MDPE BIT #8 AND RHCS3 IPCK2 BIT #2
                                                        CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                        SET IPCK2 (RHCS3 BIT #2)
                                                        MOVE ALL ZEROS INTO RHDB TWICE
                                                        THIS SHOULD NOT SET TRE SC IN RHCS1
                                                        READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
                                                        CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15
                                                        READ RHDB A SECOND TIME. CHECK RHCS1, RHCS2, RHCS3
                                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                                        ALL ERRORS ARE CLEARED
                                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                        7345    TEST 33 RHSC2 MDPE BIT #8 AND RHCS3 IPCK3 BIT #3
                                                        CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                        SET IPCK3 (RHCS3 BIT #3)
                                                        MOVE ALL ZEROS INTO RHDB TWICE
                                                        THIS SHOULD NOT SET TRE SC IN RHCS1
                                                        READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
```

```
                                MAINDEC-11-DERHAB-A

                                        CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15
                                        READ RHDB A SECOND TIME. CHECK RHCS1, RHCS2, RHCS3
                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ALL ERRORS ARE CLEARED
                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                        7648    TEST 34 RHCS2-WCE BIT #14 AND RHCS3-WCE OW BIT #12

                        7650            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                        SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                        ODD WORD BOUNDARY
                                        DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                        ODD WORD BOUNDARY
                                        THIS SHOULD SET WCE OW (RHCS3 BIT #12)
                                        AND WCE (RHCS2 BIT #14)
                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ALL ERRORS ARE CLEARED
                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                        7881    TEST 35 RHCS2-WCE BIT #14 AND RHCS3-WCE OW BIT #12

                        7383            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                        SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                        ODD WORD BOUNDARY
                                        DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                        ODD WORD BOUNDARY
                                        THIS SHOULD SET WCE OW (RHCS3 BIT #12)
                                        AND WCE (RHCS2 BIT #14)
                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ALL ERRORS ARE CLEARED
                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                        8114    TEST 36 RHCS2-WCE BIT #14 AND RHCS3-WCE EW BIT #11

                        8116            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                        SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                        EVEN WORD BOUNDARY
                                        DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                        EVEN WORD BOUNDARY
                                        THIS SHOULD SET WCE EW (RHCS3 BIT #11)
                                        AND WCE (RHCS2 BIT #14)
                                        "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ALL ERRORS ARE CLEARED
                                        CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                        8347    TEST 37 RHCS2-WCE BIT #14 AND RHCS3-WCE EW BIT #11
```

```
                                MAINDEC-11-DERHAB-A
                                                8349            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                                                SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                                                EVEN WORD BOUNDARY
                                                                DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                                                EVEN WORD BOUNDARY
                                                                THIS SHOULD SET WCE EW (RHCS3 BIT #11)
                                                                AND WCE (RHCS2 BIT #14)
                                                                "CLR" (RHCS2 BIT #5) IS GIVEN
                                                                ALL ERRORS ARE CLEARED
                                                                CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                                8581    TEST 40 TEST DBL (RHCS3 BIT #10) TEST A

                                                8583            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)

                                                8584            SET UP FOR A NINE WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                DO A NINE WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                8694    TEST 41 TEST DBL (RHCS3 BIT #10) TEST B

                                                8696            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                DO A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                8808    TEST 42 TEST DBL (RHCS3 BIT #10) TEST C

                                                8810            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
                                                                DO A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                8921    TEST 43 TEST DBL (RHCS3 BIT #10) TEST D

                                                8923            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDAR
                                                                DO A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
```

```
                                MAINDEC-11-DERHAB-A
                                                9034    TEST 44 TEST DBL (RHCS3 BIT #10) TEST E

                                                9036            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                WITH BAI IN RHCS2 BIT #3 SET
                                                                DO A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9153    TEST 45 TEST DBL (RHCS3 BIT #10) TEST F

                                                9155            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
                                                                DO A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
                                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9268    TEST 46 TEST DBL (RHCS3 BIT #10) TEST G

                                                9270            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A NINE WORD READ FROM AN EVEN WORD BOUNDARY
                                                                DO A NINE WORD READ FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9381    TEST 47 TEST DBL (RHCS3 BIT #10) TEST H

                                                9383            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
                                                                DO A TEN WORD READ FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9495    TEST 50 TEST DBL (RHCS3 BIT #10) TEST I

                                                9497            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TEN WORD READ FROM AN ODD WORD BOUNDARY
                                                                DO A TEN WORD READ FROM AN ODD WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3

                                                9502            CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9608    TEST 51 TEST DBL (RHCS3 BIT #10) TEST J
```

```
                                MAINDEC-11-DERHAB-A
                                                9610            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
                                                                DO A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9721    TEST 52 TEST DBL (RHCS3 BIT #10) TEST K

                                                9723            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
                                                                DO A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
                                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9835    TEST 53 TEST DBL (RHCS3 BIT #10) TEST L

                                                9837            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
                                                                WITH BAI IN RHCS2 BIT #3 SET
                                                                DO A TEN WORD READ FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                9955    TEST 54 TEST DBL (RHCS3 BIT #10) TEST M

                                                9957            CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A ONE WORD WRITE REVERSE FROM AN EVEN WORD BO
                                                                DO A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                10068   TEST 55 TEST DBL (RHCS3 BIT #10) TEST N

                                                10070           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BO
                                                                DO A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                                10181   TEST 56 TEST DBL (RHCS3 BIT #10) TEST O

                                                10183           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                                SET UP FOR A TWO WORD WRITE REVERSE FROM AN ODD WORD BOU
                                                                DO A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
                                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                                CHECK RHCS3
                                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
```

```
                                MAINDEC-11-DERHAB-A
                        10295   TEST 57 TEST DBL (RHCS3 BIT #10) TEST P

                        10297           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        SET UP FOR A THREE WORD WRITE REVERSE FROM AN EVEN WORD
                                        DO A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                                        THIS SHOULD SET RHCS3 BIT #10 DBL
                                        CHECK RHCS3
                                        CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                        10409   TEST 60 TEST DBL (RHCS3 BIT #10) TEST Q

                        10411           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        SET UP FOR A THREE WORD WRITE REVERSE FROM AN ODD WORD B
                                        DO A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
                                        THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                        CHECK RHCS3
                                        CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                        10522   TEST 61 TEST DBL (RHCS3 BIT #10) TEST R

                        10524           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BO
                                        WITH BAI IN RHCS2 BIT #3 SET
                                        DO A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY

                        10528           THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                        CHECK RHCS3
                                        CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                        10642   TEST 62 TEST DBL (RHCS3 BIT #10) TEST S

                        10644           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        SET UP FOR A NINE WORD READ REVERSE FROM AN EVEN WORD BO
                                        DO A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
                                        THIS SHOULD SET RHCS3 BIT #10 DBL
                                        CHECK RHCS3
                                        CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                        IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE

                        10761   TEST 63 TEST DBL (RHCS3 BIT #10) TEST T

                        10763           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        SET UP FOR A TEN WORD READ REVERSE FROM AN EVEN WORD BOU
                                        DO A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
                                        THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                        CHECK RHCS3
                                        CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                        IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
```

```
                                MAINDEC-11-DERHAB-A
                                10879   TEST 64 TEST DBL (RHCS3 BIT #10) TEST U
                                10881           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUN
                                                DO A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                CHECK RHCS3
                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                                IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE

                                10998   TEST 65 TEST DBL (RHCS3 BIT #10) TEST V

                                11000           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                SET UP FOR A ELEVEN WORD READ REVERSE FROM AN EVEN WORD
                                                DO A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
                                                THIS SHOULD SET RHCS3 BIT #10 DBL
                                                CHECK RHCS3
                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                                IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE

                                11117   TEST 66 TEST DBL (RHCS3 BIT #10) TEST W

                                11119           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                SET UP FOR A ELEVEN WORD READ REVERSE FROM AN ODD WORD B
                                                DO A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY

                                11122           THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                CHECK RHCS3
                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                                IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE

                                11235   TEST 67 TEST DBL (RHCS3 BIT #10) TEST X

                                11237           CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                                SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUN
                                                WITH BAI IN RHCS2 BIT #3 SET
                                                DO A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
                                                THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                                CHECK RHCS3
                                                CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                                IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE

                                11362   TEST 70 END OF ONE RH
                                                THIS IS THE END OF TEST FOR ONE RH
                                                IF THERE ARE MORE RH THEN THE PROGRAM
                                                JUMPS TO TEST 2 FOR NEXT RH TEST
                                                END PASS IS REACHED ONLY AFTER ALL RH ARE COMPLETE
```

```
1007                            MAINDEC-11-DERHAB-A
1008                                    ;*****************************************************************
1009                            11400   ;END OF PASS ROUTINE
1010                                    ;*****************************************************************
1011                                    ;
1012                                    ;11402   INCREMENT THE PASS NUMBER ($PASS)
1013                                    ;        TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
1014                                    ;        IF THERES A MONITOR GO TO IT
1015                                    ;        IF THERE ISN'T JUMP TO TST2
1016                                    ;
1017                                    ;*****************************************************************
1018                            11439   ;SUBROUTINES
1019                                    ;*****************************************************************
1020                                    ;
1021                                    ;*****************************************************************
1022                            11655   ;RS DATA TRANSFER SUBROUTINE
1023                                    ;*****************************************************************
1024                                    ;
1025                                    ;11657           THIS SUBROUTINE WRITES OR READS OR DOES A
1026                                    ;                WRITE CHECK ON CYLINDER 0 TRACK 0 SECTOR 0
1027                                    ;                IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY
1028                                    ;                FILLED
1029                                    ;                WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
1030                                    ;                MAIN PROGRAM IT MEANS THE TRANSFER IS COMPLETE
1031                                    ;
1032                                    ;                THE CALL IS BY A JSR R0,@COMND COMAND
1033                                    ;
1034                                    ;*****************************************************************
1035                            11704   ;RP04 DATA TRANSFER SUBROUTINE
1036                                    ;*****************************************************************
1037                                    ;
1038                                    ;11706           THIS SUBROUTINE WRITE/READ/WRITE CHECK ON THE RP04 CYLIN
1039                                    ;                TRACK 0, SECTOR 0.
1040                                    ;
1041                                    ;11708           IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
1042                                    ;                WHEN IT RETURNS FROM THIS SUBROUTINE TO THE MAIN
1043                                    ;                PROGRAM IT MEANS THE COMMAND IS COMPLETE
1044                                    ;
1045                                    ;                THE CALL IS BY A JSR R0,@COMND COMMAND.
1046
```

```
                                MAINDEC-11-DERHAB-A
                        ;*****************************************************************
                11758   ;TM02 DATA TRANSFER SUBROUTINE
                        ;*****************************************************************
                        ;
                11760   ;               THIS SUBROUTINE WRITES/READS/WRITE CHECKS ON THE TM02-TU
                        ;               ON THE FIRST BLOCK FROM BEGINNING OF TAPE (BOT)
                        ;
                11762   ;               IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
                        ;               WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
                        ;               MAIN PROGRAM IT MEANS THE COMMAND IS COMPLETE.
                        ;
                        ;               THE CALL IS
                        ;               JSR     R0,@COMND
                        ;
                11915   ;               THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
                        ;               ADDRESS FROM 176700 TO ANY TYPED VALUE
                        ;
                11982   ;***************************
                        ;
                        ;*****************************************************************
                11985   ;SCOPE HANDLER ROUTINE
                        ;*****************************************************************
                        ;
                11987   ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                        ;AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7
                        ;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                        ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        ;SW14=1  LOOP ON TEST
                        ;SW11=1  INHIBIT ITERATIONS
                        ;SW09=1  LOOP ON ERROR
                        ;SW08=1  LOOP ON TEST IN SWR<7:0>
                        ;CALL
                        ;       SCOPE           ;;SCOPE=IOT
                        ;
                        ;*****************************************************************
                12051   ;CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
                        ;*****************************************************************
                        ;
                12053   ;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
                        ;
```

```
                                MAINDEC-11-DERHAB-A
                                ;*******************************************************************
                        12119   ;TYPE ROUTINE
                                ;*******************************************************************

                                12121   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

                                ;*******************************************************************
                        12192   ;TTY INPUT ROUTINE
                                ;*******************************************************************

                                12201   TK INITIALIZE ROUTINE
                                        THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                        SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                        CALL:
                                                JSR     PC,$TKINT
                                                RETURN

                                12218   TK SERVICE ROUTINE
                                        THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                        BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                        IT IN THE QUEUE.
                                        IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
                                        UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPE

                                12245   THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                        CALL:
                                                RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
                                                RETURN HERE             ;;CHARACTER IS ON THE STACK
                                                                        ;;WITH PARITY BIT STRIPPED OFF

                                12266   THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                        CALL:
                                                RDLIN                   ;;INPUT A STRING FROM THE TTY
                                                RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WIL
                                                                        ;;TERMINATOR WILL BE A BYTE OF A
```

```
                        MAINDEC-11-DERHAB-A
                                ;*******************************************************************
        12303                   ;READ AN OCTAL NUMBER FROM THE TTY
                                ;*******************************************************************
                                ;
        12305                   ;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                ;CHANGE IT TO BINARY.
                                ;THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                ;OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPE
                                ;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUS
                                ;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RE
                                ;CALL:
                                ;       RDOCT                   ;;READ AN OCTAL NUMBER
                                ;       RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF T
                                ;                               ;;HIGH ORDER BITS ARE IN $HIOCT
                                ;
                                ;*******************************************************************
        12357                   ;ERROR HANDLER ROUTINE
                                ;*******************************************************************
                                ;
        12359                   ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                ;AND GO TO $ERRTYP ON ERROR
                                ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;SW15=1  HALT ON ERROR
                                ;               HALT CAN OCCUR BEFORE AND AFTER THE ERROR TYPEOU
                                ;SW13=1  INHIBIT ERROR TYPEOUTS
                                ;SW10=1  BELL ON ERROR
                                ;SW09=1  LOOP ON ERROR
                                ;CALL
                                ;       ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                                ;
                                ;*******************************************************************
        12401                   ;ERROR MESSAGE TYPEOUT ROUTINE
                                ;*******************************************************************
                                ;
        12402                   ;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE
                                ;ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE
                                ;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
                                ;IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
                                ;WITH ONLY MINOR CHANGES
                                ;FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
                                ;ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
                                ;
        12409                   ;SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
                                ;AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
                                ;AND NOT THE ERROR MESSAGE AND HEADER.
                                ;
        12627                   ;*******************************************************************
                                ;*******************************************************************
```

```
                        MAINDEC-11-DERHAB-A
                                ;*****************************************************************
        12631                   ;BINARY TO OCTAL (ASCII) AND TYPE
                                ;*****************************************************************
                        12633   ;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
                                ;*****************************************************************
        12709                   ;TRAP DECODER
                                ;*****************************************************************
                        12711   ;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.
                                ;*****************************************************************
        12724                   ;TRAP TABLE
                                ;*****************************************************************
                        12726   ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLE
                                ;BY THE "TRAP" INSTRUCTION.
```

```
                                MAINDEC-11-DERHAB-A
                                ;*******************************************************************
                        12746   ;POWER DOWN AND UP ROUTINES
                                ;*******************************************************************

                        12786   ;*******************************************************************
                                ;*******************************************************************
                                %
```

```
                        .TITLE  MAINDEC-11-DERHAB-A
                        ;*COPYRIGHT (C) 1975-1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.
                        ;*

                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>
                        ;*         7            STOP FURTHER COMPARES IF SW08 IS LOW
                        ;*         6            TYPE ALL REG. WITH ERROR IF SW8 LOW
                        .SBTTL  BASIC DEFINITIONS

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

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

                        ;*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  R6,SP           ;;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
000010                          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                          TBITVEC=14              ;;"T" BIT
000014                          TRTVEC= 14              ;;TRACE TRAP
000014                          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                          PWRVEC= 24              ;;POWER FAIL
000030                          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                          TRAPVEC=34              ;;"TRAP" TRAP
000060                          TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                          TPVEC=  64              ;;TTY PRINTER VECTOR
000240                          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                                .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
000174                                  .=174
000174  000000                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  005522                  JMP     @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
000046                                  .=46
000046  037544                          $ENDAD
000052                                  .=52
000052  060000                          60000
000210                                  .=210
000210  000137  005532                  JMP     @#BEGIN2        ;JUMP SELECT TEST
                                ;*STARTING ADDRESS 200 FOR NORMAL STARTS
                                ;*THIS WILL TEST ALL RP04'S ON THE SYSTEM A SINGLE DRIVE AT A TIME
                                ;*
                                ;*STARTING ADDRESS 210 WILL TEST ONLY ONE SPECIFIED DRIVE
                                ;*
                                ;*STARTING ADDRESS 220 WILL JUMP OVER THE TESTS REQUIRING AN OPERATOR
                                ;*AT THE DRIVE
                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                                ;*KT11 VECTOR ADDRESS

000250                          MMVEC=  250

                                ;*KT11 STATUS REGISTER ADDRESSES

177572                          SR0=    177572
177574                          SR1=    177574
```

```
1419            177576                  SR2=    177576
1420            172516                  SR3=    172516
1421
1422                                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
1423
1424            172300                  KIPDR0= 172300
1425            172302                  KIPDR1= 172302
1426            172304                  KIPDR2= 172304
1427            172306                  KIPDR3= 172306
1428            172310                  KIPDR4= 172310
1429            172312                  KIPDR5= 172312
1430            172314                  KIPDR6= 172314
1431            172316                  KIPDR7= 172316
1432
1433                                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
1434
1435            172340                  KIPAR0= 172340
1436            172342                  KIPAR1= 172342
1437            172344                  KIPAR2= 172344
1438            172346                  KIPAR3= 172346
1439            172350                  KIPAR4= 172350
1440            172352                  KIPAR5= 172352
1441            172354                  KIPAR6= 172354
1442            172356                  KIPAR7= 172356
1443
1444                                    ;;**************************************************************************
1445            001110                          .=1110
```

```
                                .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 ADDRESS
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 ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
                        $TKS:   177560                  ;;TTY KBD STATUS
                        $TKB:   177562                  ;;TTY KBD BUFFER
                        $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
                        $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001200  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001202  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001204  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001206  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001210  000000          $TMP5:  .WORD   0               ;;USER DEFINED
001212  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001214  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001216  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001222  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001223  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
```

```
  1502  001224  000012                    $LF:    .ASCIZ  <12>            ;;LINE FEED
  1503                                    ;;*****************************************************************
```

```
                                .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

001226                          $ERRTB:

                                ;*****************************************************************
                                ;



                                ;ITEM 1
001226  046700                          EM1                     ;THE RH BASE ADDRESS WAS 772040
                                                                ;INDICATING A RS04 OR RS03
                                                                ;BUT THE DRIVE TYPE DID NOT
                                                                ;CONTAIN 0,1,2,3 OR 4
001230  065121                          DH1                     ;PC
                                                                ;RHDT
001232  066554                          DT1                     ;$ERRPC,$BDDAT,0
001234  066736                          DF1                     ;0

                                ;ITEM 2
001236  047052                          EM2                     ;THE RH BASE ADDRESS WAS 776700
                                                                ;INDICATING AN RP04, RP05, OR RP06
                                                                ;BUT THE DRIVE TYPE DID NOT
                                                                ;CONTAIN 20020 24020, 20021, 24021,
                                                                ;20022, OR 24022

001240  065121                          DH1                     ;PC
                                                                ;RHDT
001242  066554                          DT1                     ;$ERRPC,$BDDAT,0
001244  066736                          DF1                     ;0

                                ;ITEM 3
001246  047254                          EM3                     ;THE RH BASE ADDRESS WAS 772440
                                                                ;INDICATING A TM02. BUT THE
                                                                ;DRIVE TYPE DID NOT CONTAIN
                                                                ;142010 INDICATING NO
                                                                ;TM02
001250  065121                          DH1                     ;PC
                                                                ;RHDT
001252  066554                          DT1                     ;$ERRPC,$BDDAT,0
001254  066736                          DF1                     ;0,0

                                ;ITEM 4
```

```
001256  047410                          EM4             ;THE RH BASE ADDRESS WAS 776300
                                                        ;INDICATING MORE THAN ONE
                                                        ;RH DEVICE. BUT THE DRIVE
                                                        ;TYPE DID NOT CONTAIN
                                                        ;0,1,2,3,4,20020,24020,20021,24021,
                                                        ;20022,24022, OR 142010 INDICATING NO
                                                        ;RH DEVICE IS PRESENT
001260  065121                          DH1             ;PC
                                                        ;RHDT
001262  066554                          DT1             ;$ERRPC,$BDDAT,0
001264  066736                          DF1             ;0,0

                                ;ITEM 5
001266  047707                          EM5             ;THE UNIBUS TIMED OUT
                                                        ;FOR EACH OF THE FOLLOWING
                                                        ;ADDRESSES, INDICATING
                                                        ;NO RH ON THE SYSTEM
                                                        ;
                                                        ;SO ABORT PROGRAM
001270  065130                          DH5             ;PC
                                                        ;UNIBUS ADDRESSES ON WHICH
                                                        ;TIME OUT OCCURRED
001272  066562                          DT5             ;$ERRPC,RSCS1,RPCS1,TMCS1,MIXCS1
001274  066740                          DF5             ;0,0,0,0,0

                                ;ITEM 6
001276  050062                          EM6             ;AFTER AN RH CLEAR (BIT #5
                                                        ;IN RHCS2) RHCS2 DOES NOT
                                                        ;HAVE ONLY IR AND UNIT
                                                        ;NUMBER
001300  065204                          DH6             ;PC
                                                        ;TEST NO
                                                        ;RHCS2 GOOD
                                                        ;RHCS2 BAD
001302  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001304  066745                          DF6             ;0,0,0,0

                                ;ITEM 7
001306  050200                          EM7             ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK CAUSED RHDB TO
                                                        ;HAVE  WRONG VALUE GIVEN IN BAD RHDB
001310  065247                          DH7             ;PC
                                                        ;TEST NO
                                                        ;RHDB GOOD
                                                        ;RHDB BAD
001312  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001314  066745                          DF6             ;0,0,0,0

                                ;ITEM 10
001316  050372                          EM10            ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHCS2 TO
                                                        ;HAVE WRONG DATA
```

```
001320  065204                          DH6             ;GIVEN IN BAD RHCS2
                                                        ;PC
                                                        ;TEST NUMBER
                                                        ;RHCS2 GOOD
                                                        ;RHCS2 BAD
001322  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001324  066745                          DF6             ;0,0,0,0

                                ;ITEM 11
001326  050565                          EM11            ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHCS1 TO
                                                        ;HAVE WRONG DATA
                                                        ;GIVEN IN BAD RHCS1
001330  065307                          DH11            ;PC
                                                        ;TEST NUMBER
                                                        ;RHCS1 GOOD
                                                        ;RHCS1 BAD
001332  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001334  066745                          DF6             ;0,0,0,0

                                ;ITEM 12
001336  050760                          EM12            ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHCS3 TO
                                                        ;HAVE WRONG DATA
                                                        ;GIVEN IN BAD RHCS3
001340  065351                          DH12            ;PC
                                                        ;TEST NUMBER
                                                        ;RHCS3 GOOD
                                                        ;RHCS3 BAD
001342  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001344  066745                          DF6             ;0,0,0,0

                                ;ITEM 13
001346  051153                          EM13            ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHBA TO
                                                        ;HAVE WRONG DATA
                                                        ;GIVEN IN BAD RHBA
001350  065413                          DH13            ;PC
                                                        ;TEST NUMBER
                                                        ;RHBA GOOD
                                                        ;RHBA BAD
001352  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001354  066745                          DF6             ;0,0,0,0

                                ;ITEM 14
001356  051345                          EM14            ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHBAE TO
                                                        ;HAVE WRONG DATA
```

```
1672
1673  001360  065453                          DH14                    ;GIVEN IN BAD RHBAE
1674                                                                    ;PC
1675                                                                    ;TEST NUMBER
1676                                                                    ;RHBAE GOOD
1677  001362  066576                          DT6                     ;RHBAE BAD
1678  001364  066745                          DF6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1679                                                                    ;0,0,0,0
1680                                  ;ITEM 15
1681  001366  051540                          EM15                    ;AFTER CLEARING THE RH AND
1682                                                                    ;WRITING ONE WORD INTO
1683                                                                    ;RHDB AND READING IT
1684                                                                    ;BACK, CAUSED RHWC TO
1685                                                                    ;HAVE WRONG DATA
1686                                                                    ;GIVEN IN BAD RHWC
1687  001370  065515                          DH15                    ;PC
1688                                                                    ;TEST NUMBER
1689                                                                    ;RHWC GOOD
1690                                                                    ;RHWC BAD
1691  001372  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1692  001374  066745                          DF6                     ;0,0,0,0
1693
1694                                  ;ITEM 16
1695  001376  051732                          EM16                    ;AFTER CLEARING THE RH AND
1696                                                                    ;WRITING ONE WORD INTO
1697                                                                    ;RHDB
1698                                                                    ;CAUSED RHCS2 TO
1699                                                                    ;HAVE WRONG DATA
1700                                                                    ;GIVEN IN BAD RHCS2
1701  001400  065204                          DH6                     ;PC
1702                                                                    ;TEST NUMBER
1703                                                                    ;RHCS2 GOOD
1704                                                                    ;RHCS2 BAD
1705  001402  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1706  001404  066745                          DF6                     ;0,0,0,0
1707
1708                                  ;ITEM 17
1709  001406  052125                          EM17                    ;AFTER CLEARING THE RH AND
1710                                                                    ;WRITING TWO WORD INTO
1711                                                                    ;RHDB AND READING IT
1712                                                                    ;BACK, CAUSED RHDB TO
1713                                                                    ;HAVE WRONG DATA
1714                                                                    ;GIVEN IN BAD RHDB
1715  001410  065247                          DH7                     ;PC
1716                                                                    ;TEST NUMBER
1717                                                                    ;RHDB GOOD
1718                                                                    ;RHDB BAD
1719  001412  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1720  001414  066745                          DF6                     ;0,0,0,0
1721
1722                                  ;ITEM 20
1723  001416  052317                          EM20                    ;AFTER CLEARING THE RH AND
1724                                                                    ;WRITING TWO WORD INTO
1725                                                                    ;RHDB AND READING ONE
1726                                                                    ;BACK, CAUSED RHCS2 TO
1727                                                                    ;HAVE WRONG DATA
```

```
1728
1729  001420  065204                          DH6                     ;GIVEN IN BAD RHCS2
1730                                                                  ;PC
1731                                                                  ;TEST NUMBER
1732                                                                  ;RHCS2 GOOD
1733  001422  066576                          DT6                     ;RHCS2 BAD
1734  001424  066745                          DF6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1735                                                                  ;0,0,0,0
1736                                ;ITEM 21
1737  001426  052513                          EM21                    ;AFTER CLEARING THE RH AND
1738                                                                  ;WRITING TWO WORD INTO
1739                                                                  ;RHDB AND READING IT
1740                                                                  ;BACK TWICE, CAUSED RHDB TO
1741                                                                  ;HAVE WRONG DATA,
1742                                                                  ;GIVEN IN BAD RHDB
1743  001430  065247                          DH7                     ;PC
1744                                                                  ;TEST NUMBER
1745                                                                  ;RHDB GOOD
1746                                                                  ;RHDB BAD
1747  001432  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1748  001434  066745                          DF6                     ;0,0,0,0
1749
1750                                ;ITEM 22
1751  001436  052713                          EM22                    ;AFTER CLEARING THE RH AND
1752                                                                  ;WRITING TWO WORD INTO
1753                                                                  ;RHDB AND READING IT
1754                                                                  ;BACK TWICE, CAUSED RHCS2 TO
1755                                                                  ;HAVE WRONG DATA
1756                                                                  ;GIVEN IN BAD RHCS2
1757  001440  065204                          DH6                     ;PC
1758                                                                  ;TEST NUMBER
1759                                                                  ;RHCS2 GOOD
1760                                                                  ;RHCS2 BAD
1761  001442  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1762  001444  066745                          DF6                     ;0,0,0,0
1763
1764                                ;ITEM 23
1765  001446  053114                          EM23                    ;AFTER CLEARING THE RH AND
1766                                                                  ;WRITING EIGHT WORD INTO
1767                                                                  ;RHDB
1768                                                                  ;CAUSED RHCS2 TO
1769                                                                  ;HAVE WRONG DATA
1770                                                                  ;GIVEN IN BAD RHCS2
1771  001450  065204                          DH6                     ;PC
1772                                                                  ;TEST NUMBER
1773                                                                  ;RHCS2 GOOD
1774                                                                  ;RHCS2 BAD
1775  001452  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1776  001454  066745                          DF6                     ;0,0,0,0
1777
1778                                ;ITEM 24
1779  001456  053311                          EM24                    ;THE RH WAS CLEARED
1780                                                                  ;AND A PATTERN OF 8 WORDS
1781                                                                  ;WERE WRITTEN INTO RHDB
1782                                                                  ;READING RHDB FOR THE
1783                                                                  ;"N" TH. TIME GAVE WRONG
```

```
1784
1785
1786  001460  065555                          DH24                    ;PC
1787                                                                  ;TEST NO
1788                                                                  ;WORD NO
1789                                                                  ;RHDB GOOD
1790                                                                  ;RHDB BAD
1791  001462  066610                          DT24                    ;$ERRPC,TSTNM,SILONM,$GDDAT,$BDDAT
1792  001464  066751                          DF24                    ;0,0,0,0,0
1793
1794                                  ;ITEM 25
1795  001466  053541                          EM25                    ;THE RH WAS CLEARED AND
1796                                                                  ;A PATTERN OF 8 WORDS WERE
1797                                                                  ;WRITTEN INTO RHDB
1798                                                                  ;AFTER READING ALL 8 WORDS
1799                                                                  ;FOLLWING REGISTER CONTAINED WRONG
1800                                                                  ;VALUE
1801  001470  065204                          DH6                     ;PC
1802                                                                  ;TEST NO
1803                                                                  ;RHCS2 GOOD
1804                                                                  ;RHCS2 BAD
1805  001472  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1806  001474  066745                          DF6                     ;0,0,0,0
1807
1808                                  ;ITEM 26
1809  001476  053541                          EM25
1810  001500  065307                          DH11
1811  001502  066576                          DT6
1812  001504  066745                          DF6                     ;0,0
1813
1814                                  ;ITEM 27
1815  001506  053541                          EM25
1816  001510  065351                          DH12                    ;PC
1817                                                                  ;TEST NO
1818                                                                  ;RHCS3 GOOD
1819                                                                  ;RHCS3 BAD
1820  001512  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1821  001514  066745                          DF6                     ;0,0,0,0
1822
1823                                  ;ITEM 30
1824  001516  053541                          EM25
1825  001520  065413                          DH13                    ;PC
1826                                                                  ;TEST NO
1827                                                                  ;RHBA GOOD
1828                                                                  ;RHBA BAD
1829  001522  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1830  001524  066745                          DF6                     ;0,0,0,0
1831
1832                                  ;ITEM 31
1833  001526  053541                          EM25
1834  001530  065453                          DH14                    ;PC
1835                                                                  ;TEST NO
1836                                                                  ;RHBAE GOOD
1837                                                                  ;RHBAE BAD
1838  001532  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1839  001534  066745                          DF6                     ;0,0,0,0
```

```
                                ;ITEM 32
001536  053541                          EM25
001540  065515                          DH15                    ;PC
                                                                ;TEST NO
                                                                ;RHWC GOOD
                                                                ;RHWC BAD
001542  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001544  066745                          DF6                     ;0,0,0,0

                                ;ITEM 33
001546  053770                          EM33                    ;SETTING RH CLEAR BIT #5
                                                                ;IN RHCS2 CAUSED
                                                                ;ERROR REGISTER 1 TO HAVE
                                                                ;WRONG VALUE
001550  065625                          DH33                    ;PC
                                                                ;TEST NO
                                                                ;RHER1 GOOD
                                                                ;RHER1 BAD
001552  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001554  066745                          DF6

                                ;ITEM 34
001556  054104                          EM34                    ;AN RH LCLEAR WAS GIVEN
                                                                ;RHER1 WAS CHECKED TO HAVE ZERO
                                                                ;PAT (BIT #4 RHCS2) WAS SET
                                                                ;TO INVERT PARITY CHECKING
                                                                ;RHER1 WAS READ BUT DID NOT
                                                                ;CONTAIN WHAT IS IN RHER1 GOOD
001560  065625                          DH33                    ;PC
                                                                ;TEST NO
                                                                ;RHER1 GOOD
                                                                ;RHER1 BAD
001562  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001564  066745                          DF6                     ;0,0,0,0

                                ;ITEM 35
001566  054346                          EM35                    ;RH CLEAR WAS GIVEN
                                                                ;RHER1 WAS CHECKED
                                                                ;PAT (BIT #4 RHCS2) WAS SET
                                                                ;AND RHER1 WAS READ
                                                                ;RHCS1 SHOULD HAVE RDY
                                                                ;SC AND MCPE SET
001570  065307                          DH11                    ;PC
                                                                ;TEST NO
                                                                ;RHCS1 GOOD
                                                                ;RHCS1 BAD
001572  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001574  066745                          DF6                     ;0,0,0,0

                                ;ITEM 36
001576  054544                          EM36                    ;RH CLEAR WAS GIVEN
                                                                ;RHER1 WAS CHECKED
                                                                ;PAT (RHCS2-BIT #4) WAS SET
                                                                ;RHER1 WAS READ AND CHECKED
                                                                ;"1" WAS WRITTEN INTO "TRE"-RHCS1
```

```
1896
1897
1898
1899  001600  065307                          DH11            ;ON CHECKING RHCS1
                                                              ;IT DID NOT CONTAIN SC,MCPE
                                                              ;AND RDY
                                                              ;PC
1900                                                          ;TEST NUMBER
1901                                                          ;RHCS1 GOOD
1902                                                          ;RHCS1 BAD
1903  001602  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1904  001604  066745                          DF6             ;0,0,0,0
1905
1906                                  ;ITEM 37
1907  001606  055021                          EM37            ;RH WAS CLEARED
1908                                                          ;RHER1 WAS CHECKED
1909                                                          ;PAT (RHCS2-BIT #4) WAS SET
1910                                                          ;RHER1 WAS READ
1911                                                          ;"1" WAS WRITTEN IN "TRE" RHCS1
1912                                                          ;AN RH CLEAR WAS TO
1913                                                          ;GIVE WHAT IS IN "GOOD"
1914                                                          ;BUT GAVE WHAT IS IN "BAD"
1915  001610  065307                          DH11            ;PC
1916                                                          ;TEST NO
1917                                                          ;RHCS1 GOOD
1918                                                          ;RHCS1 BAD
1919  001612  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1920  001614  066745                          DF6             ;0,0,0,0
1921
1922                                  ;ITEM 40
1923  001616  055021                          EM37
1924  001620  065204                          DH6
1925  001622  066576                          DT6
1926  001624  066745                          DF6
1927
1928                                  ;ITEM 41
1929  001626  055021                          EM37
1930  001630  065351                          DH12
1931  001632  066576                          DT6
1932  001634  066745                          DF6
1933
1934                                  ;ITEM 42
1935  001636  055021                          EM37
1936  001640  065413                          DH13
1937  001642  066576                          DT6
1938  001644  066745                          DF6
1939
1940                                  ;ITEM 43
1941  001646  055021                          EM37
1942  001650  065453                          DH14
1943  001652  066576                          DT6
1944  001654  066745                          DF6
1945
1946                                  ;ITEM 44
1947  001656  055021                          EM37
1948  001660  065515                          DH15
1949  001662  066576                          DT6
1950  001664  066745                          DF6
1951
```

```
                                ;ITEM 45
001666  055323                          EM45            ;AFTER AN RH CLEAR
                                                        ;"1" WAS WRITTEN IN THE DISK
                                                        ;ADDRESS REGISTER (IN TAPE
                                                        ;DRIVES CALLED FRAME COUNT )
                                                        ;PAT IN RHCS1 WAS SET
                                                        ;ON READING RHDA (IN TAPE
                                                        ;DRIVES RHFC ) IT DID NOT
                                                        ;CONTAIN "1"
                                                        ;RHDA=RHFC
001670  065667                          DH45            ;PC
                                                        ;TEST NO
                                                        ;RHDA GOOD
                                                        ;RHDA BAD
001672  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001674  066745                          DF6             ;0,0,0,0

                                ;ITEM 46
001676  055561                          EM46            ;AFTER AN RH CLEAR
                                                        ;"1" WAS WRITTEN IN THE
                                                        ;DISK ADDRESS REGISTER (IN
                                                        ;TAPE
                                                        ;PAT IN RHCS1 WAS SET
                                                        ;ON READING RHDA (IN TAPE
                                                        ; ) FOLLOWING
                                                        ;REGISTER DID NOT CONTAIN
                                                        ;WHAT IS IN "GOOD"
001700  065307                          DH11            ;PC
                                                        ;TEST NO
                                                        ;RHCS1 GOOD
                                                        ;RHCS1 BAD
001702  066576                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001704  066745                          DF6

                                ;ITEM 47
001706  055561                          EM46
001710  065204                          DH6
001712  066576                          DT6
001714  066745                          DF6

                                ;ITEM 50
001716  055561                          EM46
001720  065351                          DH12
001722  066576                          DT6
001724  066745                          DF6

                                ;ITEM 51
001726  055561                          EM46
001730  065413                          DH13
001732  066576                          DT6
001734  066745                          DF6

                                ;ITEM 52
001736  055561                          EM46
001740  065453                          DH14
001742  066576                          DT6
```

```
 2008  001744  066745                          DF6
 2009
 2010                                  ;ITEM 53
 2011  001746  055561                          EM46
 2012  001750  065515                          DH15
 2013  001752  066576                          DT6
 2014  001754  066745                          DF6
 2015
 2016                                  ;ITEM 54
 2017  001756  056062                          EM54          ;AN RH CLEAR (RHCS2 BIT #5)
 2018                                                        ;WAS GIVEN
 2019                                                        ;A16 (RHCS1 BIT #8) WAS SET
 2020                                                        ;ON READING THE FOLLOWING
 2021                                                        ;REGISTER IT DID NOT
 2022                                                        ;CONTAIN WHAT IS IN "GOOD"
 2023  001760  065307                          DH11
 2024  001762  066576                          DT6
 2025  001764  066745                          DF6
 2026
 2027                                  ;ITEM 55
 2028  001766  056062                          EM54
 2029  001770  065453                          DH14
 2030  001772  066576                          DT6
 2031  001774  066745                          DF6
 2032
 2033                                  ;ITEM 56
 2034  001776  056273                          EM56          ;AN RH CLEAR (RHCS2 BIT #5)
 2035                                                        ;WAS GIVEN
 2036                                                        ;A16 WAS WRITTEN IN RHCS1
 2037                                                        ;ALL ZEROS WERE WRITTEN IN RHBAE
 2038                                                        ;THE FOLLOWING REGISTER DID
 2039                                                        ;NOT CONTAIN WHAT IS IN "GOOD"
 2040  002000  065453                          DH14
 2041  002002  066576                          DT6
 2042  002004  066745                          DF6
 2043
 2044                                  ;ITEM 57
 2045  002006  056273                          EM56
 2046  002010  065307                          DH11
 2047  002012  066576                          DT6
 2048  002014  066745                          DF6
 2049
 2050                                  ;ITEM 60
 2051  002016  056531                          EM60          ;AN RH CLEAR (RHCS2 BIT #5)
 2052                                                        ;WAS GIVEN
 2053                                                        ;A17 (RHCS1 BIT #9) WAS SET
 2054                                                        ;ON READING THE FOLLOWING
 2055                                                        ;REGISTER IT DID NOT
 2056                                                        ;CONTAIN WHAT IS IN "GOOD"
 2057  002020  065307                          DH11
 2058  002022  066576                          DT6
 2059  002024  066745                          DF6
 2060
 2061                                  ;ITEM 61
 2062  002026  056531                          EM60
 2063  002030  065453                          DH14
```

```
2064  002032  066576                    DT6
2065  002034  066745                    DF6
2066
2067                              ;ITEM 62
2068  002036  056742                    EM62          ;AN RH CLEAR (RHCS2 BIT #5)
2069                                                   ;WAS GIVEN
2070                                                   ;A17 WAS WRITTEN IN RHCS1
2071                                                   ;ALL ZEROS WERE WRITTEN IN RHBAE
2072                                                   ;THE FOLLOWING REGISTER DID
2073                                                   ;NOT CONTAIN WHAT IS IN "GOOD"
2074  002040  065453                    DH14
2075  002042  066576                    DT6
2076  002044  066745                    DF6
2077
2078                              ;ITEM 63
2079  002046  056742                    EM62
2080  002050  065307                    DH11
2081  002052  066576                    DT6
2082  002054  066745                    DF6
2083
2084                              ;ITEM 64
2085  002056  057177                    EM64          ;AN RH CLEAR (RHCS2 BIT #5)
2086                                                   ;WAS GIVEN
2087                                                   ;IE (RHCS1 BIT #6) WAS SET
2088                                                   ;ON READING THE FOLLOWING
2089                                                   ;REGISTER IT DID NOT
2090                                                   ;CONTAIN WHAT IS IN "GOOD"
2091  002060  065307                    DH11
2092  002062  066576                    DT6
2093  002064  066745                    DF6
2094
2095                              ;ITEM 65
2096  002066  057177                    EM64
2097  002070  065351                    DH12
2098  002072  066576                    DT6
2099  002074  066745                    DF6
2100
2101                              ;ITEM 66
2102  002076  057177                    EM64          ;AN RH CLEAR (RHCS2 BIT #5)
2103                                                   ;WAS GIVEN
2104                                                   ;A16 WAS WRITTEN IN RHCS1
2105                                                   ;ALL ZEROS WERE WRITTEN IN RHBAE
2106                                                   ;THE FOLLOWING REGISTER DID
2107                                                   ;NOT CONTAIN WHAT IS IN "GOOD"
2108  002100  065351                    DH12
2109  002102  066576                    DT6
2110  002104  066745                    DF6
2111
2112                              ;ITEM 67
2113  002106  057177                    EM64
2114  002110  065307                    DH11
2115  002112  066576                    DT6
2116  002114  066745                    DF6
2117
2118                              ;ITEM 70
2119  002116  057646                    EM70          ;RH CLEAR WAS GIVEN
```

```
                                                        ;TWO SUCCESSIVE "GO" (RHCS1 BIT #0)
                                                        ;WAS GIVEN WITHOUT GIVING
                                                        ;TIME FOR FIRST "GO" TO
                                                        ;COMPLETE
                                                        ;THE FOLLOWING REGISTER
                                                        ;DID NOT CONTAIN WHAT IS
                                                        ;IN "GOOD"
002120  065307                          DH11
002122  066576                          DT6
002124  066745                          DF6

                                ;ITEM 71
002126  057646                          EM70
002130  065204                          DH6
002132  066576                          DT6
002134  066745                          DF6

                                ;ITEM 72
002136  057646                          EM70
002140  065351                          DH12
002142  066576                          DT6
002144  066745                          DF6

                                ;ITEM 73
002146  057646                          EM70
002150  065413                          DH13
002152  066576                          DT6
002154  066745                          DF6

                                ;ITEM 74
002156  057646                          EM70
002160  065453                          DH14
002162  066576                          DT6
002164  066745                          DF6

                                ;ITEM 75
002166  057646                          EM70
002170  065515                          DH15
002172  066576                          DT6
002174  066745                          DF6

                                ;ITEM 76
002176  060122                          EM76            ;RH CLEAR WAS GIVEN A 2 WORD
                                                        ;WRITE WAS DONE FROM A
                                                        ;LOCATION TAGED WRFROM
                                                        ;AND WITH BAI BIT SET
                                                        ;AT THE END OF THE WRITE
                                                        ;THE FOLLOWING REGISTER DID
                                                        ;NOT CONTAIN WHAT IS
                                                        ;IN GOOD
002200  065307                          DH11
002202  066576                          DT6
002204  066745                          DF6

                                ;ITEM 77
002206  060122                          EM76
```

```
2176  002210  065204                          DH6
2177  002212  066576                          DT6
2178  002214  066745                          DF6
2179
2180                                  ;ITEM 100
2181  002216  060122                          EM76
2182  002220  065351                          DH12
2183  002222  066576                          DT6
2184  002224  066745                          DF6
2185
2186                                  ;ITEM 101
2187  002226  060122                          EM76
2188  002230  065413                          DH13
2189  002232  066576                          DT6
2190  002234  066745                          DF6
2191
2192                                  ;ITEM 102
2193  002236  060122                          EM76
2194  002240  065453                          DH14
2195  002242  066576                          DT6
2196  002244  066745                          DF6
2197
2198                                  ;ITEM 103
2199  002246  060122                          EM76
2200  002250  065515                          DH15
2201  002252  066576                          DT6
2202  002254  066745                          DF6
2203
2204                                  ;ITEM 104
2205  002256  060415                          EM104                ;RH CLEAR WAS GIVEN AN
2206                                                               ;IPCK BIT SHOWN IN "IPCK" WAS SET
2207                                                               ;ZEROS WERE MOVED INTO RHDB
2208                                                               ;ON READING RHDB
2209                                                               ;THE FOLLOWING REGISTER
2210                                                               ;DID NOT CONTAIN WHAT
2211                                                               ;IS IN GOOD
2212  002260  065727                          DH104                ;PC,TEST NO,IPCK,RHCS3 GOOD,RHCS3 BAD
2213  002262  066624                          DT104                ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2214  002264  066756                          DF104                ;0,0,0,0,0
2215
2216                                  ;ITEM 105
2217  002266  060415                          EM104
2218  002270  065777                          DH105                ;PC,TST NO, IPCK,RHCS2 GOOD,RHCS2 BAD
2219  002272  066640                          DT105                ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2220  002274  066763                          DF105                ;0,0,0,0,0
2221
2222                                  ;ITEM 106
2223  002276  060653                          EM106                ;RH CLEAR WAS GIVEN AN
2224                                                               ;IPCK BIT SHOWN IN "IPCK" WAS SET
2225                                                               ;ZEROS WERE MOVED INTO
2226                                                               ;RHDB FROM AN ODD WORD
2227                                                               ;RHDB WAS READ
2228                                                               ;AN RH CLEAR WAS GIVEN
2229                                                               ;TO CLEAR ALL ERRORS
2230                                                               ;THE FOLLLOWING REGISTER
2231                                                               ;DID NOT CONTAIN WHAT
```

```
                                                        ;IS IN GOOD
002300  066047                    DH106                 ;PC,TSTNM,IPCK,RHCS1 GOOD,RHCS1 BAD
002302  066654                    DT106                 ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
002304  066770                    DF106                 ;0,0,0,0,0

                                ;ITEM 107
002306  060653                    EM106
002310  065777                    DH105                 ;PC,TSTNM,IPCK,RHCS2 GOOD,RHCS2 BAD
002312  066640                    DT105
002314  066763                    DF105

                                ;ITEM 110
002316  060653                    EM106
002320  065727                    DH104
002322  066624                    DT104
002324  066756                    DF104

                                ;ITEM 111
002326  060653                    EM106
002330  066117                    DH111                 ;PC,TSTNM,IPCK,RHBA GOOD,RHBA BAD
002332  066624                    DT104                 ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
002334  066756                    DF104                 ;0,0,0,0,0

                                ;ITEM 112
002336  060653                    EM106
002340  066165                    DH112                 ;PC,TSTNM,IPCK,RHBAE GOOD,RHBAE BAD
002342  066624                    DT104                 ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
002344  066756                    DF104                 ;0,0,0,0,0

                                ;ITEM 113
002346  060653                    EM106
002350  066235                    DH113                 ;PC,TSTNM,IPCK,RHWC GOOD,RHWC BAD
002352  066624                    DT104                 ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
002354  066756                    DF104                 ;0,0,0,0,0

                                ;ITEM 114
002356  061144                    EM114                 ;RH CLEAR WAS GIVEN AN
                                                        ;A WRITE CHECK WAS DONE
                                                        ;THE FOLLOWING REGISTER
                                                        ;DID NOT CONTAIN WHAT IS
                                                        ;IN "GOOD"
002360  065204                    DH6
002362  066576                    DT6
002364  066745                    DF6

                                ;ITEM 115
002366  061144                    EM114                 ;RH CLEAR WAS GIVEN AN
                                                        ;A WRITE CHECK WAS DONE
                                                        ;THE FOLLOWING REGISTER
                                                        ;DID NOT CONTAIN WHAT IS
                                                        ;IN "GOOD"
002370  065351                    DH12
002372  066576                    DT6
002374  066745                    DF6

                                ;ITEM 116
```

```
2288  002376  061313                        EM116           ;RH CLEAR WAS GIVEN AN
2289                                                        ;A WRITE CHECK WAS DONE
2290                                                        ;THEN ANOTHER RH CLEAR WAS
2291                                                        ;GIVEN TO CLEAR ALL ERRORS
2292                                                        ;THE FOLLOWING REGISTER DID
2293                                                        ;NOT CONTAIN WHAT IS IN "GOOD"
2294  002400  065307                        DH11
2295  002402  066576                        DT6
2296  002404  066745                        DF6
2297
2298                                ;ITEM 117
2299  002406  061313                        EM116
2300  002410  065204                        DH6
2301  002412  066576                        DT6
2302  002414  066745                        DF6
2303
2304                                ;ITEM 120
2305  002416  061313                        EM116
2306  002420  065351                        DH12
2307  002422  066576                        DT6
2308  002424  066745                        DF6
2309
2310                                ;ITEM 121
2311  002426  061313                        EM116
2312  002430  065413                        DH13
2313  002432  066576                        DT6
2314  002434  066745                        DF6
2315
2316                                ;ITEM 122
2317  002436  061313                        EM116
2318  002440  065453                        DH14
2319  002442  066576                        DT6
2320  002444  066745                        DF6
2321
2322                                ;ITEM 123
2323  002446  061313                        EM116
2324  002450  065515                        DH15
2325  002452  066576                        DT6
2326  002454  066745                        DF6
2327
2328                                ;ITEM 124
2329  002456  061523                        EM124           ;ON A SILO TEST TO
2330                                                        ;TEST DBL RHCS3-BIT #10
2331                                                        ;THE FOLLOWING REGISTER
2332                                                        ;DID NOT CONTAIN WHAT
2333                                                        ;IS IN "GOOD"
2334  002460  065351                        DH12
2335  002462  066576                        DT6
2336  002464  066745                        DF6
2337
2338                                ;ITEM 125
2339  002466  061523                        EM124
2340  002470  065307                        DH11
2341  002472  066576                        DT6
2342  002474  066745                        DF6
2343
```

```
                                        ;ITEM 126
002476  061523                                EM124
002500  065204                                DH6
002502  066576                                DT6
002504  066745                                DF6

                                        ;ITEM 127
002506  061523                                EM124
002510  065413                                DH13
002512  066576                                DT6
002514  066745                                DF6

                                        ;ITEM 130
002516  061523                                EM124
002520  065[illegible]                        DH14
002522  066[illegible]                        DT6
002524  066[illegible]                        DF6

                                        ;ITEM 131
002526  061523                                EM124
002530  065515                                DH15
002532  066576                                DT6
002534  066745                                DF6
                                        ;ITEM132
002536  061666                                EM132           ;BEFORE DATA TRANSFER
                                                            ;COMMAND WAS TO BE GIVEN
                                                            ;THE RP DRIVE
                                                            ;STATUS REGISTER DID NOT
                                                            ;CONTAIN WHAT IS IN GOOD

002540  066303                                DH132           ;PC
                                                            ;TEST NO.
                                                            ;RHDS1 GOOD
                                                            ;RHDS1 BAD

002542  066576                                DT6
002544  066745                                DF6
                                        ;ITEM133
002546  062044                                EM133           ;BEFORE DATA TRANSFER
                                                            ;COMMAND WAS TO BE GIVEN
                                                            ;THE RS DRIVE
                                                            ;STATUS REGISTER DID NOT
                                                            ;CONTAIN WHAT IS IN GOOD

002550  066303                                DH132           ;PC
                                                            ;TEST NO.
                                                            ;RHDS1 GOOD
                                                            ;RHDS1 BAD

002552  066576                                DT6
002554  066745                                DF6
                                        ;ITEM 134
002556  062222                                EM134           ;BEFORE DATA TRANSFER COMMAND
                                                            ;WAS TO BE GIVEN THE
                                                            ;MAG TAPE DRIVE STATUS
                                                            ;REGISTER DID NOT CONTAIN WHAT
                                                            ;IS IN GOOD
```

```
2400  002560  066303                    DH132
2401  002562  066576                    DT6
2402  002564  066745                    DF6
2403                              ;ITEM135
2404  002566  062412                    EM135          ;WAS WATIING FOR A BIT TO SET
2405                                                    ;BIT IN QUESTION IS IN "BIT WAITED FOR"
2406                                                    ;REGISTER IN QUESTION IN "REG ADDR"
2407  002570  066345                    DH135          ;PC
2408                                                    ;TEST NO
2409                                                    ;PC OF WAT
2410                                                    ;BIT
2411                                                    ;REG ADDR
2412  002572  066670                    DT135          ;$ERRPC,TSTNM,WATIPC,WAITBT,WAITRE
2413  002574  066775                    DF135          ;0,0,0,0,0
2414                              ;ITEM136
2415  002576  062612                    EM136          ;WAS WATIING FOR A BIT TO RESET
2416                                                    ;BIT IN QUESTION IS IN "BIT WAITED FOR"
2417                                                    ;REGISTER IN QUESTION IN "REG ADDR"
2418  002600  066345                    DH135          ;PC
2419                                                    ;TEST NO
2420                                                    ;PC OF WAT
2421                                                    ;BIT
2422                                                    ;REG ADDR
2423  002602  066670                    DT135          ;$ERRPC,TSTNM,WATIPC,WAITBT,WAITRE
2424  002604  066775                    DF135          ;0,0,0,0,0
2425                              ;ITEM 137
2426  002606  063014                    EM137          ;ON WRITING AND READING
2427                                                    ;THE REGISTER # IN "REG. ADDR"
2428                                                    ;IT DID NOT CONTAIN
2429                                                    ;EXPECTED VALUE.
2430  002610  066425                    DH137          ;PC
2431                                                    ;TEST NO
2432                                                    ;REG ADDR
2433                                                    ;GOOD DATA
2434                                                    ;BAD DATA
2435  002612  066704                    DT137          ;$ERRPC,TSTNM,$BDADR,$GDDAT,$BDDAT
2436  002614  067002                    DF137          ;0,0,0,0,0
2437
2438
2439                              ;ITEM 140
2440  002616  063142                    EM140
2441  002620  066047                    DH106
2442  002622  066654                    DT106
2443  002624  066770                    DF106
2444                              ;ITEM 141
2445  002626  063374                    EM141          ;AFTER CLEARING THE RH AND
2446                                                    ;WRITING TWO WORD INTO
2447                                                    ;RHDB AND READING IT
2448                                                    ;BACK TWICE, CAUSED RHCS1 TO
2449                                                    ;HAVE WRONG DATA,
2450                                                    ;GIVEN IN BAD RHCS1
2451  002630  065307                    DH11           ;PC
2452                                                    ;TEST NUMBER
2453                                                    ;RHCS1 GOOD
2454                                                    ;RHCS1 BAD
2455  002632  066576                    DT6            ;$ERRPC,TSTNM,$GDDAT,$BDDAT
```

```
2456  002634  066745                          DF6                     ;0,0,0,0
2457
2458
2459                                  ;ITEM 142
2460  002636  063575                          EM142                   ;AFTER CLEARING THE RH AND
2461                                                                   ;WRITING TWO WORD INTO
2462                                                                   ;RHDB AND READING IT
2463                                                                   ;BACK TWICE, CAUSED RHCS3 TO
2464                                                                   ;HAVE WRONG DATA,
2465                                                                   ;GIVEN IN BAD RHCS3
2466  002640  065351                          DH12                    ;PC
2467                                                                   ;TEST NUMBER
2468                                                                   ;RHCS3 GOOD
2469                                                                   ;RHCS3 BAD
2470  002642  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2471  002644  066745                          DF6                     ;0,0,0,0
2472
2473
2474                                  ;ITEM 143
2475  002646  063776                          EM143                   ;AFTER CLEARING THE RH AND
2476                                                                   ;WRITING TWO WORD INTO
2477                                                                   ;RHDB AND READING IT
2478                                                                   ;BACK TWICE, CAUSED RHBA TO
2479                                                                   ;HAVE WRONG DATA,
2480                                                                   ;GIVEN IN BAD RHBA
2481  002650  065413                          DH13                    ;PC
2482                                                                   ;TEST NUMBER
2483                                                                   ;RHBA GOOD
2484                                                                   ;RHBA BAD
2485  002652  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2486  002654  066745                          DF6                     ;0,0,0,0
2487
2488
2489                                  ;ITEM 144
2490  002656  064176                          EM144                   ;AFTER CLEARING THE RH AND
2491                                                                   ;WRITING TWO WORD INTO
2492                                                                   ;RHDB AND READING IT
2493                                                                   ;BACK TWICE, CAUSED RHBAE TO
2494                                                                   ;HAVE WRONG DATA,
2495                                                                   ;GIVEN IN BAD RHBAE
2496  002660  065453                          DH14                    ;PC
2497                                                                   ;TEST NUMBER
2498                                                                   ;RHBAE GOOD
2499                                                                   ;RHBAE BAD
2500  002662  066576                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2501  002664  066745                          DF6                     ;0,0,0,0
2502
2503
2504                                  ;ITEM 145
2505  002666  064377                          EM145                   ;AFTER CLEARING THE RH AND
2506                                                                   ;WRITING TWO WORD INTO
2507                                                                   ;RHDB AND READING IT
2508                                                                   ;BACK TWICE, CAUSED RHWC TO
2509                                                                   ;HAVE WRONG DATA,
2510                                                                   ;GIVEN IN BAD RHWC
2511  002670  065515                          DH15                    ;PC
```

```
2512                                                            ;TEST NUMBER
2513                                                            ;RHWC GOOD
2514                                                            ;RHWC BAD
2515  002672  066576                    DT6                    ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2516  002674  066745                    DF6                    ;0,0,0,0
2517                            ;ITEM146
2518  002676  064577                    EM146                  ;A DEVICE BASE ADDRESS DID NOT
2519                                                            ;TIME OUT BUT CORRESPONDING
2520                                                            ;VECTOR ADDRESS DID TIME OUT
2521  002700  066477                    DH146                  ;PC
2522                                                            ;BASE
2523                                                            ;VECTOR
2524  002702  066720                    DT146                  ;$ERRPC,TESTDV,TESTVC,0
2525  002704  067007                    DF146                  ;0,0,0
2526
2527                            ;ITEM147
2528  002706  064771                    EM147                  ;A DEVICE ADDRESS DID NOT
2529                                                            ;TIME OUT BUT NO UNITS HAD
2530                                                            ;APPROPRIATE DRIVE TYPE
2531  002710  066526                    DH147                  ;PC
2532                                                            ;BASE ADDRESS
2533  002712  066730                    DT147                  ;$ERRPC,TESTDV
2534  002714  067012                    DF147
2535
```

```
                                ;;*****************************************************************
                                ;RH70 REGISTERS


002716  000254                  RPVEC:  254                             ;RP VECTOR ADDRESS
                                ;;*****************************************************************
                                ;WORD COUNT REGISTER (RHWC)
                                ;EACH BIT IS CALLED BY BIT NUMBER


                                ;BUS ADDRESS REGISTER (RHBA)
                                ;EACH BIT IS CALLED BY BIT NUMBER


                                ;CONTROL AND STATUS REGISTER 2 (RHCS2)

000001                          US1=    1                               ;UNIT SELECT (BIT #0)
000002                          US2=    2                               ;UNIT SELECT (BIT #1)
000004                          US4=    4                               ;UNIT SELECT (BIT #2)
000010                          BAI=    10                              ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
000020                          PAT=    20                              ;INVERT PARITY
000040                          CLR=    40                              ;CLEAR (BIT #5)
000100                          IR=     100                             ;INPUT READY (BIT #6)
000200                          OR=     200                             ;OUTPUT READY (BIT #7)
000400                          MPE=    400                             ;MASS BUS PARITY ERROR (BIT #8)
000400                          MDPE=   400                             ;MASS BUS PARITY ERROR (BIT #8)
001000                          MXF=    1000                            ;MISSED TRANSFER ERROR (BIT #9)
002000                          PGE=    2000                            ;PROGRAM ERROR (BIT #10)
004000                          NEM=    4000                            ;NON EXISTANT MEMORY (BIT #11)
010000                          NED=    10000                           ;NON EXISTANT DRIVE (BIT #12)
020000                          PE=     20000                           ;UNIBUS PARITY ERROR (BIT #13)
040000                          WCE=    40000                           ;WRITE CHECK ERROR (BIT #14)
100000                          DLT=    100000                          ;DATA LATE (BIT #15)

                                ;CONTROL AND STATUS REGISTER 3 (RHCS3)

000001                          IPCK0=  1                               ;INVERT PARITY CHECK BIT 0
000002                          IPCK1=  2                               ;INVERT PARITY CHECK BIT 1
000004                          IPCK2=  4                               ;INVERT PARITY CHECK BIT 2
000010                          IPCK3=  10                              ;INVERT PARITY CHECK BIT 3
000100                          IE=     100                             ;INTERRUPT ENABLE (BIT #6)
002000                          DBL=    2000                            ;DOUBLE WORD BOUNDRY (BIT #10)
004000                          WCEEW=  4000                            ;WRITE CHECK EVEN WORD (BIT #11)
010000                          WCEOW=  10000                           ;WRITE CHECK ODD WORD (BIT #12)
020000                          DPEEW=  20000           ;DATA PARITY ERROR EVEN WORD (BIT #13)
040000                          DPEOW=  40000           ;DATA PARITY ERROR ODD WORD (BIT #14)
100000                          APE=    100000          ;ADDRESS PARITY ERROR (BIT #15)


                                ;DATA BUFFER REGISTER (RHDB)
```

```
                                ;EACH BIT IS CALLED BY BIT NUMBER


                                ;;*****************************************************************
                                ;RP04 REGISTERS
                                ;;*****************************************************************


                                ;CONTROL AND STATUS 1 REGISTER. (#00)

000001                          GO=     1               ;GO (BIT #0)
000100                          IE=     100             ;INTERRUPT ENABLE (BIT #6)
000200                          RDY=    200             ;READY (BIT #7)
000400                          A16=    400             ;HIGH ORDER UNIBUS BITS (BIT #8)
001000                          A17=    1000            ;HIGH ORDER UNIBUS BITS (BIT #9)
002000                          PSEL=   2000            ;PORT SELECT (BIT #10)
004000                          DVA=    4000            ;DEVICE AVAILABLE (BIT #11)
020000                          MCPE=   20000           ;MASSBUSS PARITY ERROR (BIT #13)
040000                          TRE=    40000           ;TRANSFER ERROR (BIT #14)
100000                          SC=     100000          ;SPECIAL CONDITION (BIT #15)

                                ;STATUS REGISTER (RHDS1) (#01)

000001                          DFF5=   1               ;DRIVE FORWARD 5"/SEC.  (BIT #0)
000002                          BOT=    2               ;BEGINING OF TAPE (BIT #1)
000002                          DFF20=  2               ;DRIVE FORWARD 20"/SEC.  (BIT #1)
000004                          DIGB=   4               ;DRIVE TO INNER GAVRD BAND (BIT #2)
000010                          GRV=    10              ;GO REVERSE (BIT #3)
000020                          DL64=   20              ;DIFFERENCE LESS THAN 64 (BIT #4)
000040                          DE1=    40              ;DIFFERENCE EQUALS 1 (BIT #5)
000100                          VV=     100             ;VOLUME VALID (BIT #6)
000200                          DRY=    200             ;DRIVE READY (BIT #7)
000400                          DPR=    400             ;DRIVE PRESENT (BIT #8)
001000                          PROG=   1000            ;PROGRAMABLE (BIT #9)
002000                          LBT=    2000            ;LAST SECTOR TRANSFERRED (BIT #10)
004000                          WRL=    4000            ;WRITE LOCK (BIT #11)
010000                          MOL=    10000           ;MEDIUM ON-LINE (BIT #12)
020000                          PIP=    20000           ;POSITIONING OPERATION IN PROGRESS (BIT #13)
040000                          ERR=    40000           ;COMPOSIT ERROR. (BIT #14)
100000                          ATA=    100000          ;ATTENTION ACTIVE (BIT #15)

                                ;ERROR REGISTER #01 (RHER1) (#02)
000001                          ILF=    1               ;ILLEGAL FUNCTION (BIT #0)
000002                          ILR=    2               ;ILLEGAL REGISTER (BIT #1)
000004                          RMR=    4               ;REGISTER MODIFICATION REFUSED (BIT #2)
000010                          PAR=    10              ;PARITY ERROR (BIT #3)
000020                          FER=    20              ;FORMAT ERROR (BIT #4)
000040                          WCF=    40              ;WRITE CLOCK FAIL (BIT #5)
000100                          ECH=    100             ;ECC HARD ERROR (BIT #6)
000200                          HCE=    200             ;HEADER COMPARE ERROR (BIT #7)
000400                          HCRC=   400             ;HEADER CRC ERROR (BIT #8)
001000                          AOE=    1000            ;ADDRESS OVERFLOW ERROR (BIT #9)
002000                          IAE=    2000            ;INVALID ADDRESS ERROR (BIT #10)
004000                          WLE=    4000            ;WRITE LOCK ERROR (BIT #11)
010000                          DTE=    10000           ;DRIVE TIMING ERROR (BIT #12)
```

```
                020000          OPI=    20000           ;OPERATION INCOMPLETE (BIT #13)
                040000          UNS=    40000           ;DRIVE UNSAFE (BIT #14)
                100000          DCK=    100000          ;DATA CHECK ERROR (BIT 15)

                                ;MAINTAINABILITY REGISTER (RHMR)(#03)

                000001          DMD=    1               ;DIAGINOSTIC MODE (BIT #0)
                000002          MCLK=   2               ;MAINTAINABILITY CLOCK (BIT #1)
                000004          MINX=   4               ;MAINTAINABILITY INDEX (BIT #2)
                000010          MSTCK=  10              ;MAINTAINABILITY SECTOR CLOCK (BIT #3)
                000020          MRD=    20              ;MAINTAINABILITY READ (BIT #4)
                000040          MWR=    40              ;MAINTAINABILITY WRITE (BIT #5)
                001000          DTSY=   1000            ;MAINTAINABILITY SYNC DETECTED (BIT #9)

                                ;ATTENTION SUMMARY PSEUDO-REGISTER (RHAS) (#04)

                000001          AT0=    1               ;DEVICE 0 (BIT #0)
                000002          AT1=    2               ;DEVICE 1 (BIT #1)
                000004          AT2=    4               ;DEVICE 2 (BIT #2)
                000010          AT3=    10              ;DEVICE 3 (BIT #3)
                000020          AT4=    20              ;DEVICE 4 (BIT #4)
                000040          AT5=    40              ;DEVICE 5 (BIT #5)
                000100          AT6=    100             ;DEVICE 6 (BIT #6)
                000200          AT7=    200             ;DEVICE 7 (BIT #7)

                                ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RHDST) (#1)
                                ;EACH BIT IS CALLED BY BIT NUMBER
                                ;DRIVE TYPE REGISTER (RHDT) (#06)
                                ;EACH BIT IS CALLED BY BIT NUMBER
                                ;LOOK-AHEAD REGISTER (RHLA) (#07)

        000001                  EXT1=   1               ;EXTENSION 1 (BIT #0)
        000002                  EXT2=   2               ;EXTENSION 2 (BIT #1)
        000004                  EXT4=   4               ;EXTENSION 3 (BIT #2)
        000010                  EXT10=  10              ;EXTENSION 4 (BIT #3)
        000020                  EXT20=  20              ;EXTENSION 5 (BIT #4)
        000040                  EXT40=  40              ;EXTENSION 6 (BIT #5)
        000100                  SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
        000200                  SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
        000400                  SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
        001000                  SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
        002000                  SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
        004000                  TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
        010000                  TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
        020000                  TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
        040000                  TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
        100000                  TRK20=  100000          ;TRACK FIELD 5 (BIT #15)

                                ;ERROR REGISTER #2 (RHER2) (#10)

        000001                  WCU=    1               ;WRITE CURRENT UNSAFE (BIT #0)
        000002                  CSF=    2               ;CURRENT SINK FAILURE (BIT #1)
        000004                  WSU=    4               ;WRITE SELECT UNSAFE (BIT #2)
        000010                  CSU=    10              ;CURRENT SWITCH UNSAFE (BIT #3)
        000020                  MSE=    20              ;MOTOR SEQUENCE ERROR (BIT #4)
        000040                  TDF=    40              ;TRANSITIONS DETECTOR FAILURE (BIT #5)
```

```
2704            000100                  TUF=    100                     ;TRANSITIONS UNSAFE (BIT #6)
2705            000200                  FEN=    200                     ;FAILSAFE ENABLED (BIT #7)
2706            000400                  WRU=    400                     ;WRITE READY UNSAFE (BIT #8)
2707            001000                  MHS=    1000                    ;MULTIPLE HEAD SELECT (BIT #9)
2708            002000                  NHS=    2000                    ;NO HEAD SELECTION (BIT #10)
2709            004000                  IXE=    4000                    ;INDEX ERROR (BIT #11)
2710            010000                  VU30=   10000                   ;30VOLT UNSAFE (BIT #12)
2711            020000                  PLU=    20000                   ;PLO UNSAFE (BIT #13)
2712            100000                  ACU=    100000                  ;ACUNSAFE (BIT #15)
2713
2714                                    ;OFFSET REGISTER (RHOF) (#11)
2715
2716            000001                  OF25=   1                       ;OFFSET 25 MICRO INCHES (BIT #0)
2717            000002                  OF50=   2                       ;OFFSET 50 MICRO INCHES (BIT #1)
2718            000004                  OF100=  4                       ;OFFSET 100 MICRO INCHES (BIT #2)
2719            000010                  OF200=  10                      ;OFFSET 200 MICRO INCHES (BIT #3)
2720            000020                  OF400=  20                      ;OFFSET 400 MICRO INCHES (BIT #4)
2721            000040                  OF800=  40                      ;OFFSET 800 MICRO INCHES (BIT #5)
2722
2723            000200                  OFREV=  200                     ;OFFSET NEGATIVE (REVERSE) (BIT #5)
2724            002000                  HCI=    2000                    ;HEADER COMPARE INHIBIT (BIT #10)
2725            004000                  ECI=    4000                    ;ERROR CORRECTION CODE INHIBIT (BIT #11)
2726            010000                  FMT22=  10000                   ;FORMAT BIT (BIT #12
2727
2728                                    ;TAPE CONTROL REGISTER (RHTC)
2729
2730            000001                  S1=     1                       ;SLAVE NUMBER
2731            000002                  S2=     2                       ;SLAVE NUMBER
2732            000004                  S4=     4                       ;SLAVE NUMBER
2733            000010                  EPAR=   10                      ;EVEN PARITY
2734            000020                  FMT1=   20                      ;FORMAT
2735            000040                  FMT2=   40                      ;FORMAT
2736            000100                  FMT4=   100                     ;FORMAT
2737            000200                  FMT8=   200                     ;FORMAT
2738            000400                  DEN1=   400                     ;DENSITY
2739            001000                  DEN2=   1000                    ;DENSITY
2740            002000                  DEN4=   2000                    ;DENSITY
2741            001400                  BPI8=   1400                    ;800 BPI
2742            000300                  NML=    300                     ;NORMAL - 2 FRAMES PER WORD
2743
2744
2745
2746
2747
2748                                    ;DRIVE TYPE REGISTER
2749
2750            002000                  SLVPR=  2000                    ;SLAVE PRESENT (BIT #10)
2751            010000                  CH7=    10000                   ;CHANNEL 7 (BIT #12)
2752
2753                                    ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
2754                                    ;EACH BIT IS CALLED BY BIT NUMBER.
2755
2756
2757
2758
2759                                    ;CURRENT CYLINDER ADDRESS (RHCC) (#13)
```

```
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;SERIAL NUMBER REGISTER (RHSN) (#14)
                                ;EACH IS CALLED BY BIT NUMBER



                                ;ERROR REGISTER #03 (RHER3) (#15)

000001                          PSU=    1               ;PACK SPEED UNSAFE (BIT #0)
000002                          VUF=    2               ;VELOCITY UNSAFE (BIT #1)
000010                          UWR=    10              ;ANY UNSAFE EXCEPT READ/WRITE (BIT #3)
000020                          PRE=    20              ;DISK PACK ROTATION ERROR (BIT #4)
000040                          ACL=    40              ;AC LOW (BIT #5)
000100                          DCL=    100             ;DC LOW (BIT #6)
040000                          SKI=    40000           ;SEEK INCOMPLETE (BIT #14)
100000                          OCYL=   100000          ;OFF CYLINDER (BIT #15)



                                ;ECC POSITION REGISTER (RHEC1) (#16)
                                ;EACH BIT IS CALLED BY BIT NUMBER



                                ;ECC PATTERN REGISTER (RHEC2) (#17)
                                ;EACH BIT IS CALLED BY BIT NUMBER



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

```
2798
2799
2800
2801            000000          CS1=    0               ;CONTROL STATUS 1
2802            000002          WC=     2               ;WORD COUNT
2803            000004          BA=     4               ;BUS ADDRESS
2804            000006          FC=     6               ;FRAME COUNT
2805            000006          DA=     6               ;DESIRED SECTOR/TRACK
2806            000010          CS2=    10              ;CONTROL STATUS 2
2807            000012          DS=     12              ;DRIVE STATUS
2808            000014          ER1=    14              ;ERROR 1
2809            000016          AS=     16              ;ATTENTION SUMMERY
2810            000020          LA=     20              ;LOOK AHEAD
2811            000022          DB=     22              ;DATA BUFFER
2812            000024          MR=     24              ;MAINTENANCE REGISTER
2813            000026          DT=     26              ;DRIVE TYPE
2814            000030          SN=     30              ;SERIAL NUMBER
2815            000032          OF=     32              ;OFFSET
2816            000032          TC=     32              ;TAPE CONTROL
2817            000034          DC=     34              ;DESIRED CYLINDER
2818            000036          CC=     36              ;CURRENT CYLINDER
2819            000040          ER2=    40              ;ERROR 2
2820            000042          ER3=    42              ;ERROR 3
2821            000044          EC1=    44              ;ECC 1
2822            000046          EC2=    46              ;ECC 2
2823
2824
2825                            ;BUFFERS
2826            003000          .=3000
2827    003000                  BUFEW:
2828    003000  000000          BFEVEN: 0
2829    003002                  BUFOW:
2830    003002  000422          BFODD:  .BLKW   274.            ;BUFFER
2831                            ;STARTING ADDRESS OF REGISTERS
2832
2833    004046  172040          RSCS1:  172040                  ;RS ALONG
2834    004050  176700          RPCS1:  176700                  ;RP ALONE
2835    004052  172440          TMCS1:  172440                  ;TM ALONE
2836    004054  176300          MIXCS1: 176300                  ;MIXED SYSTEM
2837    004056  000000          PRESENT:        0
2838
2839
2840
2841                            .SBTTL  REGISTER ADDRESSES
```

```
                                        ;THE FOLLOWING ARE THE I/O REGISTERS FOR THE DEVICE UNDER TEST
                                        ;THEY WILL BE FILLED WITH ADDRESSES DEPENDING ON WHAT DEVICE IS ON
                                        ;THE SYSTEM.
                                        ;IN THE CASE OF A MIXED SYSTEM THAT IS WITH MORE THAN ONE RH DEVICE
                                        ;THE FIRST ONE THAT EXISTS IN THE FOLLOWING LIST WILL BE PICKED
                                        ;AND THE OTHERS WILL NOT BE USED.
                                        ;LIST:-
                                        ;1)     RS03
                                        ;2)     RS04
                                        ;3)     RP04,5,6
                                        ;4)     TM02
                                        ;THE CONTENTS OF RHCS1 WHICH IS THE BASE ADDRESS IS
                                        ;172040 FOR RS03/04
                                        ;176700 FOR RP04,5,6
                                        ;172440 FOR TM02
                                        ;THE REST OF THE LOCATIONS WILL BE FILLED BY THE PROGRAM
004060  000000                  RHCS1:  0                       ;CONTROL AND STATUS 1
004062  000000                  RHWC:   0                               ;WORD COUNT
004064  000000                  RHBA:   0                               ;BUS ADDRESS
004066                          RHFC:                           ;FRAME COUNT
004066                          RHDA:                           ;DISK ADDRESS
004066  000000                  RHDST:  0                       ;DESIRED SECTOR/TRACK ADDRESS
004070  000000                  RHCS2:  0                               ;CONTROL AND STATUS 2
004072  000000                  RHDS1:  0                       ;DRIVE STATUS
004074  000000                  RHER1:  0                       ;ERROR #1
004076  000000                  RHAS:   0                       ;ATTENTION SUMMARY
004100                          RHCC:                           ;CHECK CHARACTER
004100  000000                  RHLA:   0                       ;LOOK-AHEAD
004102  000000                  RHDB:   0                               ;DATA BUFFER
004104  000000                  RHMR:   0                       ;MAINTAINABILITY
004106  000000                  RHDT:   0                       ;DRIVE TYPE
004110  000000                  RHSN:   0                       ;SERIAL NUMBER
004112                          RHTC:                           ;TAPE CONTROL
004112  000000                  RHOF:   0                       ;OFFSET
004114  000000                  RHCA:   0                       ;DESIRED CYLINDER ADDRESS
004116  000000                  RHCCA:  0                       ;CURRENT CYLINDER ADDRESS
004120  000000                  RHER2:  0                       ;ERROR #2
004122  000000                  RHER3:  0                       ;ERROR #3
004124  000000                  RHEC1:  0                       ;ECC POSITION
004126  000000                  RHEC2:  0                       ;ECC PATTERN
004130  000000                  RHBAE:  0                               ;BUS ADDRESS EXTENSION
004132  000000                  RHCS3:  0                       ;CONTROL AND STATUS 3


                                ;FUNCTION EQUATES

                                        ;TABLE OF FUNCTIONS FOR RHCS1 THEN "GO" BIT HAS TO BE SET
004134                          FUTABL:
004134  000000                  NOPERA: 0                       ;NO OPERATION
004136  000002                  UNLOAD: 2                       ;UNLOAD (STAND BY)
004140  000006                  RECALI: 6                       ;RECALIBRATE
004142  000010                  DCLEAR: 10                      ;DRIVE CLEAR
004144  000012                  RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
```

```
                                SERCH:  30                      ;SEARCH COMMAND
                                WRCHEK: 50                      ;WRITE CHECK DATA
                                WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
                                WRIDAT: 60                      ;WRITE DATA
                                WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
                                READAT: 70                      ;READ DATA
                                REFOR:  72                      ;READ HEADER AND DATA
                                SEECOM: 4                       ;SEEK COMMAND
                                OFSETC: 14                      ;OFFSET COMMAND
                                RETCL:  16                      ;RETURNTO CENTERLINE
                                PKACK:  22                      ;PACK ACKNOWLEDGE
                                READIN: 20                      ;READ IN
                                REWIND: 6                       ;REWIND
                                REVRED: 76                      ;REVERSE READ
                                SPACFD: 30                      ;SPACE FORWARD
                                REVWRT: 66              ;REVERSE WRITE
                                ILLEGL: .WORD   0               ;COMPUTED ILLEGAL FUNCTION

                                ;DATA BUFFER FOR READ WRITE
                                WRFROM: .BLKW   70.             ;WRITE FROM THIS BUFFER
                                REINTO: .BLKW   70.             ;READ INTO THIS BUFFER
                                COMND:  0               ;STORE COMMAND FOR COMMAND ROUTINE
                                COMACD: 0               ;STORE COMMACD
                                NOGO:   0                       ;IF ZERO GO IS TO BE GIVEN IN COMMAND ROUTINE
                                                                ;IF ONES GO IS NOT TO BE GIVEN IN COMMAND ROUTINE
                                                                ;USED IN PGE TEST
                                WATO:   0                       ;IF ZERO WAIT FOR BIT TO SET
                                                                ;IF ONES WAIT FOR BIT TO RESET
                                                                ;USED IN WAIT TRAP
                                WRTBIT: 0                       ;BITS NOT WRITTEN INTO
                                SETBIT: 0                       ;BITS ALWAYS SET
                                CLRBIT: 0                       ;BITS ALWAYS CLEARED


                                ;RESERVED LOCATIONS
                                TSTNM:  0                       ;TEST NUMBER
                                SLAVE:  0                       ;KEEP SLAVE NO.
                                IP:     0               ;IPCK NUMBER FOR ERROR PRINTOUT
                                SILONM: 0               ;SILO WORD NUMBER FOR ERROR PRINT OUT
                                WAITPC: 0               ;WAIT TRAP PC
                                WAITRE: 0                       ;WAITING FOR REGISTER ADDRESS IN WAIT TRAP
                                WAITBT: 0                       ;WAITING FOR BIT IN WAIT TRAP

                                ;TABLE FOR ATTENTION BITS
                                ;ATTENTION TABLE
004674  001  002  004  ATABLE: .BYTE   1,2,4,10,20,40,100,200
004677  010  020  040
004702  100  200


                                ;RESERVED LOCATIONS FOR UNIT SELECT
004704  172040          BASEAD: 172040          ;RS BASE ADDRESS
004706  176700          BASPAD: 176700          ;RP BASE ADDRESS
004710  172440          BASTAD: 172440          ;TU BASE ADDRESS
```

```
2954  004712  176300          BASEMAD:        176300          ;MIXED SYSTEM BASE ADDRESS
2955
2956
2957  004714  000000          BASEVC: 0               ;RS VECTOR
2958  004716  000254          BASP:   254                     ;RP VECTOR
2959  004720  000001          BAST:   1               ;TU VECTOR
2960  004722  000002          BASM:   2               ;MIXED VECTOR
2961
2962                          ;TABLE FOR GIVEN BASE ADDRESSES
2963  004724  000010          BSGIVA: .BLKW   8.
2964
2965
2966                          ;TABLE FOR GIVEN VECTOR
2967  004744  000010          BSGIVV: .BLKW   8.
2968
2969
2970  004764  000004          NMRHS:  4               ;NUMBER OF RH
2971  004766  000000          BASINX: 0               ;INDEX FOR BASE
2972  004770  000000          WORKBS: 0               ;WORKING BASE
2973  004772  000000          WORKNM: 0               ;WORKING NUMBER FOR RH
2974  004774  000000          WORKVC: 0               ;WORKING VECTOR FOR RH
2975  004776  000000          FORBAE: 0               ;TOTAL NO. OF REG. FOR BAE CALCULATION
2976  005000  000000          LOPCT:  0               ;LOOP COUNT FOR 200 START
2977  005002  000000          LOPCT1: 0               ;LOOP COUNT FOR 210 START
2978  005004  000000          USEVEC: 0               ;USE VECTOR
2979
2980
2981
2982  005006  000000          GIVE:   0               ;IF ONES OPERATOR WILL GIVE ADDRESSS
2983  005010  000000          UNIT:   0                       ;UNIT UNDER TEST
2984  005012  000000          ST200:  0               ;ALL ONES INDICATE STARTING FROM 200
2985                          ;TESTING TABLE
2986  005014  000010          TSRHNO: .BLKW   8.      ;RH NO TO BE TESTED IN ORDER OF TEST
2987
2988  005034  000010          TSDEVICE:       .BLKW   8.      ;DEVICE TO BE TESTED IN ORDER OF TEST
2989                                                          ;1=RS03,3/L,3/LA  2=RS04,4/L  4=RP04,5,6 SINGLE PORT
2990                                                          ;10=RP04,5,6 DUAL PORT  20=TM02
2991
2992
2993  005054  000010          TSUNIT: .BLKW   8.              ;UNIT NO. TO BE TESTED IN ORDER OF TEST
2994
2995
2996  005074  000010          TSSLAV: .BLKW   8.              ;SLAVE NO TO BE TESTED IN ORDER OF TEST
2997
2998
2999  005114  000000          TSTOTL: 0                       ;TOTAL NO. OF UNITS TO BE TESTED
3000  005116  000000          TSINDX: 0                       ;INDEX TO ONE UNDER TEST
3001
3002
3003  005120  000010          TSVEC:  .BLKW   8.              ;VECTOR ADDRESS IN ORDER OF TEST
3004
3005
3006  005140  000010          TSBAE:  .BLKW   8.              ;RHBAE ADDRESS IN ORDER OF TEST
3007
3008
3009  005160  000010          TSCS3:  .BLKW   8.              ;RHCS3 ADDRESS IN ORDER OF TEST
```

```
3010
3011
3012  005200  000010                TSCOMD: .BLKW   8.              ;COMMAND ADDRESS IN ORDER OF TEST
3013
3014
3015  005220  000010                TSBASE: .BLKW   8.              ;BASE ADDRESS IN ORDER OF TEST
3016
3017
3018
3019
3020
3021
3022  005240  000000                ERFLG$: 0                       ;ERROR FLAG
3023  005242  000000                FIRST:  0                       ;IF ZERO WILL TYPE HEADER
3024                                                                ;IF ONES WILL NOT TYPE HEADER
3025
3026
3027
3028  005244  000000                ATTENT: 0               ;ATTENTION BIT FOR PRESENT UNIT
3029  005246  000000                TOTALAT:        0                       ;TATAL ATTENTION BITS
3030
3031  005250  000000                TMP0:   .WORD   0               ;TEMP STORAGE
3032  005252  000000                TMP1:   .WORD   0
3033  005254  000000                TMP4:   .WORD   0               ;TEMP STORAGE
3034                                ;PERMANENT TABLE
3035  005256  172040                PERRS:  172040                  ;RS BASE ADDRESS
3036  005260  176700                PERRP:  176700                  ;RP BASE ADDRESS
3037  005262  172440                PERTU:  172440                  ;TU BASE ADDRESS
3038  005264  176300                PERMX:  176300                  ;MIXED BASE ADDRESS
3039
3040  005266  000204                PERRSV: 204                     ;RS VECTOR ADDRESS
3041  005270  000254                PERRPV: 254                     ;RP VECTOR ADDRESS
3042  005272  000224                PERTUV: 224                     ;TU VECTOR ADDRESS
3043  005274  000000                PERMXV: 0                       ;MIXED VECTOR ADDRESS
3044
3045  005276  000004                PERNUM: 4                       ;NUMBER OF RH
3046
3047                                ;WORKING TABLE
3048
3049  005300  000004                DEVPNT: .BLKW   4               ;BASE ADDRESS TO BE TESTED
3050  005310  000004                VECPNT: .BLKW   4               ;VECTOR ADDRESS TO BE TESTED
3051  005320  000000                LTRY:   0                       ;NO. OF UNITS TO BE TESTED
3052  005322  000000                TFOUND: 0                       ;TWICE NUMBER OF RH FOUND
3053  005324  000000                TESTDV: 0                       ;STORES BASE ADDRESS OF TEST DEVICE
3054  005326  000000                TESTVC: 0                       ;STORES VECTOR OF TEST DEVICE
3055                                                                ;THE ABOVE TWO IS BEFORE ANY
3056                                                                ;DEVICE IS FOUND.
3057                                ;TABLE FOR DRIVE TYPES
3058                                ;0=RS03, 1=RS03/L, 2=RS04, 3=RS04/L, 4=RS03/LA
3059                                ;20020=SINGLE PORT RP04, 24020=DUAL PORT RP04
3060                                ;20021=SINGLE PORT RP05, 24021=DUAL PORT RP05
3061                                ;20022=SINGLE PORT RP06, 24022=DUAL PORT RP06
3062                                ;142010=TU16
3063
3064  005330  000000  000001  000002  TYPNT:   .WORD   0,1,2,3,4,20020,24020,20021,24021,20022,24022,142010
3065  005336  000003  000004  020020
```

```
3066  005344  024020  020021  024021
3067  005352  020022  024022  142010
3068  005360  000000          POINTT: 0                       ;POITER TO ABOVE TABLE
3069  005362  000014          NTYPNT: 14                      ;NUMBER OF ABOVE TYPES
3070
3071  005364  000000          TMPSLV: 0                       ;TEMPORARY SLAVE COUNTER
3072
3073                                  ;TABLE OF DATA FOR DEVICES FOUND
3074  005366  000004          FDEVIC: .BLKW   4               ;DEVICE FOUND, 1=TU, 2=RP DUAL
3075                                                          ;3=RP SINGLE, 4=RS04, 5=RS03
3076  005376  000004          FUNITN: .BLKW   4               ;UNIT NUMBER FOUND
3077  005406  000004          FTYPE:  .BLKW   4               ;DDRIVE TYPE FOUND
3078  005416  000004          FVECTR: .BLKW   4               ;VECTOR FOUND
3079  005426  000004          FBASEA: .BLKW   4               ;BASE ADDRESS FOUND
3080  005436  000004          FRHNM:  .BLKW   4               ;RH NUMBER FOUND
3081  005446  000004          FSLAVE: .BLKW   4               ;TU16 SLAVE NUMBER FOUND
3082  005456  000004          FDRIVR: .BLKW   4               ;DRIVER ADDRESS FOUND
3083  005466  000004          FBAE:   .BLKW   4               ;BAE ADDRESS FOUND
3084  005476  000004          FCS3:   .BLKW   4               ;CS3 ADDRESS FOUND
3085
3086  005506  000000          TSTUNT: 0                       ;TOTAL NUMBER OF RH FOUND AT
3087                                                          ;END OF SIZE
3088  005510  000000          WORUNT: 0                       ;SAME AS ABOVE BUT TO BE
3089                                                          ;DECREASED AT END PROGRAM
3090
3091  005512  000000          VECTOR: 0                       ;VECTOR UNDER TEST
3092  005514  177740          LERADD: 177740                  ;LOW ERROR ADDRESS REG.
3093  005516  177742          HERADD: 177742                          ;HIGH ERROR ADDRESS REG
3094  005520  177744          MEMERR: 177744                          ;MEMORY SYSTEM ADDRESS REG
3095          000114          PARE70=114                      ;PARITY ERROR VECTOR FOR 70
```

```
                                                        .SBTTL  REGISTER TEST
005522  012737  177777  005012  BEGIN:  MOV     #-1,ST200
005530  000402                  BR      START
005532  005037  005012          BEGIN2: CLR     ST200

005536                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
005536  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
005542  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
005544  022706  001140                  CMP     #SWR,R6 ;;DONE?
005550  001374                          BNE     .-6             ;;LOOP BACK IF NO
005552  012706  001000                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
005556  012737  043120  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005564  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005572  012737  044744  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005600  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005606  012737  046436  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005614  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005622  012737  046516  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
005630  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
005636  005037  001212                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
005642  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
005646  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
005654  012737  005654  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
005662  012737  005662  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
005670  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
005674  012737  005730  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
005702  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
005710  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
005716  022777  177777  173214          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
005724  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
005726  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
005730  012716  005736          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
005734  000002                          RTI
005736  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
005744  012737  000174  001142          MOV     #DISPREG,DISPLAY
005752  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR


005756  012737  000000  177776          MOV     #0,PS                   ;SET PROCESSOR STATUS TO 0
005764  012737  000200  000036          MOV     #200,@#TRAPVEC+2                ;TRAP PRIORITY = 4
005772  013700  002716                  MOV     @#RPVEC,R0              ;GET RP VECTOR ADDRESS
005776  012720  043056                  MOV     #RPVECT,(R0)+           ;THIS IS FOR UNTIMELY INTERRUPTS
006002  012710  000340                  MOV     #340,(R0)               ;RP04 INTERRUPT SERVICE ROUTINE
                                                                        ;PRIORITY = 7
006006  004737  044064                  JSR     PC,@#$TKINT     ;INITILIZE THE TK
006012  005037  045100                  CLR     PRITEM                  ;CLEAR FOR ERROR MESSAGE PRINTOUT
006016  005737  005242                  TST     @#FIRST                 ;IS THIS FIRST TIME ROUND
006022  001001                          BNE     1$                      ;BRANCH IF NOT
006024  000402                          BR      2$
```

```
006026  000137  006114          1$:     JMP     @#SND1
006032                          2$:
006032  104401  006040                  TYPE    ,69$            ;;TYPE ASCIZ STRING
006036  000426                          BR      67$             ;;GET OVER THE ASCIZ
                                ;;68$:  .ASCIZ  <15><12>/RH70 FUNCTIONAL CONTROLLER TEST  DERHA-B/
006114                          67$:
006114  012737  177777  005242  SND1:   MOV     #-1,@#FIRST             ;NEXT TIME DO NOT GIVE HEADER
006122  012737  037736  000114          MOV     #PARITY,@#PARE70        ;PARITY VECTOR
006130  012737  000340  000116          MOV     #340,@#PARE70+2         ;PRIORITY 7
006136  012737  037600  000004          MOV     #TIEOUT,@#ERRVEC        ;TIMEOUT VECTOR
006144  012737  000340  000006          MOV     #340,@#ERRVEC+2         ;PRIORITY 7
```

```
;;*****************************************************************
;*TEST 1         SIZE FOR RH DEVICES
;*       THIS TEST DETERMINS WHICH RH DEVICE IS ON THE SYSTEM.
;*       IF THE SYSTEM HAS MORE THAN ONE RH DEVICE THEN ONLY ONE OF
;*       THE DEVICES WILL BE USED IN THE FOLLOWING TESTS.
;*       THE ONE THAT WILL BE USED IS THE FIRST ONE THAT IS PRESENT
;*       IN THE FOLLOWING LIST
;*       RS
;*       RP
;*       TM
;*
;*       THE WAY THE TEST WORKS IS AS FOLLOWS:-
;*       A REFRENCE IS MADE TO THE CONTROL AND STATUS REGISTER 1
;*       FOR THE RS BY A TST INSTRUCTION. IF IT RESPONDS THEN IT IS
;*       ASSUMED PRESENT AND THE LOCATIONS FOR THE I/O REGISTERS ARE
;*       FILLED WITH THE APPROPIATE ADDRESSES.
;*       THEN THE DRIVE TYPE REGISTER IS CHECKED TO HAVE GOOD
;*       VALUES.
;*       IF THE TST INSTRUCTION TRAPS TO A NON EXISTANT VECTOR
;*       THEN A SIMILAR ATTEMPT IS MADE TO A RP CONTROL AND STATUS
;*       REGISTER 1.
;*       THEN A TM IS TRYED.
;*       THEN A MIXED SYSTEM WITH MORE THAN ONE RH DEVICS IS TRYED.

;;*****************************************************************
006152  000004  TST1:   SCOPE
```

```
3208  006154  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
3209  006162  005737  005012                  TST     ST200           ;IS IT A 200 START
3210  006166  001413                          BEQ     2$              ;BRANCH IF NOT 200 START
3211  006170  012700  005256                  MOV     #ERRS,R0        ;POINTER TO MOVE 9 WMRDS FROM
3212  006174  012701  000011                  MOV     #9.,R1          ;NUMBER TO MOVE
3213  006200  012702  005300                  MOV     #DEVPNT,R2      ;POINTER TO MOVE 9 TORDS TO
3214  006204  012022                  1$:     MOV     (R0)+,(R2)+     ;FILL WORK TABLE
3215  006206  005301                          DEC     R1
3216  006210  001375                          BNE     1$              ;BRANCH IF 9 NOT DONE
3217  006212  000137  006512                  JMP     SETSIZ
3218  006216                          2$:
3219  006216  104401  006224                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3220  006222  000417                          BR      64$             ;;GET OVER THE ASCIZ
3221                                  ;;65$:  .ASCIZ  <15><12>/HOW MANY RH TO BE TESTED   /
3222  006262                          64$:
3223  006262  104401  006270                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3224  006266  000402                          BR      66$             ;;GET OVER THE ASCIZ
3225                                  ;;67$:  .ASCIZ  <15><12>/ /
3226  006274                          66$:
3227  006274  104410                          RDOCT
3228  006276  011637  005320                  MOV     (SP),LTRY       ;GET NUMBER OF RH TO BE TESTED
3229  006302  012637  005250                  MOV     (SP)+,TMP0      ;WORKING NUMBER OF RH
3230  006306  012737  000001  005252          MOV     #1,TMP1         ;RH NUMBER TO BE TYPED
3231  006314  005000                          CLR     R0              ;INDEX TO WORKING TABLE
3232
3233  006316                          3$:
3234  006316  104401  006324                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3235  006322  000417                          BR      68$             ;;GET OVER THE ASCIZ
3236                                  ;;69$:  .ASCIZ  <15><12>/TYPE BASE ADDRESS OF RH NO/
3237  006362                          68$:
3238  006362  013746  005252                  MOV     TMP1,-(SP)
3239  006366  104405                          TYPDS
3240  006370  104401  006376                  TYPE    ,71$            ;;TYPE ASCIZ STRING
3241  006374  000402                          BR      70$             ;;GET OVER THE ASCIZ
3242                                  ;;71$:  .ASCIZ  <15><12>/ /
3243  006402                          70$:
3244  006402  104410                          RDOCT
3245  006404  012660  005300                  MOV     (SP)+,DEVPNT(R0)  ;FILL WORKING TABLE WITH BASE ADDRESS
3246  006410  104401  006416                  TYPE    ,73$            ;;TYPE ASCIZ STRING
3247  006414  000420                          BR      72$             ;;GET OVER THE ASCIZ
3248                                  ;;73$:  .ASCIZ  <15><12>/TYPE VECTOR ADDRESS OF RH NO/
3249  006456                          72$:
3250  006456  013746  005252                  MOV     TMP1,-(SP)
3251  006462  104405                          TYPDS
3252  006464  104401  000200                  TYPE    ,CRLF
3253  006470  104410                          RDOCT
3254  006472  012660  005310                  MOV     (SP)+,VECPNT(R0)  ;FILL WORKING TABLE WITH VECTOR ADDRESS
3255  006476  005720                          TST     (R0)+           ;ADD 2 TO R0
3256  006500  005237  005252                  INC     TMP1            ;GET NEXT RH NUMBER
3257  006504  005337  005250                  DEC     TMP0            ;COUNT DOWN RH TO BE TESTED
3258  006510  001302                          BNE     3$              ;BRANCH IF ALL NOT DONE
3259
3260                                  ;R0 WILL HAVE DEVICE POINTER, R1 WILL HAVE VECTOR POINTER
3261  006512  012700  005300          SETSIZ: MOV     #DEVPNT,R0      ;DEVICE POINTER
3262  006516  012701  005310                  MOV     #VECPNT,R1      ;VECTOR PMINTER
3263  006522  005037  005322                  CLR     TFOUND          ;WILL BE USED AS POINTER
```

```
3264  006526                                SS2:                    ;TO FOUND TABLE.
3265  006526  012737  010520  000004  1$:   MOV   #SS4,ERRVEC       ;TIME OUT VECTOR TO REDUCE LTRY
3266  006534  012037  005324              MOV   (R0)+,TESTDV        ;ADDRESS OF DEVICE TO BE TESTED
3267  006540  012137  005326              MOV   (R1)+,TESTVC        ;VECTOR TO BE TESTED
3268  006544  005777  176554              TST   @TESTDV             ;TRY TIME OUT ON DEVICE BASE
3269                                      ;IF NO TIME OUT OCCURS THEN DEVICE IN TESTDV IS PRESENT
3270                                      ;NOW FILL ALL REGISTER ADDRESS TABLE
3271  006550  012702  000026              MOV   #22.,R2             ;COUNT 22
3272  006554  012703  004060              MOV   #RHCS1,R3           ;GET BASE LOCATION
3273  006560  01370   005324              MOV   TESTDV,R4           ;GET BASE ADDRESS
3274  006564  01042                 2$:    MOV   R4,(R3)+            ;FILL PROPER ADDRESS
3275  006566  062704  000002              ADD   #2,R4               ;INCREMENT ADDRESS BY 2
3276  006572  005302                      DEC   R2                  ;COUNT DOWN
3277  006574  001373                      BNE   2$                  ;BRANCH IF 22 NOT DONE.
3278  006576  005077  175266              CLR   @RHCS2              ;SET UNIT NUMBER
3279  006602                         SS6:
3280  006602  012737  005330  005360  3$:   MOV   #TYPNT,POINTT     ;POINTER TO DRIVE TYPE
3281  006610  012737  000014  005362        MOV   #14,NTYPNT          ;NUMBER OF DRIVE TYPES
3282
3283  006616                         SS7:
3284  006616  022737  000001  005362  4$:   CMP   #1,NTYPNT          ;IS IT FOR TU
3285  006624  001424                      BEQ   5$                  ;IF FOR TU BRANCH
3286  006626  017737  175254  001126        MOV   @RHDT,$BDDAT        ;READ DRIVE TYPE
3287  006634  012702  000014              MOV   #14,R2  ;GET NUMBER FOR DT NOT DONE
3288  006640  163702  005362              SUB   NTYPNT,R2                 ;GET DT TO DO.
3289  006644  006302                      ASL   R2                  ;MULTIPLY R2 BY 2
3290  006646  026237  005330  001126        CMP   TYPNT(R2),$BDDAT  ;IS DRIVE TYPE FOUND
3291  006654  001402                      BEQ   16$                 ;DRIVE TYPE FOUND SO BRANCH
3292  006656  000137  010544              JMP   SS8
3293  006662  032777  010000  175200  16$:  BIT   #NED,@RHCS2        ;IS IT NON EXISTANT DRIVE
3294  006670  001437                      BEQ   8$                  ;NED NOT SET, DRIVE TYPE FOUND SO BRANCH
3295                                      ;A DEVICE HAS NOT BEEN FOUND
3296  006672  000137  010536              JMP   SS5                 ;A DEVICE HAS NOT BEEN FOUND SO BRANCH
3297
3298                                      ;TRYING THE TU16 DRIVE TYPE
3299  006676  005037  005364        5$:   CLR   TMPSLV              ;SLAVE 0
3300  006702  013777  005364  175202  6$:   MOV   TMPSLV,@RHTC       ;MOVE SLAVE NUMBER IN TAPE CONTROL
3301  006710  017737  175172  001126        MOV   @RHDT,$BDDAT        ;READ DRIVE TYPE
3302  006716  032737  002000  001126        BIT   #SLVPR,$BDDAT       ;IS SLAVE PRESENT
3303  006724  001014                      BNE   7$
3304  006726  022737  000007  005364        CMP   #7,TMPSLV           ;IS 7 DONE
3305  006734  001002                      BNE   17$                 ;BRANCH IF 7 DONE AND NONE FOUND
3306  006736  000137  010544              JMP   SS8
3307  006742  005237  005364        17$:  INC   TMPSLV              ;GET NEXT SLAVE
3308  006746  012777  040000  175104        MOV   #TRE,@RHCS1         ;CLEAR CONTROLLER ERRORS
3309  006754  000752                      BR    6$                  ;TRY NEXT SLAVE
3310                                      ;A DEVICE HAS BEEN FOUND SAVE SLAVE
3311  006756  013702  005322        7$:   MOV   TFOUND,R2           ;GET NO OF RH FOUND X2 FOR INDEX
3312  006762  013762  005364  005446        MOV   TMPSLV,FSLAVE(R2)
3313                                      ;A DEVICE HAS BEEN FOUND TEST VECTOR
3314  006770  012737  007004  000004  8$:   MOV   #9$,ERRVEC         ;TIME OUT VECTOR TO REPORT ERROR
3315  006776  005777  176324              TST   @TESTVC             ;TRY TIME OUT ON VECTOR
3316  007002  000403                      BR    10$                 ;  VECTOR EXISTS SO CONTINUE
3317  007004  104146                 9$:   ERROR 146                 ;A DEVICE BASE ADDRESS DID NOT
3318                                                                  ;TIME OUT BUT ITS CORRESPONDING
3319                                                                  ;VECTOR ADDRESS TIMED OUT
```

```
                                                        ;HIT CONTINUE TO REPEAT THIS
                                                        ;TEST
007006  000000                          HALT
007010  000767                          BR      8$              ;REPEAT THIS TEST
                                        ;A DEVICE HAS BEEN FOUND AND VECTOR RESPONDS SO STORE RESULTS
007012  013702  005322          10$:    MOV     TFOUND,R2       ;GET NO OF RH FOUND X2 FOR INDEX
007016  013762  005362  005366          MOV     NTYPNT,FDEVIC(R2)  ;DEVICE 1=TU; 2,3,4,5,6,7=RP;
                                                                ;10,11,12,13,14=RS
007024  017762  175040  005376          MOV     @RHCS2,FUNITN(R2)  ;GET RHCS2
007032  042762  177770  005376          BIC     #177770,FUNITN(R2)  ;KEEP UNIT NUMBER
007040  017762  175042  005406          MOV     @RHDT,FTYPE(R2) ;DRIVE BYTE
007046  013762  005326  005416          MOV     TESTVC,FVECTR(R2)  ;VECTOR
007054  013762  005324  005426          MOV     TESTDV,FBASEA(R2)  ;BASE ADDRESS
007062  013762  005322  005436          MOV     TFOUND,FRHNM(R2)  ;RH NUMBER X2 MINUS ONE
007070  006262  005436                  ASR     FRHNM(R2)       ;RH NUMBER MINUS ONE
007074  005262  005436                  INC     FRHNM(R2)       ;RH NUMBER
007100  022762  000001  005366          CMP     #1,FDEVIC(R2)   ;IS IT TU
007106  001016                          BNE     11$             ;BRANCH IF NOT TU
007110  012762  041710  005456          MOV     #CMNDTM,FDRIVR(R2)  ;DRIVER ADDRESS FOR TU
007116  013746  005324                  MOV     TESTDV,-(SP)    ;BASE ADDRESS
007122  062716  000034                  ADD     #<2*14.>,(SP)   ;15TH ADDRESS IS BAE FOR TU
007126  011662  005466                  MOV     (SP),FBAE(R2)   ;SAVE BAE FOR TU
007132  062716  000002                  ADD     #2,(SP)         ;16TH ADDRESS IS CS3 FOR TU
007136  012662  005476                  MOV     (SP)+,FCS3(R2)  ;SAVE CS3 FOR TU
007142  000511                          BR      SS1
007144  022762  000002  005366  11$:    CMP     #2,FDEVIC(R2)   ;IS IT RP DUAL PORT
007152  001424                          BEQ     12$             ;BRANCH IF RP FOUND
007154  022762  000003  005366          CMP     #3,FDEVIC(R2)   ;IS IT RP
007162  001420                          BEQ     12$             ;BRANCH IF RP FOUND
007164  022762  000004  005366          CMP     #4,FDEVIC(R2)   ;IS IT RP DUAL PORT
007172  001414                          BEQ     12$             ;BRANCH IF RP FOUND
007174  022762  000005  005366          CMP     #5,FDEVIC(R2)   ;IS IT RP
007202  001410                          BEQ     12$             ;BRANCH IF RP FOUND
007204  022762  000006  005366          CMP     #6,FDEVIC(R2)   ;IS IT RP DUAL PORT
007212  001404                          BEQ     12$             ;BRANCH IF RP FOUND
007214  022762  000007  005366          CMP     #7,FDEVIC(R2)   ;IS IT RP
007222  001016                          BNE     13$             ;BRANCH IF NO RP
007224  012762  041572  005456  12$:    MOV     #CMNDRP,FDRIVR(R2)  ;DRIVER ADDRESS FOR RP
007232  013746  005324                  MOV     TESTDV,-(SP)    ;BASE ADDRESS
007236  062716  000050                  ADD     #<2*20.>,(SP)   ;21ST ADDRESS IS BAE FOR RP
007242  011662  005466                  MOV     (SP),FBAE(R2)   ;SAVE BAE FOR RP
007246  062716  000002                  ADD     #2,(SP)         ;22ND ADDRESS IS CS3 FOR RP
007252  012662  005476                  MOV     (SP)+,FCS3(R2)  ;SAVE CS3 FOR RP
007256  000443                          BR      SS1
007260  022762  000010  005366  13$:    CMP     #10,FDEVIC(R2)  ;IS IT RS03/LA
007266  001420                          BEQ     14$             ;BRANCH IF RS03/LA
007270  022762  000011  005366          CMP     #11,FDEVIC(R2)  ;IS IT RS04/L
007276  001414                          BEQ     14$             ;BRANCH IF RS04/L
007300  022762  000012  005366          CMP     #12,FDEVIC(R2)  ;IS IT RS04
007306  001410                          BEQ     14$             ;BRANCH IF RS04
007310  022762  000013  005366          CMP     #13,FDEVIC(R2)  ;IS IT RS03/L
007316  001404                          BEQ     14$             ;BRANCH IF RS03/L
007320  022762  000014  005366          CMP     #14,FDEVIC(R2)  ;IS IT RS03
007326  001016                          BNE     15$             ;BRANCH IF NOT RS03
```

```
                                                                                
 3376  007330  012762  041502  005456  14$:    MOV     #CMNDRS,FDRIVR(R2)  ;DRIVER ADDRESS FOR RS
 3377  007336  013746  005324                  MOV     TESTDV,-(SP)        ;BASE ADDRESS
 3378  007342  062716  000030                  ADD     #(2*12.),(SP)       ;13TH ADDRESS IS BAE FOR RS
 3379  007346  011662  005466                  MOV     (SP),FBAE(R2)       ;SAVE BAE FOR RS
 3380  007352  062716  000002                  ADD     #2,(SP)             ;14TH ADDRESS IS CS3 FOR RS
 3381  007356  012662  005476                  MOV     (SP)+,FCS3(R2)      ;SAVE CS3 FOR RS
 3382  007362  000401                          BR      SS1                 ;CONTINUE
 3383  007364  000000              15$:        HALT                        ;PROGRAM ERROR
 3384
 3385                                          ;NOW TYPE RESULT
 3386
 3387  007366                      SS1:
 3388  007366  104401  007374                  TYPE    ,65$                ;;TYPE ASCIZ STRING
 3389  007372  000406                          BR      64$                 ;;GET OVER THE ASCIZ
 3390                              ;;65$:      .ASCIZ  <15><12>/ON RH NO/
 3391  007410                      64$:
 3392  007410  016246  005436                  MOV     FRHNM(R2),-(SP) ;TYPE RH NUMBER
 3393  007414  104405                          TYPDS
 3394  007416  104401  007424                  TYPE    ,67$                ;;TYPE ASCIZ STRING
 3395  007422  000411                          BR      66$                 ;;GET OVER THE ASCIZ
 3396                              ;;67$:      .ASCIZ  /          BASE /
 3397  007446                      66$:
 3398  007446  016246  005426                  MOV     FBASEA(R2),-(SP)  ;TYPE BASE ADDRESS
 3399  007452  104402                          TYPOC
 3400  007454  104401  007462                  TYPE    ,69$                ;;TYPE ASCIZ STRING
 3401  007460  000404                          BR      68$                 ;;GET OVER THE ASCIZ
 3402                              ;;69$:      .ASCIZ  / FOUND/
 3403  007472                      68$:
 3404  007472  022762  000001  005366          CMP     #1,FDEVIC(R2)       ;IS IT TU
 3405  007500  001020                          BNE     1$                  ;BRANCH IF NOT TU
 3406  007502  104401  007510                  TYPE    ,71$                ;;TYPE ASCIZ STRING
 3407  007506  000410                          BR      70$                 ;;GET OVER THE ASCIZ
 3408                              ;;71$:      .ASCIZ  / TU16 AT SLAVE/
 3409  007530                      70$:
 3410  007530  016246  005446                  MOV     FSLAVE(R2),-(SP)  ;TYPE SLAVE NUMBER
 3411  007534  104405                          TYPDS
 3412  007536  000137  010336                  JMP     6$
 3413                                                                  ;THE FOLLOWING TYPES OUT
 3414                                                                  ;THE DEVICE AND REDEFINES
 3415                                                                  ;FDEVIC: 1=TU, 2=RP DUAL PORT
 3416                                                                  ;3=RP SINGLE PORT
 3417                                                                  ;4=RS04/L,RS04
 3418                                                                  ;5=RS03/LA,RS03/L,RS03
 3419  007542  022762  000002  005366  1$:     CMP     #2,FDEVIC(R2)       ;IS IT RP06 DUAL PORT
 3420  007550  001016                          BNE     2$                  ;BRANCH IF NOT RP06 DUAL PORT
 3421  007552  104401  007560                  TYPE    ,73$                ;;TYPE ASCIZ STRING
 3422  007556  000411                          BR      72$                 ;;GET OVER THE ASCIZ
 3423                              ;;73$:      .ASCIZ  / RP06 DUAL PORT /
 3424  007602                      72$:
 3425  007602  000137  010336                  JMP     6$
 3426  007606  022762  000003  005366  2$:     CMP     #3,FDEVIC(R2)       ;IS IT RP06 SINGLE PORT
 3427  007614  001016                          BNE     3$                  ;BRANCH IF NOT RP06 SINGLE PORT
 3428  007616  104401  007624                  TYPE    ,75$                ;;TYPE ASCIZ STRING
 3429  007622  000411                          BR      74$                 ;;GET OVER THE ASCIZ
 3430                              ;;75$:      .ASCIZ  / RP06 SINGLE PORT/
 3431  007646                      74$:
```

```
3432  007646  000137  010336                      JMP     6$
3433  007652  022762  000004  005366  3$:         CMP     #4,FDEVIC(R2)   ;IS IT RP05 DUAL PORT
3434  007660  001020                              BNE     4$              ;BRANCH IF NOT RP05 DUAL PORT
3435  007662  012762  000002  005366              MOV     #2,FDEVIC(R2)
3436  007670  104401  007676                      TYPE    ,77$            ;;TYPE ASCIZ STRING
3437  007674  000410                              BR      76$             ;;GET OVER THE ASCIZ
3438                                        ;;77$:  .ASCIZ  / RP05 DUAL PORT/
3439  007716                                    76$:
3440  007716  000137  010336                      JMP     6$
3441  007722  022762  000005  005366  4$:         CMP     #5,FDEVIC(R2)   ;IS IT RP05 SINGLE PORT
3442  007730  001020                              BNE     8$              ;BRANCH IF NOT RP05 SINGLE PORT
3443  007732  012762  000003  005366              MOV     #3,FDEVIC(R2)
3444  007740  104401  007746                      TYPE    ,79$            ;;TYPE ASCIZ STRING
3445  007744  000411                              BR      78$             ;;GET OVER THE ASCIZ
3446                                        ;;79$:  .ASCIZ  / RP05 SINGLE PORT/
3447  007770                                    78$:
3448  007770  000562                              BR      6$
3449  007772  022762  000006  005366  8$:         CMP     #6,FDEVIC(R2)   ;IS IT RP04 DUAL PORT
3450  010000  001017                              BNE     9$              ;BRANCH IF NOT RP04 DUAL PORT
3451  010002  012762  000002  005366              MOV     #2,FDEVIC(R2)
3452  010010  104401  010016                      TYPE    ,81$            ;;TYPE ASCIZ STRING
3453  010014  000410                              BR      80$             ;;GET OVER THE ASCIZ
3454                                        ;;81$:  .ASCIZ  / RP04 DUAL PORT/
3455  010036                                    80$:
3456  010036  000537                              BR      6$
3457  010040  022762  000007  005366  9$:         CMP     #7,FDEVIC(R2)   ;IS IT RP04 SINGLE PORT
3458  010046  001020                              BNE     10$             ;BRANCH IF NOT RP04 SINGLE PORT
3459  010050  012762  000003  005366              MOV     #3,FDEVIC(R2)
3460  010056  104401  010064                      TYPE    ,83$            ;;TYPE ASCIZ STRING
3461  010062  000411                              BR      82$             ;;GET OVER THE ASCIZ
3462                                        ;;83$:  .ASCIZ  / RP04 SINGLE PORT/
3463  010106                                    82$:
3464  010106  000513                              BR      6$
3465  010110  022762  000010  005366  10$:        CMP     #10,FDEVIC(R2)  ;IS IT RS03-LA
3466  010116  001014                              BNE     11$             ;BRANCH IF NOT RS03-LA
3467  010120  012762  000005  005366              MOV     #5,FDEVIC(R2)
3468  010126  104401  010134                      TYPE    ,85$            ;;TYPE ASCIZ STRING
3469  010132  000405                              BR      84$             ;;GET OVER THE ASCIZ
3470                                        ;;85$:  .ASCIZ  / RS03-LA/
3471  010146                                    84$:
3472  010146  000473                              BR      6$
3473  010150  022762  000011  005366  11$:        CMP     #11,FDEVIC(R2)  ;IS IT RS04-L
3474  010156  001013                              BNE     12$             ;BRANCH IF NOT RS04-L
3475  010160  012762  000004  005366              MOV     #4,FDEVIC(R2)
3476  010166  104401  010174                      TYPE    ,87$            ;;TYPE ASCIZ STRING
3477  010172  000404                              BR      86$             ;;GET OVER THE ASCIZ
3478                                        ;;87$:  .ASCIZ  / RS04-L/
3479  010204                                    86$:
3480  010204  000454                              BR      6$
3481  010206  022762  000012  005366  12$:        CMP     #12,FDEVIC(R2)  ;IS IT RS04
3482  010214  001012                              BNE     13$             ;BRANCH IF NOT RS04
3483  010216  012762  000004  005366              MOV     #4,FDEVIC(R2)
3484  010224  104401  010232                      TYPE    ,89$            ;;TYPE ASCIZ STRING
3485  010230  000403                              BR      88$             ;;GET OVER THE ASCIZ
3486                                        ;;89$:  .ASCIZ  / RS04/
3487  010240                                    88$:
```

```
3488  010240  000436                          BR      6$
3489  010242  022762  000013  005366  13$:    CMP     #13,FDEVIC(R2)  ;IS IT RS03-L
3490  010250  001013                          BNE     14$             ;BRANCH IF NOT RS03-L
3491  010252  012762  000005  005366          MOV     #5,FDEVIC(R2)
3492  010260  104401  010266                  TYPE    ,91$            ;;TYPE ASCIZ STRING
3493  010264  000404                          BR      90$             ;;GET OVER THE ASCIZ
3494                                  ;;91$:  .ASCIZ  / RS03-L/
3495  010276                          90$:
3496  010276  000417                          BR      6$
3497  010300  022762  000014  005366  14$:    CMP     #14,FDEVIC(R2)  ;IS IT RS03
3498  010306  001012                          BNE     5$              ;BRANCH IF NOT RS03
3499  010310  012762  000005  005366          MOV     #5,FDEVIC(R2)
3500  010316  104401  010324                  TYPE    ,93$            ;;TYPE ASCIZ STRING
3501  010322  000403                          BR      92$             ;;GET OVER THE ASCIZ
3502                                  ;;93$:  .ASCIZ  / RS03/
3503  010332                          92$:
3504  010332  000401                          BR      6$
3505  010334  000000                  5$:     HALT                    ;PROGRAM ERROR
3506  010336                          6$:
3507  010336  104401  010344                  TYPE    ,95$            ;;TYPE ASCIZ STRING
3508  010342  000411                          BR      94$             ;;GET OVER THE ASCIZ
3509                                  ;;95$:  .ASCIZ  <15><12>/AT UNIT NUMBER/
3510  010366                          94$:
3511  010366  016246  005376                  MOV     FUNITN(R2),-(SP)  ;TYPE UNIT NUMBER
3512  010372  104405                          TYPDS
3513  010374  104401  010402                  TYPE    ,97$            ;;TYPE ASCIZ STRING
3514  010400  000411                          BR      96$             ;;GET OVER THE ASCIZ
3515                                  ;;97$:  .ASCIZ  /          VECTOR /
3516  010424                          96$:
3517  010424  016246  005416                  MOV     FVECTR(R2),-(SP)  ;TYPE VECTOR
3518  010430  104402                          TYPOC
3519  010432  104401  010440                  TYPE    ,99$            ;;TYPE ASCIZ STRING
3520  010436  000402                          BR      98$             ;;GET OVER THE ASCIZ
3521                                  ;;99$:  .ASCIZ  <15><12>/ /
3522  010444                          98$:
3523  010444  104401  010452                  TYPE    ,101$           ;;TYPE ASCIZ STRING
3524  010450  000402                          BR      100$            ;;GET OVER THE ASCIZ
3525                                  ;;101$: .ASCIZ  <15><12>/ /
3526  010456                          100$:
3527
3528                                          ;UPDATE TFOUND
3529  010456  062737  000002  005322          ADD     #2,TFOUND       ;GET READY FOR NEXT DEVICE
3530
3531                                          ;ALL UNITS DONE? IF LTRY BECOMES ZERO YES ALL DONE
3532  010464  005337  005320                  DEC     LTRY            ;REDUCE DEVICES LEFT TO TRY
3533  010470  001402                          BEQ     7$              ;BRANCH IF ALL COMPLETE
3534  010472  000137  006526                  JMP     SS2             ;ALL NOT DONE
3535  010476                          7$:
3536  010476  013746  005322          SS3:    MOV     TFOUND,-(SP)    ;GET TWICE NUMBER OF RH FOUND
3537  010502  006216                          ASR     (SP)            ;DIVIDE BY 2
3538  010504  011637  005506                  MOV     (SP),TSTUNT
3539  010510  012637  005510                  MOV     (SP)+,WORUNT
3540  010514  000137  010622                  JMP     TST2    ;JUMP TO NEXT TEST
3541
3542                                          ;A RH TIMED OUT SO TRY NEXT UNIT OR END TRYING
3543  010520  005337  005320          SS4:    DEC     LTRY            ;RH TIMES OUT SO DECREASE RH TYPED
```

```
3544  010524  001402                          BEQ     1$              ;BRANCH IF ALL DONE
3545  010526  000137  006525                  JMP     SS2             ;ALL NOT DONE SO JUMP.
3546  010532  000137  010476          1$:     JMP     SS3             ;ALL DONE SO STORE NUMBER OF RH FOUND
3547
3548                                          ;A DRIVE TYPE DID NOT MATCH SO TRY NEXT
3549  010536  012777  040000  173314  SS5:    MOV     #TRE,@RHCS1     ;CLEAR NED ERROR
3550  010544  005337  005362          SS8:    DEC     NTYPNT          ;DECREASE NUMBER OF DRIVE TYPES TRIED
3551  010550  001402                          BEQ     1$              ;BRANCH IF ALL DONE
3552  010552  000137  006616                  JMP     SS7             ;ALL NOT DONE
3553  010556  005277  173306          1$:     INC     @RHCS2          ;GET NEXT UNIT NO.
3554  010562  012777  040000  173270          MOV     #TRE,@RHCS1     ;CLEAR NED ERROR
3555  010570  017746  173274                  MOV     @RHCS2,-(SP)    ;GET RHCS2
3556  010574  042716  177770                  BIC     #177770,(SP)    ;GET UNIT NO
3557  010600  022726  000010                  CMP     #8.,(SP)+       ;ARE 7 DONE
3558  010604  001402                          BEQ     2$              ;BRANCH IF 7 DONE
3559  010606  000137  006602                  JMP     SS6             ;7 NOT DONE SO TRY NEXT UNIT NO
3560
3561  010612  104147                  2$:     ERROR   147             ;DEVICE ADDRESS DID NOT TIME
3562  010614  000000                          HALT                    ;OUT BUT NO UNITS HAD
3563                                                                  ;APPROPRIATE DRIVE TYPE.
3564  010616  000137  010520                  JMP     SS4
3565
3566
3567
3568
3569
3570
3571
3572                                  ;;**********************************************************************
3573                                  ;*TEST 2        UNIT UNDER TEST
3574                                  ;*      THIS TYPES THE UNIT TO BE TESTED
3575                                  ;*      AND IS THE FIRST TEST AFTER THE END OF THE PROGRAM
3576                                  ;;**********************************************************************
3577  010622  000004                  TST2:   SCOPE
3578  010624  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
3579  010632  012737  037600  000004          MOV     #TIEOUT,@#ERRVEC        ;TIMEOUT VECTOR
3580  010640  012737  000340  000006          MOV     #340,@#ERRVEC+2         ;PRIORITY 7
3581
3582  010646  013746  005510                  MOV     WORUNT,-(SP)    ;GET WORKING RH NUMBER LEFT
3583  010652  013702  005506                  MOV     TSTUNT,R2       ;GET TOTAL NO OF RH FOUND
3584  010656  162602                          SUB     (SP)+,R2        ;NO OF RH TO BE TESTED
3585  010660  006302                          ASL     R2              ;INDEX FOR RH TO BE TESTED.
3586  010662  104401  010670                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3587  010666  000402                          BR      64$             ;;GET OVER THE ASCIZ
3588                                  ;;65$:  .ASCIZ  <15><12>/ /
3589  010674                          64$:
3590  010674  104401  010702                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3591  010700  000410                          BR      66$             ;;GET OVER THE ASCIZ
3592                                  ;;67$:  .ASCIZ  <15><12>/TESTING RH NO/
3593  010722                          66$:
3594  010722  016246  005436                  MOV     FRHNM(R2),-(SP) ;TYPE RH NO.
3595  010726  104405                          TYPDS
3596
3597  010730  104401  010736                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3598  010734  000410                          BR      68$             ;;GET OVER THE ASCIZ
3599                                  ;;69$:  .ASCIZ  /       USING /
```

```
3600  010756                          68$:
3601  010756  104401  010764                TYPE    ,71$            ;;TYPE ASCIZ STRING
3602  010762  000404                        BR      70$             ;;GET OVER THE ASCIZ
3603                                  ;;71$:  .ASCIZ  / BASE /
3604  010774                          70$:
3605  010774  016246  005426                MOV     FBASEA(R2),-(SP)  ;TYPE BASE ADDRESS
3606  011000  104402                        TYPOC
3607  011002  022762  000001  005366        CMP     #1,FDEVIC(R2)   ;IS IT TU
3608  011010  001017                        BNE     1$              ;BRANCH IF NOT TU
3609  011012  104401  011020                TYPE    ,73$            ;;TYPE ASCIZ STRING
3610  011016  000410                        BR      72$             ;;GET OVER THE ASCIZ
3611                                  ;;73$:  .ASCIZ  / TU16 AT SLAVE/
3612  011040                          72$:
3613  011040  016246  005446                MOV     FSLAVE(R2),-(SP)  ;TYPE SLAVE NUMBER
3614  011044  104405                        TYPDS
3615  011046  000467                        BR      6$
3616  011050  022762  000002  005366  1$:   CMP     #2,FDEVIC(R2)   ;IS IT RP DUAL PORT
3617  011056  001014                        BNE     2$              ;BRANCH IF NOT RP DUAL PORT
3618  011060  104401  011066                TYPE    ,75$            ;;TYPE ASCIZ STRING
3619  011064  000410                        BR      74$             ;;GET OVER THE ASCIZ
3620                                  ;;75$:  .ASCIZ  / RP DUAL PORT /
3621  011106                          74$:
3622  011106  000447                        BR      6$
3623  011110  022762  000003  005366  2$:   CMP     #3,FDEVIC(R2)   ;IS IT RO SINGLE PORT
3624  011116  001014                        BNE     3$              ;BRANCH IF NOT RP
3625  011120  104401  011126                TYPE    ,77$            ;;TYPE ASCIZ STRING
3626  011124  000410                        BR      76$             ;;GET OVER THE ASCIZ
3627                                  ;;77$:  .ASCIZ  / RP SINGLE PORT/
3628  011146                          76$:
3629  011146  000427                        BR      6$
3630  011150  022762  000004  005366  3$:   CMP     #4,FDEVIC(R2)   ;IS IT RS04
3631  011156  001007                        BNE     4$              ;BRANCH IF NOT RS04
3632  011160  104401  011166                TYPE    ,79$            ;;TYPE ASCIZ STRING
3633  011164  000403                        BR      78$             ;;GET OVER THE ASCIZ
3634                                  ;;79$:  .ASCIZ  / RS04/
3635  011174                          78$:
3636  011174  000414                        BR      6$
3637  011176  022762  000005  005366  4$:   CMP     #5,FDEVIC(R2)   ;IS IT RS03
3638  011204  001007                        BNE     5$              ;BRANCH IF NOT RS03
3639  011206  104401  011214                TYPE    ,81$            ;;TYPE ASCIZ STRING
3640  011212  000403                        BR      80$             ;;GET OVER THE ASCIZ
3641                                  ;;81$:  .ASCIZ  / RS03/
3642  011222                          80$:
3643  011222  000401                        BR      6$
3644  011224  000000                  5$:   HALT                    ;PROGRAM ERROR
3645
3646  011226                          6$:
3647  011226  104401  011234                TYPE    ,83$            ;;TYPE ASCIZ STRING
3648  011232  000411                        BR      82$             ;;GET OVER THE ASCIZ
3649                                  ;;83$:  .ASCIZ  <15><12>/AT UNIT NUMBER/
3650  011256                          82$:
3651  011256  016246  005376                MOV     FUNITN(R2),-(SP)  ;TYPE UNIT NUMBER
3652  011262  104405                        TYPDS
3653  011264  104401  011272                TYPE    ,85$            ;;TYPE ASCIZ STRING
3654  011270  000411                        BR      84$             ;;GET OVER THE ASCIZ
3655                                  ;;85$:  .ASCIZ  /          VECTOR /
```

```
3656  011314                          84$:
3657  011314  016246  005416                  MOV     FVECTR(R2),-(SP)  ;TYPE VECTOR
3658  011320  104402                          TYPOC
3659  011322  012703  000024                  MOV     #20.,R3           ;COUNT 20
3660  011326  012704  004060                  MOV     #RHCS1,R4         ;GET BASE LOCATION
3661  011332  016205  005426                  MOV     FBASEA(R2),R5     ;GET BASE ADDRESS
3662  011336  010524                  7$:     MOV     R5,(R4)+          ;FILL PROPER ADDRESS
3663  011340  062705  000002                  ADD     #2,R5             ;INCREMENT BY 2
3664  011344  005303                          DEC     R3                ;COUNT DOWN
3665  011346  001373                          BNE     7$                ;BRANCH IF 20 NOT DONE
3666
3667  011350  016237  005376  005010          MOV     FUNITN(R2),UNIT ;UNIT NUMBER
3668  011356  016237  005416  005512          MOV     FVECTR(R2),VECTOR  ;VECTOR
3669  011364  016237  005446  004660          MOV     FSLAVE(R2),SLAVE  ;SLAVE NO
3670  011372  016237  005456  004640          MOV     FDRIVR(R2),COMND  ;DRIVER ADDRESS
3671  011400  016237  005466  004130          MOV     FBAE(R2),RHBAE  ;BAE ADDRESS
3672  011406  016237  005476  004132          MOV     FCS3(R2),RHCS3  ;CS3 ADDRESS
3673
3674  011414  005037  045100                  CLR     PRITEM
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684                                  ;;*****************************************************************
3685                                  ;*TEST 3        BIT BANG RHCS1
3686
3687                                  ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
3688                                  ;*      IN RHCS1 REGISTER.
3689                                  ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
3690                                  ;*      AND WALKING 0'S (-2,-3,-5 ETC)
3691                                  ;*      IN THIS TEST SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO BITS WILL NOT BE WRITTEN INTO
3692                                  ;*      AND RDY!DVA BITS WILL ALWAYS BE SET
3693                                  ;*      AND BIT10 BITS WILL ALWAYS BE CLEARED
3694                                  ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
3695                                  ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
3696                                  ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
3697                                  ;*      SIGNAL.
3698
3699                                  ;;*****************************************************************
3700  011420  000004                  TST3:   SCOPE
3701  011422  012706  001000                  MOV     #STACK,SP         ;RESET STACK
3702  011426  012737  000003  004656          MOV     #3,@#TSTNM        ;SAVE TEST NUMBER
3703
3704  011434  004737  040312                  JSR     PC,@#CLDISK       ;GIVE RH INITIALIZE
3705                                                                    ;SETUP UNIT NUBER
3706                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
3707
3708  011440  012737  176201  004650          MOV     #SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO,WRTBIT      ;SC!TRE!MCPE!DVA!BIT12!B
3709  011446  012737  004200  004652          MOV     #RDY!DVA,SETBIT ;RDY!DVA ARE BITS ALWAYS SET
3710  011454  012737  002000  004654          MOV #BIT10,CLRBIT       ;BIT10 ARE BITS ALWAYS CLEARED
3711
```

```
3712                                        ;FLOAT 1'S THRU THE RHCS1 REGISTER
3713  011462  012737  011506  001110        MOV     #2$,$LPERR          ;SET LOOP ON ERROR ADDRESS
3714  011470  012705  000001                MOV     #1,R5               ;GETTING READY TO FLOAT A ONE
3715  011474  010537  001124        1$:     MOV     R5,$GDDAT           ;START WITH DATA
3716  011500  042737  176201  001124        BIC     #SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO,$GDDAT      ;CLEAR BITS NOT WRITTEN
3717  011506  032737  041400  001140  2$:   BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3718  011514  001403                        BEQ     3$                  ;BRANCH IF ANY ONE SET
3719  011516  052777  000040  172344        BIS     #CLR,@RHCS2         ;GIVE CLEAR FOR SCOPE
3720                                        ;SYNC IF IN DEBUG:
3721  011524  013777  001124  172326  3$:   MOV     $GDDAT,@RHCS1       ;WRITE RHCS1 REGISTER
3722  011532  017737  172322  001126        MOV     @RHCS1,$BDDAT       ;READ RHCS1 REGISTER
3723  011540  043737  004654  001124        BIC     CLRBIT,$GDDAT       ;CLEAR ALWAYS CLEARED BITS
3724  011546  053737  004652  001124        BIS     SETBIT,$GDDAT       ;SET ALWAYS SET BITS
3725  011554  023737  001124  001126        CMP     $GDDAT,$BDDAT       ;TEST
3726  011562  001404                        BEQ     4$                  ;BRANCH IF GOOD
3727  011564  013737  004060  001122        MOV     RHCS1,$BDADR        ;GET REGISTER ADDRESS
3728  011572  104137                        ERROR   137                 ;ONE WAS BEING FLOATED
3729                                                                    ;THRU RHCS1 REGISTER
3730                                                                    ;ON READING RHCS1 BACK
3731                                                                    ;IT DID NOT CONTAIN WHAT
3732                                                                    ;WAS EXPECTED.
3733  011574  000241                4$:     CLC                         ;CLEAR CARRY
3734  011576  006305                        ASL     R5                  ;GET 1 ONE LEFT
3735  011600  103335                        BCC     1$                  ;BRANCH IF 16 NOT DONE
3736
3737                                        ;FLOAT A ZERO THRU RHCS1 REGISTER.
3738
3739  011602  012737  011626  001110        MOV     #6$,$LPERR          ;SET LOOP BACK POINT.
3740  011610  012705  177776                MOV     #177776,R5          ;GET READY TO FLOAT A ZERO
3741  011614  010537  001124        5$:     MOV     R5,$GDDAT           ;GET READY TO WRITE DATA
3742  011620  042737  176201  001124        BIC     #SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO,$GDDAT      ;CLEAR BITS NOT WRITTEN
3743  011626  032737  041400  001140  6$:   BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3744  011634  001403                        BEQ     7$                  ;BRANCH IF ANY OF THE ABOVE SET
3745  011636  012777  000040  172224        MOV     #CLR,@RHCS2         ;CLEAR FOR SCOPE AID TO
3746                                                                    ;SYNC IF IN DEBUG
3747  011644  013777  001124  172206  7$:   MOV     $GDDAT,@RHCS1       ;WRITE INTO RH'XA.
3748  011652  017737  172202  001126        MOV     @RHCS1,$BDDAT       ;READ RHCS1
3749  011660  053737  004652  001124        BIS     SETBIT,$GDDAT       ;SET BITS ALWAYS SET
3750  011666  043737  004654  001124        BIC CLRBIT,$GDDAT           ;CLEAR BITS ALWAYS 0
3751  011674  023737  001124  001126        CMP     $GDDAT,$BDDAT       ;TEST
3752  011702  001404                        BEQ     8$                  ;BRANCH IF GOOD
3753  011704  013737  004060  001122        MOV     RHCS1,$BDADR        ;GET REGISTER ADDRESS
3754  011712  104137                        ERROR   137                 ;ZERO WAS BEING FLOATED
3755                                                                    ;THRU RHCS1 REGISTER
3756                                                                    ;ON READING IT BACK IT
3757                                                                    ;DID NOT CONTAIN WHAT
3758                                                                    ;WAS EXPECTED
3759
3760  011714  000261                8$:     SEC
3761  011716  006105                        ROL     R5
3762  011720  103735                        BCS     5$
3763
3764
3765                                        ;;*****************************************************************
3766                                        ;*TEST 4        BIT BANG RHCS2
3767
```

```
                                ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
                                ;*      IN RHCS2 REGISTER.
                                ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                                ;*      AND WALKING 0'S (-2,-3,-5 ETC)
                                ;*      IN THIS TEST 177740 BITS WILL NOT BE WRITTEN INTO
                                ;*      AND IR BITS WILL ALWAYS BE SET
                                ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                                ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                                ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                                ;*      SIGNAL.
                                ;
                                ;;*****************************************************************
011722  000004                  TST4:   SCOPE
011724  012706  001000                  MOV     #STACK,SP       ;RESET STACK
011730  012737  000004  004656          MOV     #4,@#TSTNM      ;SAVE TEST NUMBER
                                
011736  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                
011742  012737  177740  004650          MOV     #177740,WRTBIT  ;177740 ARE BITS NOT WRITTEN INTO
011750  012737  000100  004652          MOV     #IR,SETBIT      ;IR ARE BITS ALWAYS SET
                                
                                        ;FLOAT 1'S THRU THE RHCS2 REGISTER
011756  012737  012002  001110          MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
011764  012705  000001                  MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
011770  010537  001124          1$:     MOV     R5,$GDDAT       ;START WITH DATA
011774  042737  177740  001124          BIC     #177740,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
012002  032737  041400  001140  2$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012010  001403                          BEQ     3$              ;BRANCH IF ANY ONE SET
012012  052777  000040  172050          BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                                ;SYNC IF IN DEBUG:
012020  013777  001124  172042  3$:     MOV     $GDDAT,@RHCS2   ;WRITE RHCS2 REGISTER
012026  017737  172036  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 REGISTER
012034  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
012042  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
012050  001404                          BEQ     4$              ;BRANCH IF GOOD
012052  013737  004070  001122          MOV     RHCS2,$BDADR    ;GET REGISTER ADDRESS
012060  104137                          ERROR   137             ;ONE WAS BEING FLOATED
                                                                ;THRU RHCS2 REGISTER
                                                                ;ON READING RHCS2 BACK
                                                                ;IT DID NOT CONTAIN WHAT
                                                                ;WAS EXPECTED.
012062  000241          4$:     CLC                             ;CLEAR CARRY
012064  006305                  ASL     R5                      ;GET 1 ONE LEFT
012066  103340                  BCC     1$                      ;BRANCH IF 16 NOT DONE
                                
                                        ;FLOAT A ZERO THRU RHCS2 REGISTER.
                                
012070  012737  012114  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
012076  012705  177776                  MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
012102  010537  001124          5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
012106  042737  177740  001124          BIC     #177740,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
012114  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012122  001403                          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
012124  012777  000040  171736          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
```

```
3824
3825  012132  013777  001124  171730  7$:   MOV   $GDDAT,@RHCS2   ;SYNC IF IN DEBUG
3826  012140  017737  171724  001126        MOV   @RHCS2,$BDDAT   ;WRITE INTO RH'XA.
3827  012146  053737  004652  001124        BIS   SETBIT,$GDDAT   ;READ RHCS2
3828  012154  023737  001124  001126        CMP   $GDDAT,$BDDAT   ;SET BITS ALWAYS SET
3829  012162  001404                        BEQ   8$              ;TEST
3830  012164  013737  004070  001122        MOV   RHCS2,$BDADR    ;BRANCH IF GOOD
3831  012172  104137                        ERROR 137             ;GET REGISTER ADDRESS
3832                                                              ;ZERO WAS BEING FLOATED
3833                                                              ;THRU RHCS2 REGISTER
3834                                                              ;ON READING IT BACK IT
3835                                                              ;DID NOT CONTAIN WHAT
3836                                                              ;WAS EXPECTED
3837  012174  000261                8$:   SEC
3838  012176  006105                        ROL   R5
3839  012200  103740                        BCS   5$
3840
3841
3842                        ;;*****************************************************************
3843                        ;*TEST 5        BIT BANG RHCS3
3844
3845                        ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
3846                        ;*      IN RHCS3 REGISTER.
3847                        ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
3848                        ;*      AND WALKING 0'S (-2,-3,-5 ETC)
3849                        ;*      IN THIS TEST 177660 BITS WILL NOT BE WRITTEN INTO
3850                        ;*      AND 0 BITS WILL ALWAYS BE SET
3851                        ;*      AND BIT9!BIT8!BIT7!BI 5!BIT4 BITS WILL ALWAYS BE CLEARED
3852                        ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
3853                        ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
3854                        ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
3855                        ;*      SIGNAL.
3856
3857                        ;;*****************************************************************
3858  012202  000004        TST5:   SCOPE
3859  012204  012706  001000        MOV   #STACK,SP       ;RESET STACK
3860  012210  012737  000005  004656 MOV   #5,@#TSTNM      ;SAVE TEST NUMBER
3861
3862  012216  004737  040312        JSR   PC,@#CLDISK     ;GIVE RH INITIALIZE
3863                                                      ;SETUP UNIT NUBER
3864                                                      ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
3865
3866  012222  012737  177660  004650 MOV   #177660,WRTBIT  ;177660 ARE BITS NOT WRITTEN INTO
3867  012230  012737  000000  004652 MOV   #0,SETBIT       ;0 ARE BITS ALWAYS SET
3868  012236  012737  001660  004654 MOV #BIT9!BIT8!BIT7!BIT5!BIT4,CLRBIT    ;BIT9!BIT8!BIT7!BIT5!BIT4 ARE BITS ALWAY
3869
3870                                 ;FLOAT 1'S THRU THE RHCS3 REGISTER
3871  012244  012737  012270  001110 MOV   #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
3872  012252  012705  000001        MOV   #1,R5           ;GETTING READY TO FLOAT A ONE
3873  012256  010537  001124  1$:   MOV   R5,$GDDAT       ;START WITH DATA
3874  012262  042737  177660  001124 BIC   #177660,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
3875  012270  032737  041400  001140 2$:   BIT   #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3876  012276  001403                BEQ   3$              ;BRANCH IF ANY ONE SET
3877  012300  052777  000040  171562 BIS   #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
3878                                                      ;SYNC IF IN DEBUG:
3879  012306  013777  001124  171616 3$:   MOV   $GDDAT,@RHCS3   ;WRITE RHCS3 REGISTER
```

```
3880  012314  017737  171612  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 REGISTER
3881  012322  043737  004654  001124          BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
3882  012330  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
3883  012336  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
3884  012344  001404                          BEQ     4$              ;BRANCH IF GOOD
3885  012346  013737  004132  001122          MOV     RHCS3,$BDADR    ;GET REGISTER ADDRESS
3886  012354  104137                          ERROR   137             ;ONE WAS BEING FLOATED
3887                                                                  ;THRU RHCS3 REGISTER
3888                                                                  ;ON READING RHCS3 BACK
3889                                                                  ;IT DID NOT CONTAIN WHAT
3890                                                                  ;WAS EXPECTED.
3891  012356  000241                  4$:     CLC                     ;CLEAR CARRY
3892  012360  006305                          ASL     R5              ;GET 1 ONE LEFT
3893  012362  103335                          BCC     1$              ;BRANCH IF 16 NOT DONE
3894
3895                                          ;FLOAT A ZERO THRU RHCS3 REGISTER.
3896
3897  012364  012737  012410  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
3898  012372  012705  177776                  MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
3899  012376  010537  001124          5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
3900  012402  042737  177660  001124          BIC     #177660,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
3901  012410  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3902  012416  001403                          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
3903  012420  012777  000040  171442          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
3904                                                                  ;SYNC IF IN DEBUG
3905  012426  013777  001124  171476  7$:     MOV     $GDDAT,@RHCS3   ;WRITE INTO RH'XA.
3906  012434  017737  171472  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3
3907  012442  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
3908  012450  043737  004654  001124          BIC CLRBIT,$GDDAT       ;CLEAR BITS ALWAYS 0
3909  012456  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
3910  012464  001404                          BEQ     8$              ;BRANCH IF GOOD
3911  012466  013737  004132  001122          MOV     RHCS3,$BDADR    ;GET REGISTER ADDRESS
3912  012474  104137                          ERROR   137             ;ZERO WAS BEING FLOATED
3913                                                                  ;THRU RHCS3 REGISTER
3914                                                                  ;ON READING IT BACK IT
3915                                                                  ;DID NOT CONTAIN WHAT
3916                                                                  ;WAS EXPECTED
3917
3918  012476  000261                  8$:     SEC
3919  012500  006105                          ROL     R5
3920  012502  103735                          BCS     5$
3921
3922
3923                                  ;;*****************************************************************
3924                                  ;*TEST 6         BIT BANG RHWC
3925
3926                                  ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
3927                                  ;*      IN RHWC REGISTER.
3928                                  ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
3929                                  ;*      AND WALKING 0'S (-2,-3,-5 ETC)
3930                                  ;*      IN THIS TEST 0 BITS WILL NOT BE WRITTEN INTO
3931                                  ;*      AND 0 BITS WILL ALWAYS BE SET
3932                                  ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
3933                                  ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
3934                                  ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
3935                                  ;*      SIGNAL.
```

```
                                ;;*****************************************************************
012504  000004                TSTS:   SCOPE
012506  012706  001000                MOV     #STACK,SP       ;RESET STACK
012512  012737  000006  004656        MOV     #6,@#TSTNM      ;SAVE TEST NUMBER
                                
012520  004737  040312                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                              ;SETUP UNIT NUBER
                                                              ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                
012524  012737  000000  004650        MOV     #0,WRTBIT       ;0 ARE BITS NOT WRITTEN INTO
012532  012737  000000  004652        MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
                                
                                        ;FLOAT 1'S THRU THE RHWC REGISTER
012540  012737  012564  001110        MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
012546  012705  000001                MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
012552  010537  001124        1$:     MOV     R5,$GDDAT       ;START WITH DATA
012556  042737  000000  001124        BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
012564  032737  041400  001140  2$:   BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012572  001403                        BEQ     3$              ;BRANCH IF ANY ONE SET
012574  052777  000040  171266        BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                              ;SYNC IF IN DEBUG:
012602  013777  001124  171252  3$:   MOV     $GDDAT,@RHWC    ;WRITE RHWC REGISTER
012610  017737  171246  001126        MOV     @RHWC,$BDDAT    ;READ RHWC REGISTER
012616  053737  004652  001124        BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
012624  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;TEST
012632  001404                        BEQ     4$              ;BRANCH IF GOOD
012634  013737  004062  001122        MOV     RHWC,$BDADR     ;GET REGISTER ADDRESS
012642  104137                        ERROR   137             ;ONE WAS BEING FLOATED
                                                              ;THRU RHWC REGISTER
                                                              ;ON READING RHWC BACK
                                                              ;IT DID NOT CONTAIN WHAT
                                                              ;WAS EXPECTED.
012644  000241                4$:     CLC                     ;CLEAR CARRY
012646  006305                        ASL     R5              ;GET 1 ONE LEFT
012650  103340                        BCC     1$              ;BRANCH IF 16 NOT DONE
                                
                                        ;FLOAT A ZERO THRU RHWC REGISTER.
                                
012652  012737  012676  001110        MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
012660  012705  177776                MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
012664  010537  001124        5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
012670  042737  000000  001124        BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
012676  032737  041400  001140  6$:   BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012704  001403                        BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
012706  012777  000040  171154        MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
                                                              ;SYNC IF IN DEBUG
012714  013777  001124  171140  7$:   MOV     $GDDAT,@RHWC    ;WRITE INTO RH'XA.
012722  017737  171134  001126        MOV     @RHWC,$BDDAT    ;READ RHWC
012730  053737  004652  001124        BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
012736  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;TEST
012744  001404                        BEQ     8$              ;BRANCH IF GOOD
012746  013737  004062  001122        MOV     RHWC,$BDADR     ;GET REGISTER ADDRESS
012754  104137                        ERROR   137             ;ZERO WAS BEING FLOATED
                                                              ;THRU RHWC REGISTER
                                                              ;ON READING IT BACK IT
```

```
                                                        ;DID NOT CONTAIN WHAT
                                                        ;WAS EXPECTED

012756  000261                8$:     SEC
012760  006105                        ROL     R5
012762  103740                        BCS     5$

                        ;;**************************************************************
                        ;*TEST 7         BIT BANG RHBA

                        ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
                        ;*      IN RHBA REGISTER.
                        ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                        ;*      AND WALKING 0'S (-2,-3,-5 ETC)
                        ;*      IN THIS TEST 0 BITS WILL NOT BE WRITTEN INTO
                        ;*      AND 0 BITS WILL ALWAYS BE SET
                        ;*      AND BIT00 BITS WILL ALWAYS BE CLEARED
                        ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                        ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                        ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                        ;*      SIGNAL.

                        ;;**************************************************************
012764  000004          TST7:   SCOPE
012766  012706  001000          MOV     #STACK,SP       ;RESET STACK
012772  012737  000007  004656  MOV     #7,@#TSTNM      ;SAVE TEST NUMBER

013000  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

013004  012737  000000  004650  MOV     #0,WRTBIT       ;0 ARE BITS NOT WRITTEN INTO
013012  012737  000000  004652  MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
013020  012737  000001  004654  MOV  #BIT00,CLRBIT       ;BIT00 ARE BITS ALWAYS CLEARED

                        ;FLOAT 1'S THRU THE RHBA REGISTER
013026  012737  013052  001110  MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
013034  012705  000001          MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
013040  010537  001124  1$:     MOV     R5,$GDDAT       ;START WITH DATA
013044  042737  000000  001124  BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
013052  032737  041400  001140  2$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
013060  001403                  BEQ     3$              ;BRANCH IF ANY ONE SET
013062  052777  000040  171000  BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                        ;SYNC IF IN DEBUG:
013070  013777  001124  170766  3$:     MOV     $GDDAT,@RHBA    ;WRITE RHBA REGISTER
013076  017737  170762  001126  MOV     @RHBA,$BDDAT    ;READ RHBA REGISTER
013104  043737  004654  001124  BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
013112  053737  004652  001124  BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
013120  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;TEST
013126  001404                  BEQ     4$              ;BRANCH IF GOOD
013130  013737  004064  001122  MOV     RHBA,$BDADR     ;GET REGISTER ADDRESS
013136  104137                  ERROR   137             ;ONE WAS BEING FLOATED
                                                        ;THRU RHBA REGISTER
                                                        ;ON READING RHBA BACK
                                                        ;IT DID NOT CONTAIN WHAT
```

```
                                                        ;WAS EXPECTED.
013140  000241          4$:     CLC                    ;CLEAR CARRY
013142  006305                  ASL     R5             ;GET 1 ONE LEFT
013144  103335                  BCC     1$             ;BRANCH IF 16 NOT DONE

                        ;FLOAT A ZERO THRU RHBA REGISTER.

013146  012737  013172  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
013154  012705  177776          MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
013160  010537  001124  5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
013164  042737  000000  001124  BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
013172  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
013200  001403          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
013202  012777  000040  170660          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
                                                        ;SYNC IF IN DEBUG
013210  013777  001124  170646  7$:     MOV     $GDDAT,@RHBA    ;WRITE INTO RH'XA.
013216  017737  170642  001126          MOV     @RHBA,$BDDAT    ;READ RHBA
013224  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
013232  043737  004654  001124          BIC     CLRBIT,$GDDAT   ;CLEAR BITS ALWAYS 0
013240  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
013246  001404          BEQ     8$              ;BRANCH IF GOOD
013250  013737  004064  001122          MOV     RHBA,$BDADR     ;GET REGISTER ADDRESS
013256  104137          ERROR   137             ;ZERO WAS BEING FLOATED
                                                        ;THRU RHBA REGISTER
                                                        ;ON READING IT BACK IT
                                                        ;DID NOT CONTAIN WHAT
                                                        ;WAS EXPECTED

013260  000261          8$:     SEC
013262  006105                  ROL     R5
013264  103735                  BCS     5$

                        ;;*******************************************************************
                        ;*TEST 10       BIT BANG RHBAE

                        ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
                        ;*      IN RHBAE REGISTER.
                        ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
                        ;*      AND WALKING 0'S (-2,-3,-5 ETC)
                        ;*      IN THIS TEST 177700 BITS WILL NOT BE WRITTEN INTO
                        ;*      AND 0 BITS WILL ALWAYS BE SET
                        ;*      AND 177700 BITS WILL ALWAYS BE CLEARED
                        ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
                        ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
                        ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
                        ;*      SIGNAL.

                        ;;*******************************************************************
013266  000004          TST10:  SCOPE
013270  012706  001000          MOV     #STACK,SP       ;RESET STACK
013274  012737  000010  004656  MOV     #10,@#TSTNM     ;SAVE TEST NUMBER

013302  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
```

```
4104
4105  013306  012737  177700  004650          MOV     #177700,WRTBIT  ;177700 ARE BITS NOT WRITTEN INTO
4106  013314  012737  000000  004652          MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
4107  013322  012737  177700  004654          MOV #177700,CLRBIT      ;177700 ARE BITS ALWAYS CLEARED
4108
4109                                          ;FLOAT 1'S THRU THE RHBAE REGISTER
4110  013330  012737  013354  001110          MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
4111  013336  012705  000001                  MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
4112  013342  010537  001124          1$:     MOV     R5,$GDDAT       ;START WITH DATA
4113  013346  042737  177700  001124          BIC     #177700,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
4114  013354  032737  041400  001140  2$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
4115  013362  001403                          BEQ     3$              ;BRANCH IF ANY ONE SET
4116  013364  052777  000040  170476          BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
4117                                          ;SYNC IF IN DEBUG:
4118  013372  013777  001124  170530  3$:     MOV     $GDDAT,@RHBAE   ;WRITE RHBAE REGISTER
4119  013400  017737  170524  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE REGISTER
4120  013406  043737  004654  001124          BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
4121  013414  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
4122  013422  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
4123  013430  001404                          BEQ     4$              ;BRANCH IF GOOD
4124  013432  013737  004130  001122          MOV     RHBAE,$BDADR    ;GET REGISTER ADDRESS
4125  013440  104137                          ERROR   137             ;ONE WAS BEING FLOATED
4126                                          ;THRU RHBAE REGISTER
4127                                          ;ON READING RHBAE BACK
4128                                          ;IT DID NOT CONTAIN WHAT
4129                                          ;WAS EXPECTED.
4130  013442  000241                  4$:     CLC                     ;CLEAR CARRY
4131  013444  006305                          ASL     R5              ;GET 1 ONE LEFT
4132  013446  103335                          BCC     1$              ;BRANCH IF 16 NOT DONE
4133
4134                                          ;FLOAT A ZERO THRU RHBAE REGISTER.
4135
4136  013450  012737  013474  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
4137  013456  012705  177776                  MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
4138  013462  010537  001124          5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
4139  013466  042737  177700  001124          BIC     #177700,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
4140  013474  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
4141  013502  001403                          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
4142  013504  012777  000040  170356          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
4143                                          ;SYNC IF IN DEBUG
4144  013512  013777  001124  170410  7$:     MOV     $GDDAT,@RHBAE   ;WRITE INTO RH'XA.
4145  013520  017737  170404  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE
4146  013526  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
4147  013534  043737  004654  001124          BIC CLRBIT,$GDDAT       ;CLEAR BITS ALWAYS 0
4148  013542  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
4149  013550  001404                          BEQ     8$              ;BRANCH IF GOOD
4150  013552  013737  004130  001122          MOV     RHBAE,$BDADR    ;GET REGISTER ADDRESS
4151  013560  104137                          ERROR   137             ;ZERO WAS BEING FLOATED
4152                                          ;THRU RHBAE REGISTER
4153                                          ;ON READING IT BACK IT
4154                                          ;DID NOT CONTAIN WHAT
4155                                          ;WAS EXPECTED
4156
4157  013562  000261                  8$:     SEC
4158  013564  006105                          ROL     R5
4159  013566  103735                          BCS     5$
```

```
                                ;*****************************************************************
                                ;*TEST 11       SILO TEST 1 (ONE WORD WRITE)
                                ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH
                                ;*      TOGETHER WITH UNIT NUMBER
                                ;*      ONE WORD OF ALL ZEROS IS WRITTEN INTO RHDB
                                ;*      BY "CLR"
                                ;*      "OR" (BIT #7) IS WAITED FOR BY A TRAP INSTRUCTION
                                ;*      CALLED "WAT" (NO TIMING IS DONE)
                                ;*      HOWEVER IF "OR" DOES NOT SET WITHIN "WAT" COUNT
                                ;*      DOWN AN ERROR IS REPORTED
                                ;*      RHDB IS READ AND CHECKED TO CONTAIN ZEROS
                                ;*      RHCS2 WILL BE CHECKED TO HAVE "IR" AND UNIT NUMBER
                                ;*      RHCS1,RHCS3,RHBA,RHBAE,RHWC, WILL BE CHECKED TO HAVE
                                ;*      APPROPIATE VALUES
                                ;*****************************************************************
013570  000004                 TST11:  SCOPE
013572  012706  001000                 MOV     #STACK,SP       ;RESET STACK
013576  012737  000011  004656         MOV     #11,@#†STNM     ;SAVE TEST NUMBER

013604  004737  040312                 JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS2 HAS IR
013610  012737  000100  001124         MOV     #IR,$GDDAT      ;GET GOOD = 100
013616  053737  005010  001124         BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

013624  017737  170240  001126         MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
013632  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
013640  001401                         BEQ     65$             ;BRANCH IF GOOD
013642  104006                         ERROR   6
                                                                ;AFTER SETTING CLR BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND HAVING DONE NOTHING ELSE
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
013644                          65$:

                                        ;WRITE ONE WORD OF ALL ZEROS INTO RHDB
013644  005077  170232                 CLR     @RHDB           ;WRITE 0 IN RHDB

                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
013650  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
013652  004070                         RHCS2                   ;AND WAIT FOR OR BIT IN
```

```
4216  013654  000200                          OR                      ;RHCS2 REGISTER
4217                                                                  ;IF ERROR OCCURS HERE
4218                                                                  ;IT MEANS "OR" DID NOT
4219                                                                  ;SET FOR THE FULL COUNT
4220                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
4221                                                                  ;THIS TIME IS APPROXIMATELY
4222                                                                  ;LARGER THAN 500 MILLISECONDS
4223
4224
4225                                          ;CHECK THAT RHDB HAS 0
4226  013656  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =
4227
4228  013664  017737  170212  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
4229  013672  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4230                                                                  ;DATA WITH DATA READ FROM
4231                                                                  ;RHDB
4232  013700  001401                          BEQ     67$             ;BRANCH IF GOOD
4233  013702  104007                          ERROR   7
4234                                                                  ;AFTER CLEARING THE RH
4235                                                                  ;AND WRITING ALL ZEROS
4236                                                                  ;INTO RHDB
4237                                                                  ;THEN READING IT BACK
4238                                                                  ;RHDB SHOULD HAVE 0
4239                                                                  ;=
4240                                                                  ;BUT CONTAINED WHAT IS
4241                                                                  ;GIVEN IN BAD RHDB
4242  013704                          67$:
4243                                          ;CHECK THAT RHCS2 HAS IR
4244  013704  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
4245  013712  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4246
4247  013720  017737  170144  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4248  013726  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4249                                                                  ;DATA WITH DATA READ FROM
4250                                                                  ;RHCS2
4251  013734  001401                          BEQ     69$             ;BRANCH IF GOOD
4252  013736  104010                          ERROR   10
4253                                                                  ;AFTER CLEARING THE RH
4254                                                                  ;AND WRITING ALL ZEROS
4255                                                                  ;INTO RHDB
4256                                                                  ;THEN READING IT BACK
4257                                                                  ;RHCS2 SHOULD HAVE IR
4258                                                                  ;=100
4259                                                                  ;TOGETHER WITH UNIT NUMBER
4260                                                                  ;BUT CONTAINED WHAT IS
4261                                                                  ;GIVEN IN BAD RHCS2
4262  013740                          69$:
4263                                          ;CHECK THAT RHCS1 HAS DVA!RDY
4264  013740  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
4265
4266  013746  017737  170106  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
4267  013754  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4268                                                                  ;DATA WITH DATA READ FROM
4269                                                                  ;RHCS1
4270  013762  001401                          BEQ     71$             ;BRANCH IF GOOD
4271  013764  104011                          ERROR   11
```

```
                                                                        ;AFTER CLEARING THE RH
                                                                        ;AND WRITING ALL ZEROS
                                                                        ;INTO RHDB
                                                                        ;THEN READING IT BACK
                                                                        ;RHCS1 SHOULD HAVE DVA!RDY
                                                                        ;=4200
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS1
013766                                  71$:
                                                ;CHECK THAT RHCS3 HAS 0
013766  012737  000000  001124                 MOV     #0,$GDDAT       ;GET GOOD = 0

013774  017737  170132  001126                 MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
014002  023737  001124  001126                 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS3
014010  001401                                 BEQ     73$             ;BRANCH IF GOOD
014012  104012                                 ERROR   12

                                                                        ;AFTER CLEARING THE RH
                                                                        ;AND WRITING ALL ZEROS
                                                                        ;INTO RHDB
                                                                        ;THEN READING IT BACK
                                                                        ;RHCS3 SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS3
014014                                  73$:
                                                ;CHECK THAT RHBA HAS 0
014014  012737  000000  001124                 MOV     #0,$GDDAT       ;GET GOOD =

014022  017737  170036  001126                 MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
014030  023737  001124  001126                 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBA
014036  001401                                 BEQ     75$             ;BRANCH IF GOOD
014040  104013                                 ERROR   13

                                                                        ;AFTER CLEARING THE RH
                                                                        ;AND WRITING ALL ZEROS
                                                                        ;INTO RHDB
                                                                        ;THEN READING IT BACK
                                                                        ;RHBA SHOULD HAVE 0
                                                                        ;=
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBA
014042                                  75$:
                                                ;CHECK THAT RHBAE HAS 0
014042  012737  000000  001124                 MOV     #0,$GDDAT       ;GET GOOD =

014050  017737  170054  001126                 MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
014056  023737  001124  001126                 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBAE
014064  001401                                 BEQ     77$             ;BRANCH IF GOOD
014066  104014                                 ERROR   14

                                                                        ;AFTER CLEARING THE RH
                                                                        ;AND WRITING ALL ZEROS
                                                                        ;INTO RHDB
```

```
                                                                ;THEN READING IT BACK
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;=
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
014070                                  77$:
                                                ;CHECK THAT RHWC HAS 0
014070  012737  000000  001124                 MOV     #0,$GDDAT       ;GET GOOD =

014076  017737  167760  001126                 MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
014104  023737  001124  001126                 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
014112  001401                                 BEQ     79$             ;BRANCH IF GOOD
014114  104015                                 ERROR   15

                                                                ;AFTER CLEARING THE RH
                                                                ;AND WRITING ALL ZEROS
                                                                ;INTO RHDB
                                                                ;THEN READING IT BACK
                                                                ;RHWC SHOULD HAVE 0
                                                                ;=
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
014116                                  79$:


                                ;;*****************************************************************
                                ;*TEST 12        SILO TEST 2 (ONE WORD WRITE)
                                ;*       AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                ;*       RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH
                                ;*       TOGETHER WITH UNIT NUMBER
                                ;*       ONE WORD OF ALL ONES IS WRITTEN INTO RHDB
                                ;*       BY "CLR"
                                ;*       "OR" (BIT #7) IS WAITED FOR BY A TRAP INSTRUCTION
                                ;*       CALLED "WAT" (NO TIMING IS DONE)
                                ;*       HOWEVER IF "OR" DOES NOT SET WITHIN "WAT" COUNT
                                ;*       DOWN AN ERROR IS REPORTED
                                ;*       RHDB IS READ AND CHECKED TO CONTAIN ALL ONES
                                ;*       RHCS2 WILL BE CHECKED TO HAVE "IR" AND UNIT NUMBER
                                ;*       RHCS1,RHCS3,RHBA,RHBAE,RHWC, WILL BE CHECKED TO HAVE
                                ;*       APPROPIATE VALUES
                                ;;*****************************************************************
014116  000004                 TST12:  SCOPE
014120  012706  001000                  MOV     #STACK,SP       ;RESET STACK
014124  012737  000012  004656         MOV     #12,@#TSTNM     ;SAVE TEST NUMBER

014132  004737  040312                 JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                                ;CHECK THAT RHCS2 HAS IR
014136  012737  000100  001124                 MOV     #IR,$GDDAT      ;GET GOOD = 100
014144  053737  005010  001124                 BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
4384  014152  017737  167712  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4385  014160  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4386                                                                  ;DATA WITH DATA READ FROM
4387                                                                  ;RHCS2
4388  014166  001401                          BEQ     65$             ;BRANCH IF GOOD
4389  014170  104006                          ERROR   6
4390                                                                  ;AFTER SETTING CLR BIT #5
4391                                                                  ;IN RHCS2 TO INIT THE RH
4392                                                                  ;AND HAVING DONE NOTHING ELSE
4393                                                                  ;RHCS2 SHOULD HAVE IR
4394                                                                  ;=100
4395                                                                  ;TOGETHER WITH UNIT NUMBER
4396                                                                  ;BUT CONTAINED WHAT IS
4397                                                                  ;GIVEN IN BAD RHCS2
4398  014172                          65$:
4399
4400                                          ;WRITE ONE WORD OF ALL ONES INTO RHDB
4401  014172  012777  177777  167702          MOV     #177777,@RHDB   ;WRITE 1 IN RHDB
4402
4403                                          ;WAIT FOR OR BIT IN.RHCS2 REGISTER TO SET
4404  014200  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
4405  014202  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
4406  014204  000200                          OR                      ;RHCS2 REGISTER
4407                                                                  ;IF ERROR OCCURS HERE
4408                                                                  ;IT MEANS "OR" DID NOT
4409                                                                  ;SET FOR THE FULL COUNT
4410                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
4411                                                                  ;THIS TIME IS APPROXIMATELY
4412                                                                  ;LARGER THAN 500 MILLISECONDS
4413
4414
4415                                          ;CHECK THAT RHDB HAS 177777
4416  014206  012737  177777  001124          MOV     #177777,$GDDAT  ;GET GOOD = 0
4417
4418  014214  017737  167662  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
4419  014222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4420                                                                  ;DATA WITH DATA READ FROM
4421                                                                  ;RHDB
4422  014230  001401                          BEQ     67$             ;BRANCH IF GOOD
4423  014232  104007                          ERROR   7
4424                                                                  ;AFTER CLEARING THE RH
4425                                                                  ;AND WRITING ALL ONES
4426                                                                  ;INTO RHDB
4427                                                                  ;THEN READING IT BACK
4428                                                                  ;RHDB SHOULD HAVE 177777
4429                                                                  ;BUT CONTAINED WHAT IS
4430                                                                  ;GIVEN IN BAD RHDB
4431  014234                          67$:
4432                                          ;CHECK THAT RHCS2 HAS IR
4433  014234  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
4434  014242  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4435
4436  014250  017737  167614  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4437  014256  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4438                                                                  ;DATA WITH DATA READ FROM
4439                                                                  ;RHCS2
```

```
4440  014264  001401                        BEQ     69$             ;BRANCH IF GOOD
4441  014266  104010                        ERROR   10
4442                                                                  ;AFTER CLEARING THE RH
4443                                                                  ;AND WRITING ALL ONES
4444                                                                  ;INTO RHDB
4445                                                                  ;THEN READING IT BACK
4446                                                                  ;RHCS2 SHOULD HAVE IR
4447                                                                  ;=100
4448                                                                  ;TOGETHER WITH UNIT NUMBER
4449                                                                  ;BUT CONTAINED WHAT IS
4450                                                                  ;GIVEN IN BAD RHCS2
4451  014270                          69$:
4452                                          ;CHECK THAT RHCS3 HAS 0
4453  014270  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =
4454
4455  014276  017737  167630  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
4456  014304  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4457                                                                  ;DATA WITH DATA READ FROM
4458                                                                  ;RHCS3
4459  014312  001401                          BEQ     71$             ;BRANCH IF GOOD
4460  014314  104012                          ERROR   12
4461                                                                  ;AFTER CLEARING THE RH
4462                                                                  ;AND WRITING ALL ONES
4463                                                                  ;INTO RHDB
4464                                                                  ;THEN READING IT BACK
4465                                                                  ;RHCS3 SHOULD HAVE 0
4466                                                                  ;=
4467                                                                  ;BUT CONTAINED WHAT IS
4468                                                                  ;GIVEN IN BAD RHCS3
4469  014316                          71$:
4470                                          ;CHECK THAT RHBA HAS 0
4471  014316  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =
4472
4473  014324  017737  167534  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
4474  014332  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4475                                                                  ;DATA WITH DATA READ FROM
4476                                                                  ;RHBA
4477  014340  001401                          BEQ     73$             ;BRANCH IF GOOD
4478  014342  104013                          ERROR   13
4479                                                                  ;AFTER CLEARING THE RH
4480                                                                  ;AND WRITING ALL ONES
4481                                                                  ;INTO RHDB
4482                                                                  ;THEN READING IT BACK
4483                                                                  ;RHBA SHOULD HAVE 0
4484                                                                  ;=
4485                                                                  ;BUT CONTAINED WHAT IS
4486                                                                  ;GIVEN IN BAD RHBA
4487  014344                          73$:
4488                                          ;CHECK THAT RHBAE HAS 0
4489  014344  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =
4490
4491  014352  017737  167552  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
4492  014360  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4493                                                                  ;DATA WITH DATA READ FROM
4494                                                                  ;RHBAE
4495  014366  001401                          BEQ     75$             ;BRANCH IF GOOD
```

```
014370  104014                          ERROR   14
                                        ;AFTER CLEARING THE RH
                                        ;AND WRITING ALL ONES
                                        ;INTO RHDB
                                        ;THEN READING IT BACK
                                        ;RHBAE SHOULD HAVE 0
                                        ;=
                                        ;BUT CONTAINED WHAT IS
                                        ;GIVEN IN BAD RHBAE
014372                          75$:
                                        ;CHECK THAT RHWC HAS 0
014372  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =

014400  017737  167456  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
014406  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                        ;DATA WITH DATA READ FROM
                                        ;RHWC
014414  001401                          BEQ     77$             ;BRANCH IF GOOD
014416  104015                          ERROR   15

                                        ;AFTER CLEARING THE RH
                                        ;AND WRITING ALL ONES
                                        ;INTO RHDB
                                        ;THEN READING IT BACK
                                        ;RHWC SHOULD HAVE 0
                                        ;=
                                        ;BUT CONTAINED WHAT IS
                                        ;GIVEN IN BAD RHWC
014420                          77$:


                                        ;;*****************************************************************
                                        ;*TEST 13       SILO TEST 3 (TWO WORD WRITE)

                                        ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                        ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                        ;*      TOGETHER WITH UNIT NUMBER
                                        ;*      ONE WORD = 52525 IS WRITTEN INTO RHDB
                                        ;*      RHCS2 IS CHECKED TO HAVE "IR" AND UNIT NUMBER
                                        ;*      A SECOND WORD = 12525 IS WRITTEN INTO RHDB
                                        ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                        ;*      INSTRUCTION CALLED "WAT"
                                        ;*      RHDB IS READ AND CHECKED TO CONTAIN 52525
                                        ;*      RHCS2 IS CHECKED TO HAVE "IR", "OR" AND UNIT NUMBER
                                        ;*      RHDB IS READ A SECOND TIME AND CHECKED TO
                                        ;*      CONTAIN 125252
                                        ;*      RHCS2 IS CHECKED TO HAVE "IR" AND UNIT NUMBER
                                        ;*      THEN ALL REGISTERS RHCS1, RHCS3, RHBA, RHBAE, RHWC
                                        ;*      ARE CHECKED TO HAVE APPROPIATE VALUE
                                        ;*
                                        ;*****************************************************************
014420  000004                  TST13:  SCOPE
014422  012706  001000                  MOV     #STACK,SP       ;RESET STACK
014426  012737  000013  004656          MOV     #13,@#TSTNM     ;SAVE TEST NUMBER

014434  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
```

```
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS2 HAS IR
014440  012737  000100  001124         MOV     #IR,$GDDAT      ;GET GOOD = 100
014446  053737  005010  001124         BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

014454  017737  167410  001126         MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
014462  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
014470  001401                         BEQ     65$             ;BRANCH IF GOOD
014472  104006                         ERROR   6
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND HAVING DONE NOTHING ELSE

                                                        ;RHCS2 SHOULD HAVE IR
                                                        ;=100
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
014474                          65$:

                                        ;WRITE ONE WORD = 52525 INTO RHDB
014474  012777  052525  167400         MOV     #52525,@RHDB    ;WRITE IN RHDB

                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
014502  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
014504  004070                         RHCS2                   ;AND WAIT FOR OR BIT IN
014506  000200                         OR                      ;RHCS2 REGISTER
                                                        ;IF ERROR OCCURS HERE
                                                        ;IT MEANS "OR" DID NOT
                                                        ;SET FOR THE FULL COUNT
                                                        ;DOWN OF THE WAIT.T SUBROUTINE
                                                        ;THIS TIME IS APPROXIMATELY
                                                        ;LARGER THAN 500 MILLISECONDS

                                        ;CHECK THAT RHCS2 HAS IR!OR
014510  012737  000300  001124         MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
014516  053737  005010  001124         BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

014524  017737  167340  001126         MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
014532  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
014540  001401                         BEQ     67$             ;BRANCH IF GOOD
014542  104016                         ERROR   16
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;THEN WRITING 52525 INTO
                                                        ;RHDB
                                                        ;RHCS2 SHOULD HAVE IR!OR
                                                        ;=300
                                                        ;TOGETHER WITH UNIT NUMBER
```

```
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
014544                          67$:

                                        ;WRITE A SECOND WORD = 125252 INTO RHDB
014544  012777  125252  167330          MOV     #125252,@RHDB   ;WRITE 125252 INTO RHDB

                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
014552  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
014554  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
014556  000200                          OR                      ;RHCS2 REGISTER
                                                        ;IF ERROR OCCURS HERE
                                                        ;IT MEANS "OR" DID NOT
                                                        ;SET FOR THE FULL COUNT
                                                        ;DOWN OF THE WAIT.T SUBROUTINE
                                                        ;THIS TIME IS APPROXIMATELY
                                                        ;LARGER THAN 500 MILLISECONDS

                                        ;CHECK THAT RHDB HAS 52525
014560  012737  052525  001124          MOV     #52525,$GDDAT   ;GET GOOD = 0

014566  017737  167310  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
014574  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
014602  001401                          BEQ     69$             ;BRANCH IF GOOD
014604  104017                          ERROR   17
                                                        ;AFTERCLEARING THE RH THEN
                                                        ;WRITING TWO WORDS INTO
                                                        ;RHDB 52525 ND 125252
                                                        ;THEN READING RHDB ONCE
                                                        ;RHDB SHOULD HAVE 52525
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
014606                          69$:
                                        ;CHECK THAT RHCS2 HAS IR!OR
014606  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
014614  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

014622  017737  167242  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
014630  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
014636  001401                          BEQ     71$             ;BRANCH IF GOOD
014640  104020                          ERROR   20
                                                        ;AFTERCLEARING THE RH THEN
                                                        ;WRITING TWO WORDS INTO
                                                        ;RHDB 52525 ND 125252
                                                        ;THEN READING RHDB ONCE
                                                        ;RHCS2 SHOULD HAVE IR!OR
                                                        ;=300
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
014642                          71$:
```

```
                                        ;READ RHDB A SECOND TIME
                                        ;CHECK THAT RHDB HAS 125252
014642  012737  125252  001124          MOV     #125252,$GDDAT  ;GET GOOD = D

014650  017737  167226  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
014656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
014664  001401                          BEQ     73$             ;BRANCH IF GOOD
014666  104021                          ERROR   21
                                                                ;AFTER CLEARING RH THEN
                                                                ;WRITING TWO WORDS INTO
                                                                ;RHDB 52525 AND 125252
                                                                ;THEN READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHDB SHOULD HAVE 125252
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
014670                          73$:
                                        ;CHECK THAT RHCS2 HAS IR
014670  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
014676  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

014704  017737  167160  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
014712  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
014720  001401                          BEQ     75$             ;BRANCH IF GOOD
014722  104022                          ERROR   22
                                                                ;AFTER CLEARING RH THEN
                                                                ;WRITING TWO WORDS INTO
                                                                ;RHDB 52525 AND 125252
                                                                ;THEN READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
014724                          75$:
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
014724  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

014732  017737  167122  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
014740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
014746  001401                          BEQ     77$             ;BRANCH IF GOOD
014750  104141                          ERROR   141
                                                                ;AFTER CLEARING RH THEN
                                                                ;WRITING TWO WORDS INTO
                                                                ;RHDB 52525 AND 125252
                                                                ;THEN READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHCS1 SHOULD HAVE DVA!RDY
```

```
4720                                                                    ;=4200
4721                                                                    ;BUT CONTAINED WHAT IS
4722                                                                    ;GIVEN IN BAD RHCS1
4723  014752                                77$:
4724                                              ;CHECK THAT RHCS3 HAS 0
4725  014752  012737  000000  001124              MOV     #0,$GDDAT       ;GET GOOD = 0
4726
4727  014760  017737  167146  001126              MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
4728  014766  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4729                                                                    ;DATA WITH DATA READ FROM
4730                                                                    ;RHCS3
4731  014774  001401                            BEQ     79$             ;BRANCH IF GOOD
4732  014776  104142                            ERROR   142
4733                                                                    ;AFTER CLEARING RH THEN
4734                                                                    ;WRITING TWO WORDS INTO
4735                                                                    ;RHDB 52525 AND 125252
4736                                                                    ;THEN READING RHDB THE
4737                                                                    ;SECOND TIME
4738                                                                    ;RHCS3 SHOULD HAVE 0
4739                                                                    ;BUT CONTAINED WHAT IS
4740                                                                    ;GIVEN IN BAD RHCS3
4741  015000                                79$:
4742                                              ;CHECK THAT RHBA HAS 0
4743  015000  012737  000000  001124              MOV     #0,$GDDAT       ;GET GOOD = 0
4744
4745  015006  017737  167052  001126              MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
4746  015014  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4747                                                                    ;DATA WITH DATA READ FROM
4748                                                                    ;RHBA
4749  015022  001401                            BEQ     81$             ;BRANCH IF GOOD
4750  015024  104143                            ERROR   143
4751                                                                    ;AFTER CLEARING RH THEN
4752                                                                    ;WRITING TWO WORDS INTO
4753                                                                    ;RHDB 52525 AND 125252
4754                                                                    ;THEN READING RHDB THE
4755                                                                    ;SECOND TIME
4756                                                                    ;RHBA SHOULD HAVE 0
4757                                                                    ;BUT CONTAINED WHAT IS
4758                                                                    ;GIVEN IN BAD RHBA
4759  015026                                81$:
4760                                              ;CHECK THAT RHBAE HAS 0
4761  015026  012737  000000  001124              MOV     #0,$GDDAT       ;GET GOOD = 0
4762
4763  015034  017737  167070  001126              MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
4764  015042  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4765                                                                    ;DATA WITH DATA READ FROM
4766                                                                    ;RHBAE
4767  015050  001401                            BEQ     83$             ;BRANCH IF GOOD
4768  015052  104144                            ERROR   144
4769                                                                    ;AFTER CLEARING RH THEN
4770                                                                    ;WRITING TWO WORDS INTO
4771                                                                    ;RHDB 52525 AND 125252
4772                                                                    ;THEN READING RHDB THE
4773                                                                    ;SECOND TIME
4774                                                                    ;RHBAE SHOULD HAVE 0
4775                                                                    ;BUT CONTAINED WHAT IS
```

```
                                                        ;GIVEN IN BAD RHBAE
015054                          83$:
                                        ;CHECK THAT RHWC HAS 0
015054  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

015062  017737  166774  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
015070  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
015076  001401                          BEQ     85$             ;BRANCH IF GOOD
015100  104145                          ERROR   145

                                                                ;AFTER CLEARING RH THEN
                                                                ;WRITING TWO WORDS INTO
                                                                ;RHDB 52525 AND 125252
                                                                ;THEN READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
015102                          85$:


                        ;;**************************************************************
                        ;*TEST 14        SILO TEST 4 (COUNT PATTERN)

                        ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                        ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH, TOGETHER
                        ;*      WITH UNIT NUMBER
                        ;*      EIGHT WORDS, A COUNT PATTERN 0 THRU 7 IS WRITTEN
                        ;*      INTO RHDB
                        ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                        ;*      INSTRUCTION CALLED "WAT"
                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                        ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                        ;*      THEN RHBIS READ AND COMPARED TO HAVE THE RIGHT VALUE
                        ;*      EIGHT TIES
                        ;*      THEN RHCS2, RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                        ;*      CHECKED TO HAVE THE APPROPIATE VALUE
                        ;;**************************************************************
015102  000004          TST14:  SCOPE
015104  012706  001000          MOV     #STACK,SP       ;RESET STACK
015110  012737  000014  004656  MOV     #14,@#TSTNM     ;SAVE TEST NUMBER

015116  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS2 HAS IR
015122  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
015130  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

015136  017737  166726  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
015144  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
015152  001401                          BEQ     65$             ;BRANCH IF GOOD
```

```
4832  015154  104006                          ERROR   6
4833                                                  ;AFTER SETTING "CLR" BIT #5
4834                                                  ;IN RHCS2 TO INIT THE RH
4835                                                  ;AND HAVING DONE NOTHING ELSE
4836                                                  ;RHCS2 SHOULD HAVE IR
4837                                                  ;=100
4838                                                  ;TOGETHER WITH UNIT NUMBER
4839                                                  ;BUT CONTAINED WHAT IS
4840                                                  ;GIVEN IN BAD RHCS2
4841  015156                          65$:
4842
4843                                          ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
4844                                          ;THIS IS A COUNT PATTERN 0 THRU 7
4845  015156  012702  000010                 MOV     #8.,R2          ;8 NUMBERS TO BE WRITTEN
4846  015162  005001                          CLR     R1              ;FIRST WORD TO BE WRITTEN = 0
4847  015164  010177  166712          1$:     MOV     R1,@RHDB        ;WRITE INTO RHDB - A COUNT
4848  015170  005201                          INC     R1              ;GET NEXT WORD
4849  015172  005302                          DEC     R2              ;COUNT TO 8
4850  015174  001373                          BNE     1$              ;BRANCH IF 8 NOT DONE
4851
4852                                          ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
4853  015176  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
4854  015200  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
4855  015202  000200                          OR                      ;RHCS2 REGISTER
4856                                                  ;IF ERROR OCCURS HERE
4857                                                  ;IT MEANS "OR" DID NOT
4858                                                  ;SET FOR THE FULL COUNT
4859                                                  ;DOWN OF THE WAIT.T SUBROUTINE
4860                                                  ;THIS TIME IS APPROXIMATELY
4861                                                  ;LARGER THAN 500 MILLISECONDS
4862
4863                                          ;CHECK THAT RHCS2 HAS OR
4864  015204  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
4865  015212  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4866
4867  015220  017737  166644  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4868  015226  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4869                                                  ;DATA WITH DATA READ FROM
4870                                                  ;RHCS2
4871  015234  001401                          BEQ     67$             ;BRANCH IF GOOD
4872  015236  104023                          ERROR   23
4873                                                  ;AFTER SETTING "CLR" BIT #5
4874                                                  ;IN RHCS2 TO INIT THE RH
4875                                                  ;AND HAVING WRITTEN A
4876                                                  ;COUNT PATTERN 0 THRU 7
4877                                                  ;INTO RHDB TO FILL IT
4878                                                  ;RHCS2 SHOULD HAVE OR
4879                                                  ;=200
4880                                                  ;TOGETHER WITH UNIT NUMBER
4881                                                  ;BUT CONTAINED WHAT IS
4882                                                  ;GIVEN IN BAD RHCS2
4883  015240                          67$:
4884
4885
4886
4887  015240  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
```

```
                                        ;CHECK THAT RHDB HAS 0
015246  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

015254  017737  166622  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015262  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
015270  001401                          BEQ     69$             ;BRANCH IF GOOD
015272  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;FIRST TIME
                                                                ;RHDB SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015274                          69$:

015274  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 1
015302  012737  000001  001124          MOV     #1,$GDDAT       ;GET GOOD = 0

015310  017737  166566  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015316  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
015324  001401                          BEQ     71$             ;BRANCH IF GOOD
015326  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHDB SHOULD HAVE 1
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015330                          71$:

015330  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 2
015336  012737  000002  001124          MOV     #2,$GDDAT       ;GET GOOD = 0

015344  017737  166532  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015352  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
015360  001401                          BEQ     73$             ;BRANCH IF GOOD
015362  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
```

```
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;THIRD TIME
                                                                ;RHDB SHOULD HAVE 2
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015364                                  73$:
015364  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 3
015372  012737  000003  001124          MOV     #3,$GDDAT       ;GET GOOD = 0
015400  017737  166476  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015406  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
015414  001401                          BEQ     75$             ;BRANCH IF GOOD
015416  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;FOURTH TIME
                                                                ;RHDB SHOULD HAVE 3
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015420                                  75$:
015420  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 4
015426  012737  000004  001124          MOV     #4,$GDDAT       ;GET GOOD = 0
015434  017737  166442  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015442  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
015450  001401                          BEQ     77$             ;BRANCH IF GOOD
015452  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;FIFTH TIME
                                                                ;RHDB SHOULD HAVE 4
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015454                                  77$:
015454  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 5
015462  012737  000005  001124          MOV     #5,$GDDAT       ;GET GOOD = 0
```

```
5000
5001  015470  017737  166406  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5002  015476  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5003                                                                  ;DATA WITH DATA READ FROM
5004                                                                  ;RHDB
5005  015504  001401                          BEQ     79$             ;BRANCH IF GOOD
5006  015506  104024                          ERROR   24
5007                                                                  ;AFTER SETTING "CLR" BIT #5
5008                                                                  ;IN RHCS2 TO INIT THE RH
5009                                                                  ;AND WRITING A
5010                                                                  ;COUNT PATTERN 0 THRU 7
5011                                                                  ;INTO RHDB TO FILL IT
5012                                                                  ;THEN ON READING RHDB THE
5013                                                                  ;SIXTH TIME
5014                                                                  ;RHDB SHOULD HAVE 5
5015                                                                  ;BUT CONTAINED WHAT IS
5016                                                                  ;GIVEN IN BAD RHDB
5017  015510                          79$:
5018
5019  015510  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
5020                                          ;CHECK THAT RHDB HAS 6
5021  015516  012737  000006  001124          MOV     #6,$GDDAT       ;GET GOOD = 0
5022
5023  015524  017737  166352  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5024  015532  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5025                                                                  ;DATA WITH DATA READ FROM
5026                                                                  ;RHDB
5027  015540  001401                          BEQ     81$             ;BRANCH IF GOOD
5028  015542  104024                          ERROR   24
5029                                                                  ;AFTER SETTING "CLR" BIT #5
5030                                                                  ;IN RHCS2 TO INIT THE RH
5031                                                                  ;AND WRITING A
5032                                                                  ;COUNT PATTERN 0 THRU 7
5033                                                                  ;INTO RHDB TO FILL IT
5034                                                                  ;THEN ON READING RHDB THE
5035                                                                  ;SEVENTH TIME
5036                                                                  ;RHDB SHOULD HAVE 6
5037                                                                  ;BUT CONTAINED WHAT IS
5038                                                                  ;GIVEN IN BAD RHDB
5039  015544                          81$:
5040
5041  015544  012737  000010  004664          MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
5042                                          ;CHECK THAT RHDB HAS 7
5043  015552  012737  000007  001124          MOV     #7,$GDDAT       ;GET GOOD = 0
5044
5045  015560  017737  166316  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5046  015566  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5047                                                                  ;DATA WITH DATA READ FROM
5048                                                                  ;RHDB
5049  015574  001401                          BEQ     83$             ;BRANCH IF GOOD
5050  015576  104024                          ERROR   24
5051                                                                  ;AFTER SETTING "CLR" BIT #5
5052                                                                  ;IN RHCS2 TO INIT THE RH
5053                                                                  ;AND WRITING A
5054                                                                  ;COUNT PATTERN 0 THRU 7
5055                                                                  ;INTO RHDB TO FILL IT
```

```
                                                                ;THEN ON READING RHDB THE
                                                                ;EIGHTH TIME
                                                                ;RHDB SHOULD HAVE 7
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015600                                  83$:
                                                ;CHECK THAT RHCS2 HAS IR
015600  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
015606  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

015614  017737  166250  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
015622  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
015630  001401                          BEQ     85$             ;BRANCH IF GOOD
015632  104025                          ERROR   25

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
015634                                  85$:
                                                ;CHECK THAT RHCS1 HAS DVA!RDY
015634  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

015642  017737  166212  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
015650  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
015656  001401                          BEQ     87$             ;BRANCH IF GOOD
015660  104026                          ERROR   26

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHCS1 SHOULD HAVE DVA!RDY
                                                                ;=4200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
015662                                  87$:
                                                ;CHECK THAT RHCS3 HAS 0
015662  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

015670  017737  166236  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
```

```
5112  015676  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5113                                                                  ;DATA WITH DATA READ FROM
5114                                                                  ;RHCS3
5115  015704  001401                          BEQ     89$             ;BRANCH IF GOOD
5116  015706  104027                          ERROR   27
5117                                                                  ;AFTER SETTING "CLR" BIT #5
5118                                                                  ;IN RHCS2 TO INIT THE RH
5119                                                                  ;AND WRITING A
5120                                                                  ;COUNT PATTERN 0 THRU 7
5121                                                                  ;INTO RHDB TO FILL IT
5122                                                                  ;THEN ON READING RHDB THE
5123                                                                  ;TOTAL 8 TIMES
5124                                                                  ;RHCS3 SHOULD HAVE 0
5125                                                                  ;BUT CONTAINED WHAT IS
5126                                                                  ;GIVEN IN BAD RHCS3
5127  015710                          89$:
5128                                  ;CHECK THAT RHBA HAS 0
5129  015710  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5130
5131  015716  017737  166142  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
5132  015724  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5133                                                                  ;DATA WITH DATA READ FROM
5134                                                                  ;RHBA
5135  015732  001401                          BEQ     91$             ;BRANCH IF GOOD
5136  015734  104030                          ERROR   30
5137                                                                  ;AFTER SETTING "CLR" BIT #5
5138                                                                  ;IN RHCS2 TO INIT THE RH
5139                                                                  ;AND WRITING A
5140                                                                  ;COUNT PATTERN 0 THRU 7
5141                                                                  ;INTO RHDB TO FILL IT
5142                                                                  ;THEN ON READING RHDB THE
5143                                                                  ;TOTAL 8 TIMES
5144                                                                  ;RHBA SHOULD HAVE 0
5145                                                                  ;BUT CONTAINED WHAT IS
5146                                                                  ;GIVEN IN BAD RHBA
5147  015736                          91$:
5148                                  ;CHECK THAT RHBAE HAS 0
5149  015736  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5150
5151  015744  017737  166160  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
5152  015752  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5153                                                                  ;DATA WITH DATA READ FROM
5154                                                                  ;RHBAE
5155  015760  001401                          BEQ     93$             ;BRANCH IF GOOD
5156  015762  104031                          ERROR   31
5157                                                                  ;AFTER SETTING "CLR" BIT #5
5158                                                                  ;IN RHCS2 TO INIT THE RH
5159                                                                  ;AND WRITING A
5160                                                                  ;COUNT PATTERN 0 THRU 7
5161                                                                  ;INTO RHDB TO FILL IT
5162                                                                  ;THEN ON READING RHDB THE
5163                                                                  ;TOTAL 8 TIMES
5164                                                                  ;RHBAE SHOULD HAVE 0
5165                                                                  ;BUT CONTAINED WHAT IS
5166                                                                  ;GIVEN IN BAD RHBAE
5167  015764                          93$:
```

```
                                        ;CHECK THAT RHWC HAS 0
015764  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

015772  017737  166064  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
016000  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
016006  001401                          BEQ     95$             ;BRANCH IF GOOD
016010  104032                          ERROR   32

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A
                                                                ;COUNT PATTERN 0 THRU 7
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
016012                          95$:


                        ;;**************************************************************
                        ;*TEST 15        SILO TEST 5 (FLOATING ONES)

                        ;*      AFTER ARH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                        ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                        ;*      TOGETHER WITH UNIT NUMBER
                        ;*      EIGHT WORDS, A PATTERN OF FLOATING ONES (1,2,4,10,20,40,100,200)
                        ;*      IS WRITTEN INTO RHDB
                        ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                        ;*      INSTRUCTION CALLED "WAT"
                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                        ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                        ;*      THEN RHDB IS READ AND COMPARED TO HAVE THE
                        ;*      RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR"
                        ;*      AND UNIT NUMBER
                        ;*      THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                        ;*      CHECKED TO HAVE THE APPROPIATE VALUE
                        ;;**************************************************************
016012  000004          TST15:  SCOPE
016014  012706  001000          MOV     #STACK,SP       ;RESET STACK
016020  012737  000015  004656  MOV     #15,@#TSTNM     ;SAVE TEST NUMBER

016026  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS2 HAS IR
016032  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
016040  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

016046  017737  166016  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
016054  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
016062  001401                        BEQ     65$               ;BRANCH IF GOOD
016064  104006                        ERROR   6
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND HAVING DONE NOTHING
                                                                ;ELSE
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
016066                          65$:
                                ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
                                ;DATA IS 1,2,4,10,20,40,100,200
016066  012701  000001          MOV     #1,R1                   ;GETTING READY TO FLOAT ONES
016072  012702  000010          MOV     #8.,R2                  ;COUNTER -8 DATA WORDS
016076  010177  166000  1$:     MOV     R1,@RHDB                ;WRITE INTO RHDB
016102  006301                  ASL     R1                      ;SHIFT 1 ONE POSITION LEFT
016104  005302                  DEC     R2                      ;COUNT TO 8
016106  001373                  BNE     1$                      ;BRANCH IF 8 NOT DONE
                                ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
016110  104411                  WAT                             ;TRAP TO WAIT.T SUBROUTINE
016112  004070                  RHCS2                           ;AND WAIT FOR OR BIT IN
016114  000200                  OR                              ;RHCS2 REGISTER
                                                                ;IF ERROR OCCURS HERE
                                                                ;IT MEANS "OR" DID NOT
                                                                ;SET FOR THE FULL COUNT
                                                                ;DOWN OF THE WAIT.T SUBROUTINE
                                                                ;THIS TIME IS APPROXIMATELY
                                                                ;LARGER THAN 500 MILLISECONDS
                                ;CHECK THAT RHCS2 HAS OR
016116  012737  000200  001124  MOV     #OR,$GDDAT              ;GET GOOD = 200
016124  053737  005010  001124  BIS     UNIT,$GDDAT             ;INCLUDE UNIT NUMBER
016132  017737  165732  001126  MOV     @RHCS2,$BDDAT           ;READ RHCS2 FOR COMPARISON
016140  023737  001124  001126  CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
016146  001401                  BEQ     67$                     ;BRANCH IF GOOD
016150  104023                  ERROR   23
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND HAVING WRITTEN A
                                                                ;FLOATING ONES PATTERN
                                                                ;1,2,4,10,20,40,100,200
                                                                ;INTO RHDB TO FILL IT
                                                                ;RHCS2 SHOULD HAVE OR
                                                                ;=200
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
```

```
5280  016152                          67$:
5281
5282
5283
5284  016152  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
5285                                          ;CHECK THAT RHDB HAS 1
5286  016160  012737  000001  001124          MOV     #1,$GDDAT       ;GET GOOD = 0
5287
5288  016166  017737  165710  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5289  016174  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5290                                                                  ;DATA WITH DATA READ FROM
5291                                                                  ;RHDB
5292  016202  001401                          BEQ     69$             ;BRANCH IF GOOD
5293  016204  104024                          ERROR   24
5294                                                                  ;AFTER SETTING "CLR" BIT #5
5295                                                                  ;IN RHCS2 TO INIT THE RH
5296                                                                  ;AND WRITING A PATTERN OF
5297                                                                  ;FLOATING ONES (WHICH IS
5298                                                                  ;1,2,4,10,20,40,100,200)
5299                                                                  ;INTO RHDB TO FILL IT
5300                                                                  ;THEN ON READING RHDB THE
5301                                                                  ;FIRST TIME
5302                                                                  ;RHDB SHOULD HAVE 1
5303                                                                  ;BUT CONTAINED WHAT IS
5304                                                                  ;GIVEN IN BAD RHDB
5305  016206                          69$:
5306
5307
5308  016206  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
5309                                          ;CHECK THAT RHDB HAS 2
5310  016214  012737  000002  001124          MOV     #2,$GDDAT       ;GET GOOD = 0
5311
5312  016222  017737  165654  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5313  016230  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5314                                                                  ;DATA WITH DATA READ FROM
5315                                                                  ;RHDB
5316  016236  001401                          BEQ     71$             ;BRANCH IF GOOD
5317  016240  104024                          ERROR   24
5318                                                                  ;AFTER SETTING "CLR" BIT #5
5319                                                                  ;IN RHCS2 TO INIT THE RH
5320                                                                  ;AND WRITING A PATTERN OF
5321                                                                  ;FLOATING ONES (WHICH IS
5322                                                                  ;1,2,4,10,20,40,100,200)
5323                                                                  ;INTO RHDB TO FILL IT
5324                                                                  ;THEN ON READING RHDB THE
5325                                                                  ;SECOND TIME
5326                                                                  ;RHDB SHOULD HAVE 2
5327                                                                  ;BUT CONTAINED WHAT IS
5328                                                                  ;GIVEN IN BAD RHDB
5329  016242                          71$:
5330
5331
5332  016242  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
5333                                          ;CHECK THAT RHDB HAS 4
5334  016250  012737  000004  001124          MOV     #4,$GDDAT       ;GET GOOD = 0
5335
```

```
5336  016256  017737  165620  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5337  016264  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5338                                                                  ;DATA WITH DATA READ FROM
5339                                                                  ;RHDB
5340  016272  001401                          BEQ     73$             ;BRANCH IF GOOD
5341  016274  104024                          ERROR   24
5342                                                                  ;AFTER SETTING "CLR" BIT #5
5343                                                                  ;IN RHCS2 TO INIT THE RH
5344                                                                  ;AND WRITING A PATTERN OF
5345                                                                  ;FLOATING ONES (WHICH IS
5346                                                                  ;1,2,4,10,20,40,100,200)
5347                                                                  ;INTO RHDB TO FILL IT
5348                                                                  ;THEN ON READING RHDB THE
5349                                                                  ;THIRD TIME
5350                                                                  ;RHDB SHOULD HAVE 4
5351                                                                  ;BUT CONTAINED WHAT IS
5352                                                                  ;GIVEN IN BAD RHDB
5353  016276                          73$:
5354
5355
5356  016276  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
5357                                          ;CHECK THAT RHDB HAS 10
5358  016304  012737  000010  001124          MOV     #10,$GDDAT      ;GET GOOD = 0
5359
5360  016312  017737  165564  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5361  016320  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5362                                                                  ;DATA WITH DATA READ FROM
5363                                                                  ;RHDB
5364  016326  001401                          BEQ     75$             ;BRANCH IF GOOD
5365  016330  104024                          ERROR   24
5366                                                                  ;AFTER SETTING "CLR" BIT #5
5367                                                                  ;IN RHCS2 TO INIT THE RH
5368                                                                  ;AND WRITING A PATTERN OF
5369                                                                  ;FLOATING ONES (WHICH IS
5370                                                                  ;1,2,4,10,20,40,100,200)
5371                                                                  ;INTO RHDB TO FILL IT
5372                                                                  ;THEN ON READING RHDB THE
5373                                                                  ;FOURTH TIME
5374                                                                  ;RHDB SHOULD HAVE 10
5375                                                                  ;BUT CONTAINED WHAT IS
5376                                                                  ;GIVEN IN BAD RHDB
5377  016332                          75$:
5378
5379
5380  016332  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
5381                                          ;CHECK THAT RHDB HAS 20
5382  016340  012737  000020  001124          MOV     #20,$GDDAT      ;GET GOOD = 0
5383
5384  016346  017737  165530  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5385  016354  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5386                                                                  ;DATA WITH DATA READ FROM
5387                                                                  ;RHDB
5388  016362  001401                          BEQ     77$             ;BRANCH IF GOOD
5389  016364  104024                          ERROR   24
5390                                                                  ;AFTER SETTING "CLR" BIT #5
5391                                                                  ;IN RHCS2 TO INIT THE RH
```

```
                                                        ;CHK WRITING A PATTERN OF
                                                        ;FLOATING ONES (WHICH IS
                                                        ;1,2,4,10,20,40,100,200)
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;FIFTH TIME
                                                        ;RHDB SHOULD HAVE 20
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
016366                          77$:

016366  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 40
016374  012737  000040  001124          MOV     #40,$GDDAT      ;GET GOOD = 0

016402  017737  165474  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
016410  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
016416  001401                          BEQ     79$             ;BRANCH IF GOOD
016420  104024                          ERROR   24

                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF
                                                        ;FLOATING ONES (WHICH IS
                                                        ;1,2,4,10,20,40,100,200)
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;SIXTH TIME
                                                        ;RHDB SHOULD HAVE 40
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
016422                          79$:

016422  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 100
016430  012737  000100  001124          MOV     #100,$GDDAT     ;GET GOOD = 0

016436  017737  165440  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
016444  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
016452  001401                          BEQ     81$             ;BRANCH IF GOOD
016454  104024                          ERROR   24

                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF
                                                        ;FLOATING ONES (WHICH IS
                                                        ;1,2,4,10,20,40,100,200)
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;SEVENTH TIME
                                                        ;RHDB SHOULD HAVE 100
                                                        ;BUT CONTAINED WHAT IS
```

```
                                                        ;GIVEN IN BAD RHDB
016456                          81$:
                                
                                
016456  012737  000010  004664          MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 200
016464  012737  000200  001124          MOV     #200,$GDDAT     ;GET GOOD = 0
                                
016472  017737  165404  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
016500  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
016506  001401                          BEQ     83$             ;BRANCH IF GOOD
016510  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF
                                                                ;FLOATING ONES (WHICH IS
                                                                ;1,2,4,10,20,40,100,200)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;EIGHTH TIME
                                                                ;RHDB SHOULD HAVE 200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
016512                          83$:
                                
                                
                                        ;CHECK THAT RHCS2 HAS IR
016512  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
016520  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
                                
016526  017737  165336  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
016534  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
016542  001401                          BEQ     85$             ;BRANCH IF GOOD
016544  104025                          ERROR   25
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF
                                                                ;FLOATING ONES (WHICH IS
                                                                ;1,2,4,10,20,40,100,200)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
016546                          85$:
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
016546  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
                                
016554  017737  165300  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
```

```
005504  016562  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
005505                                                                  ;DATA WITH DATA READ FROM
005506                                                                  ;RHCS1
005507  016570  001401                          BEQ     87$             ;BRANCH IF GOOD
005508  016572  104026                          ERROR   26
005509                                                                  ;AFTER SETTING "CLR" BIT #5
005510                                                                  ;IN RHCS2 TO INIT THE RH
005511                                                                  ;AND WRITING A PATTERN OF
005512                                                                  ;FLOATING ONES (WHICH IS
005513                                                                  ;1,2,4,10,20,40,100,200)
005514                                                                  ;INTO RHDB TO FILL IT
005515                                                                  ;THEN ON READING RHDB THE
005516                                                                  ;TOTAL 8 TIMES
005517                                                                  ;RHCS1 SHOULD HAVE DVA!RDY
005518                                                                    4200
005519                                                                    UT CONTAINED WHAT IS
005520                                                                  ;GIVEN IN BAD RHCS1
005521  016574                          87$:
005522                                          ;CHECK THAT RHCS3 HAS 0
005523  016574  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
005524
005525  016602  017737  165324  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
005526  016610  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
005527                                                                  ;DATA WITH DATA READ FROM
005528                                                                  ;RHCS3
005529  016616  001401                          BEQ     89$             ;BRANCH IF GOOD
005530  016620  104027                          ERROR   27
005531                                                                  ;AFTER SETTING "CLR" BIT #5
005532                                                                  ;IN RHCS2 TO INIT THE RH
005533                                                                  ;AND WRITING A PATTERN OF
005534                                                                  ;FLOATING ONES (WHICH IS
005535                                                                  ;1,2,4,10,20,40,100,200)
005536                                                                  ;INTO RHDB TO FILL IT
005537                                                                  ;THEN ON READING RHDB THE
005538                                                                  ;TOTAL 8 TIMES
005539                                                                  ;RHCS3 SHOULD HAVE 0
005540                                                                  ;BUT CONTAINED WHAT IS
005541                                                                  ;GIVEN IN BAD RHCS3
005542  016622                          89$:
005543                                          ;CHECK THAT RHBA HAS 0
005544  016622  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
005545
005546  016630  017737  165230  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
005547  016636  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
005548                                                                  ;DATA WITH DATA READ FROM
005549                                                                  ;RHBA
005550  016644  001401                          BEQ     91$             ;BRANCH IF GOOD
005551  016646  104030                          ERROR   30
005552                                                                  ;AFTER SETTING "CLR" BIT #5
005553                                                                  ;IN RHCS2 TO INIT THE RH
005554                                                                  ;AND WRITING A PATTERN OF
005555                                                                  ;FLOATING ONES (WHICH IS
005556                                                                  ;1,2,4,10,20,40,100,200)
005557                                                                  ;INTO RHDB TO FILL IT
005558                                                                  ;THEN ON READING RHDB THE
005559                                                                  ;TOTAL 8 TIMES
```

```
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
016650                                  91$:
                                                ;CHECK THAT RHBAE HAS 0
016650  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

016656  017737  165246  001126                  MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
016664  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
016672  001401                                  BEQ     93$             ;BRANCH IF GOOD
016674  104031                                  ERROR   31

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF
                                                                ;FLOATING ONES (WHICH IS
                                                                ;1,2,4,10,20,40,100,200)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
016676                                  93$:
                                                ;CHECK THAT RHWC HAS 0
016676  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

016704  017737  165152  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
016712  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
016720  001401                                  BEQ     95$             ;BRANCH IF GOOD
016722  104032                                  ERROR   32

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF
                                                                ;FLOATING ONES (WHICH IS
                                                                ;1,2,4,10,20,40,100,200)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
016724                                  95$:


                        ;;**************************************************************
                        ;*TEST 16        SILO TEST 6 (FLOATING ONES IN UPPER BYTE)

                        ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                        ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                        ;*      TOGETHER WITH UNIT NUMBER
                        ;*      EIGHT WORDS, A PATTERN OF FLOATING ONES IN UPPER BYTE
```

```
                                        ;*      400,1000,2000,4000,10000,20000,40000,100000
                                        ;*      IS WRITTEN INTO RHDB
                                        ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                        ;*      INSTRUCTION CALLED "WAT"
                                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                        ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                        ;*      THEN RHDB IS READ AND COMPARED TO HAVE THE
                                        ;*      RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR"
                                        ;*      AND UNIT NUMBER
                                        ;*      THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                        ;*      CHECKED TO HAVE THE APPROPIATE VALUE
                                        ;**************************************************************
016724  000004                    TST16:  SCOPE
016726  012706  001000                    MOV     #STACK,SP       ;RESET STACK
016732  012737  000016  004656            MOV     #16,@#TSTNM     ;SAVE TEST NUMBER
016740  004737  040312                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                  ;SETUP UNIT NUBER
                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                        ;CHECK THAT RHCS2 HAS IR
016744  012737  000100  001124            MOV     #IR,$GDDAT      ;GET GOOD = 100
016752  053737  005010  001124            BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
016760  017737  165104  001126            MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
016766  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                  ;DATA WITH DATA READ FROM
                                                                  ;RHCS2
016774  001401                            BEQ     65$             ;BRANCH IF GOOD
016776  104006                            ERROR   6
                                                                  ;AFTER SETTING "CLR" BIT #5
                                                                  ;IN RHCS2 TO INIT THE RH
                                                                  ;AND HAVING DONE NOTHING
                                                                  ;ELSE
                                                                  ;RHCS2 SHOULD HAVE IR
                                                                  ;=100
                                                                  ;TOGETHER WITH UNIT NUMBER
                                                                  ;BUT CONTAINED WHAT IS
                                                                  ;GIVEN IN BAD RHCS2
017000                          65$:
                                        ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
                                        ;DATA IS 400,1000,2000,4000,10000,20000,40000
                                        ;100000
017000  012701  000400                    MOV     #400,R1         ;GETTING READY TO FLOAT ONES
                                                                  ;IN UPPER BYTE
017004  012702  000010                    MOV     #8.,R2          ;COUNTER -8 DATA WORDS
017010  010177  165066          1$:       MOV     R1,@RHDB        ;WRITE INTO RHDB
017014  006301                            ASL     R1              ;SHIFT 1 ONE POSITION LEFT
017016  005302                            DEC     R2              ;COUNT TO 8
017020  001373                            BNE     1$              ;BRANCH IF 8 NOT DONE
                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
017022  104411                            WAT                     ;TRAP TO WAIT.T SUBROUTINE
017024  004070                            RHCS2                   ;AND WAIT FOR OR BIT IN
```

```
5672  017026  000200                          OR                      ;RHCS2 REGISTER
5673                                                                  ;IF ERROR OCCURS HERE
5674                                                                  ;IT MEANS "OR" DID NOT
5675                                                                  ;SET FOR THE FULL COUNT
5676                                                                  ;DOWN OF THE .WAIT.T SUBROUTINE
5677                                                                  ;THIS TIME IS APPROXIMATELY
5678                                                                  ;LARGER THAN 500 MILLISECONDS
5679
5680
5681                                          ;CHECK THAT RHCS2 HAS OR
5682  017030  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
5683  017036  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
5684
5685  017044  017737  165020  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
5686  017052  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5687                                                                  ;DATA WITH DATA READ FROM
5688                                                                  ;RHCS2
5689  017060  001401                          BEQ     67$             ;BRANCH IF GOOD
5690  017062  104023                          ERROR   23
5691                                                                  ;AFTER SETTING "CLR" BIT #5
5692                                                                  ;IN RHCS2 TO INIT THE RH
5693                                                                  ;AND HAVING WRITTEN A
5694                                                                  ;FLOATING ONES PATTERN INTO UPPER BYTE
5695                                                                  ;400,1000,2000,4000,10000,20000
5696                                                                  ;40000,100000
5697                                                                  ;INTO RHDB TO FILL IT
5698                                                                  ;RHCS2 SHOULD HAVE OR
5699                                                                  ;=200
5700                                                                  ;TOGETHER WITH UNIT NUMBER
5701                                                                  ;BUT CONTAINED WHAT IS
5702                                                                  ;GIVEN IN BAD RHCS2
5703  017064                          67$:
5704
5705
5706
5707  017064  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
5708                                          ;CHECK THAT RHDB HAS 400
5709  017072  012737  000400  001124          MOV     #400,$GDDAT     ;GET GOOD = 0
5710
5711  017100  017737  164776  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5712  017106  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5713                                                                  ;DATA WITH DATA READ FROM
5714                                                                  ;RHDB
5715  017114  001401                          BEQ     69$             ;BRANCH IF GOOD
5716  017116  104024                          ERROR   24
5717                                                                  ;AFTER SETTING "CLR" BIT #5
5718                                                                  ;IN RHCS2 TO INIT THE RH
5719                                                                  ;AND WRITING A PATTERN OF FLOATING
5720                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5721                                                                  ;10000,20000,40000,100000)
5722                                                                  ;INTO RHDB TO FILL IT
5723                                                                  ;THEN ON READING RHDB THE
5724                                                                  ;FIRST TIME
5725                                                                  ;RHDB SHOULD HAVE 400
5726                                                                  ;BUT CONTAINED WHAT IS
5727                                                                  ;GIVEN IN BAD RHDB
```

```
5728  017120                          69$:
5729
5730
5731  017120  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
5732                                          ;CHECK THAT RHDB HAS 1000
5733  017126  012737  001000  001124          MOV     #1000,$GDDAT    ;GET GOOD = 0
5734
5735  017134  017737  164742  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5736  017142  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5737                                                                  ;DATA WITH DATA READ FROM
5738                                                                  ;RHDB
5739  017150  001401                          BEQ     71$             ;BRANCH IF GOOD
5740  017152  104024                          ERROR   24
5741                                                                  ;AFTER SETTING "CLR" BIT #5
5742                                                                  ;IN RHCS2 TO INIT THE RH
5743                                                                  ;AND WRITING A PATTERN OF FLOATING
5744                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5745                                                                  ;10000,20000,40000,100000)
5746                                                                  ;INTO RHDB TO FILL IT
5747                                                                  ;THEN ON READING RHDB THE
5748                                                                  ;SECOND TIME
5749                                                                  ;RHDB SHOULD HAVE 1000
5750                                                                  ;BUT CONTAINED WHAT IS
5751                                                                  ;GIVEN IN BAD RHDB
5752  017154                          71$:
5753
5754
5755  017154  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
5756                                          ;CHECK THAT RHDB HAS 2000
5757  017162  012737  002000  001124          MOV     #2000,$GDDAT    ;GET GOOD = 0
5758
5759  017170  017737  164706  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5760  017176  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5761                                                                  ;DATA WITH DATA READ FROM
5762                                                                  ;RHDB
5763  017204  001401                          BEQ     73$             ;BRANCH IF GOOD
5764  017206  104024                          ERROR   24
5765                                                                  ;AFTER SETTING "CLR" BIT #5
5766                                                                  ;IN RHCS2 TO INIT THE RH
5767                                                                  ;AND WRITING A PATTERN OF FLOATING
5768                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5769                                                                  ;10000,20000,40000,100000)
5770                                                                  ;INTO RHDB TO FILL IT
5771                                                                  ;THEN ON READING RHDB THE
5772                                                                  ;THIRD TIME
5773                                                                  ;RHDB SHOULD HAVE 2000
5774                                                                  ;BUT CONTAINED WHAT IS
5775                                                                  ;GIVEN IN BAD RHDB
5776  017210                          73$:
5777
5778
5779  017210  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
5780                                          ;CHECK THAT RHDB HAS 4000
5781  017216  012737  004000  001124          MOV     #4000,$GDDAT    ;GET GOOD = 0
5782
5783  017224  017737  164652  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
```

```
5784  017232  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5785                                                                    ;DATA WITH DATA READ FROM
5786                                                                    ;RHDB
5787  017240  001401                          BEQ     75$             ;BRANCH IF GOOD
5788  017242  104024                          ERROR   24
5789                                                                    ;AFTER SETTING "CLR" BIT #5
5790                                                                    ;IN RHCS2 TO INIT THE RH
5791                                                                    ;AND WRITING A PATTERN OF FLOATING
5792                                                                    ;ONES (WHICH IS 400,1000,2000,4000,
5793                                                                    ;10000,20000,40000,100000)
5794                                                                    ;INTO RHDB TO FILL IT
5795                                                                    ;THEN ON READING RHDB THE
5796                                                                    ;FOURTH TIME
5797                                                                    ;RHDB SHOULD HAVE 4000
5798                                                                    ;BUT CONTAINED WHAT IS
5799                                                                    ;GIVEN IN BAD RHDB
5800  017244                          75$:
5801
5802
5803  017244  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
5804                                          ;CHECK THAT RHDB HAS 10000
5805  017252  012737  010000  001124          MOV     #10000,$GDDAT   ;GET GOOD = 0
5806
5807  017260  017737  164616  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5808  017266  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5809                                                                    ;DATA WITH DATA READ FROM
5810                                                                    ;RHDB
5811  017274  001401                          BEQ     77$             ;BRANCH IF GOOD
5812  017276  104024                          ERROR   24
5813                                                                    ;AFTER SETTING "CLR" BIT #5
5814                                                                    ;IN RHCS2 TO INIT THE RH
5815                                                                    ;AND WRITING A PATTERN OF FLOATING
5816                                                                    ;ONES (WHICH IS 400,1000,2000,4000,
5817                                                                    ;10000,20000,40000,100000)
5818                                                                    ;INTO RHDB TO FILL IT
5819                                                                    ;THEN ON READING RHDB THE
5820                                                                    ;FIFTH TIME
5821                                                                    ;RHDB SHOULD HAVE 10000
5822                                                                    ;BUT CONTAINED WHAT IS
5823                                                                    ;GIVEN IN BAD RHDB
5824  017300                          77$:
5825
5826
5827  017300  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
5828                                          ;CHECK THAT RHDB HAS 20000
5829  017306  012737  020000  001124          MOV     #20000,$GDDAT   ;GET GOOD = 0
5830
5831  017314  017737  164562  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5832  017322  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5833                                                                    ;DATA WITH DATA READ FROM
5834                                                                    ;RHDB
5835  017330  001401                          BEQ     79$             ;BRANCH IF GOOD
5836  017332  104024                          ERROR   24
5837                                                                    ;AFTER SETTING "CLR" BIT #5
5838                                                                    ;IN RHCS2 TO INIT THE RH
5839                                                                    ;AND WRITING A PATTERN OF FLOATING
```

```
                                                        ;ONES (WHICH IS 400,1000,2000,4000,
                                                        ;10000,20000,40000,100000)
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;SIXTH TIME
                                                        ;RHDB SHOULD HAVE 20000
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
017334                          79$:

017334  012737  000007  004664        MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
                                    ;CHECK THAT RHDB HAS 40000
017342  012737  040000  001124        MOV     #40000,$GDDAT   ;GET GOOD = 0

017350  017737  164526  001126        MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
017356  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
017364  001401                      BEQ     81$             ;BRANCH IF GOOD
017366  104024                      ERROR   24
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF FLOATING
                                                        ;ONES (WHICH IS 400,1000,2000,4000,
                                                        ;10000,20000,40000,100000)
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;SEVENTH TIME
                                                        ;RHDB SHOULD HAVE 40000
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
017370                          81$:

017370  012737  000010  004664        MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                                    ;CHECK THAT RHDB HAS 100000
017376  012737  100000  001124        MOV     #100000,$GDDAT  ;GET GOOD = 0

017404  017737  164472  001126        MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
017412  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
017420  001401                      BEQ     83$             ;BRANCH IF GOOD
017422  104024                      ERROR   24
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF FLOATING
                                                        ;ONES (WHICH IS 400,1000,2000,4000,
                                                        ;10000,20000,40000,100000)
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;EIGHTH TIME
                                                        ;RHDB SHOULD HAVE 100000
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
```

```
5896  017424                          83$:
5897
5898
5899                                          ;CHECK THAT RHCS2 HAS IR
5900  017424  012737  000100  001124        MOV     #IR,$GDDAT      ;GET GOOD = 100
5901  017432  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
5902
5903  017440  017737  164424  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
5904  017446  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5905                                                                ;DATA WITH DATA READ FROM
5906                                                                ;RHCS2
5907  017454  001401                        BEQ     85$             ;BRANCH IF GOOD
5908  017456  104025                        ERROR   25
5909                                                                ;AFTER SETTING "CLR" BIT #5
5910                                                                ;IN RHCS2 TO INIT THE RH
5911                                                                ;AND WRITING A PATTERN OF FLOATING
5912                                                                ;ONES (WHICH IS 400,1000,2000,4000,
5913                                                                ;10000,20000,40000,100000)
5914                                                                ;INTO RHDB TO FILL IT
5915                                                                ;THEN ON READING RHDB THE
5916                                                                ;TOTAL 8 TIMES
5917                                                                ;RHCS2 SHOULD HAVE IR
5918                                                                ;=100
5919                                                                ;TOGETHER WITH UNIT NUMBER
5920                                                                ;BUT CONTAINED WHAT IS
5921                                                                ;GIVEN IN BAD RHCS2
5922  017460                          85$:
5923                                          ;CHECK THAT RHCS1 HAS DVA!RDY
5924  017460  012737  004200  001124        MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
5925
5926  017466  017737  164366  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
5927  017474  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5928                                                                ;DATA WITH DATA READ FROM
5929                                                                ;RHCS1
5930  017502  001401                        BEQ     87$             ;BRANCH IF GOOD
5931  017504  104026                        ERROR   26
5932                                                                ;AFTER SETTING "CLR" BIT #5
5933                                                                ;IN RHCS2 TO INIT THE RH
5934                                                                ;AND WRITING A PATTERN OF FLOATING
5935                                                                ;ONES (WHICH IS 400,1000,2000,4000,
5936                                                                ;10000,20000,40000,100000)
5937                                                                ;INTO RHDB TO FILL IT
5938                                                                ;THEN ON READING RHDB THE
5939                                                                ;TOTAL 8 TIMES
5940                                                                ;RHCS1 SHOULD HAVE DVA!RDY
5941                                                                ;=4200
5942                                                                ;BUT CONTAINED WHAT IS
5943                                                                ;GIVEN IN BAD RHCS1
5944  017506                          87$:
5945                                          ;CHECK THAT RHCS3 HAS 0
5946  017506  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
5947
5948  017514  017737  164412  001126        MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
5949  017522  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5950                                                                ;DATA WITH DATA READ FROM
5951                                                                ;RHCS3
```

```
                                        BEQ     99$             ;BRANCH IF GOOD
                                        ERROR   27

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ONES (WHICH IS 400,1000,2000,4000,
                                                                ;10000,20000,40000,100000)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHCS3 SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
017534                          89$:
                                        ;CHECK THAT RHBA HAS 0
017534  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

017542  017737  164316  001126        MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
017550  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBA
017556  001401                        BEQ     91$             ;BRANCH IF GOOD
017560  104030                        ERROR   30

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ONES (WHICH IS 400,1000,2000,4000,
                                                                ;10000,20000,40000,100000)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
017562                          91$:
                                        ;CHECK THAT RHBAE HAS 0
017562  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

017570  017737  164334  001126        MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
017576  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
017604  001401                        BEQ     93$             ;BRANCH IF GOOD
017606  104031                        ERROR   31

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ONES (WHICH IS 400,1000,2000,4000,
                                                                ;10000,20000,40000,100000)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
017610                          93$:
```

```
                                        ;CHECK THAT RHWC HAS 0
017610  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

017616  017737  164240  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
017624  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
017632  001401                          BEQ     95$             ;BRANCH IF GOOD
017634  104032                          ERROR   32

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ONES (WHICH IS 400,1000,2000,4000,
                                                                ;10000,20000,40000,100000)
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
017636                          95$:


                                ;;*****************************************************************
                                ;*TEST 17        SILO TEST 7 (FLOATING 0 IN LOWER BYTE)

                                ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
                                ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                ;*      TOGETHER WITH UNIT NUMBER
                                ;*      EIGHT WORDS OF FLOATING ZEROS 177776, 177775, 177773,
                                ;*      177767, 177757, 177737, 177677, 177577
                                ;*      IS WRITTEN INTO RHDB
                                ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                ;*      INSTRUCTION CALLED "WAT"
                                ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                ;*      THEN RHDB IS READ AND COMPARED TO HAVE THE
                                ;*      RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                ;*      THEN RHCS2 IS CHECKED TO HAVE "IR"
                                ;*      AND UNIT NUMBER
                                ;*      THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                ;*      CHECKED TO HAVE THE APPROPIATE VALUE
                                ;*****************************************************************
017636  000004                  TST17:  SCOPE
017640  012706  001000                  MOV     #STACK,SP       ;RESET STACK
017644  012737  000017  004656          MOV     #17,@#TSTNM     ;SAVE TEST NUMBER

017652  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS2 HAS IR
017656  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
017664  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
6064  017672  017737  164172  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
6065  017700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6066                                                                  ;DATA WITH DATA READ FROM
6067                                                                  ;RHCS2
6068  017706  001401                          BEQ     65$             ;BRANCH IF GOOD
6069  017710  104006                          ERROR   6
6070                                                                  ;AFTER SETTING "CLR" BIT #5
6071                                                                  ;IN RHCS2 TO INIT THE RH
6072                                                                  ;AND HAVING DONE NOTHING
6073                                                                  ;ELSE
6074                                                                  ;RHCS2 SHOULD HAVE IR
6075                                                                  ;=100
6076                                                                  ;TOGETHER WITH UNIT NUMBER
6077                                                                  ;BUT CONTAINED WHAT IS
6078                                                                  ;GIVEN IN BAD RHCS2
6079  017712                          65$:
6080
6081                                          ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
6082                                          ;DATA IS FLOATING ZEROS 177776, 177775, 177773, 177767, 177757, 177737,
6083                                          ;177677, 177577
6084  017712  012701  177776                 MOV     #177776,R1      ;GETTING READY TO FLOAT ZEROS
6085  017716  012702  000010                 MOV     #8.,R2          ;COUNTER -8 DATA WORDS
6086  017722  010177  164154          1$:    MOV     R1,@RHDB        ;WRITE INTO RHDB
6087  017726  000261                         SEC                     ;SET CARRY
6088  017730  006101                         ROL     R1              ;GET ZERO ONE BIT LEFT
6089  017732  005302                         DEC     R2              ;COUNT TO 8
6090  017734  001372                         BNE     1$              ;BRANCH IF 8 NOT DONE
6091                                          ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
6092  017736  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
6093  017740  004070                         RHCS2                   ;AND WAIT FOR OR BIT IN
6094  017742  000200                         OR                      ;RHCS2 REGISTER
6095                                                                  ;IF ERROR OCCURS HERE
6096                                                                  ;IT MEANS "OR" DID NOT
6097                                                                  ;SET FOR THE FULL COUNT
6098                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
6099                                                                  ;THIS TIME IS APPROXIMATELY
6100                                                                  ;LARGER THAN 500 MILLISECONDS
6101
6102
6103                                          ;CHECK THAT RHCS2 HAS OR
6104  017744  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
6105  017752  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
6106
6107  017760  017737  164104  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
6108  017766  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6109                                                                  ;DATA WITH DATA READ FROM
6110                                                                  ;RHCS2
6111  017774  001401                          BEQ     67$             ;BRANCH IF GOOD
6112  017776  104023                          ERROR   23
6113                                                                  ;AFTER SETTING "CLR" BIT #5
6114                                                                  ;IN RHCS2 TO INIT THE RH
6115                                                                  ;AND HAVING WRITTEN A PATTERN
6116                                                                  ;OF FLOATING ZEROS - 177776, 177775,
6117                                                                  ;177773, 177767, 177757, 177737, 177677, 177577
6118                                                                  ;INTO RHDB TO FILL IT
6119                                                                  ;RHCS2 SHOULD HAVE OR
```

```
                                                                ;=200
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
020000                                  67$:

020000  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177776
020006  012737  177776  001124          MOV     #177776,$GDDAT  ;GET GOOD = 0

020014  017737  164062  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020022  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
020030  001401                          BEQ     69$             ;BRANCH IF GOOD
020032  104024                          ERROR   24

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS - 177776, 177775, 177773, 177767,
                                                                ;177757, 177737, 177677, 177577
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;FIRST TIME
                                                                ;RHDB SHOULD HAVE 177776
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
020034                                  69$:


020034  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177775
020042  012737  177775  001124          MOV     #177775,$GDDAT  ;GET GOOD = 0

020050  017737  164026  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020056  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
020064  001401                          BEQ     71$             ;BRANCH IF GOOD
020066  104024                          ERROR   24

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS - 177776, 177775, 177773, 177767,
                                                                ;177757, 177737, 177677, 177577
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHDB SHOULD HAVE 177775
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
020070                                  71$:
```

```
6176  020070  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
6177                                          ;CHECK THAT RHDB HAS 177773
6178  020076  012737  177773  001124          MOV     #177773,$GDDAT  ;GET GOOD = 0
6179
6180  020104  017737  163772  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6181  020112  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6182                                                                  ;DATA WITH DATA READ FROM
6183                                                                  ;RHDB
6184  020120  001401                          BEQ     73$             ;BRANCH IF GOOD
6185  020122  104024                          ERROR   24
6186                                                                  ;AFTER SETTING "CLR" BIT #5
6187                                                                  ;IN RHCS2 TO INIT THE RH
6188                                                                  ;AND WRITING A PATTERN OF FLOATING
6189                                                                  ;ZEROS - 177776, 177775, 177773, 177767,
6190                                                                  ;177757, 177737, 177677, 177577
6191                                                                  ;INTO RHDB TO FILL IT
6192                                                                  ;THEN ON READING RHDB THE
6193                                                                  ;THIRD TIME
6194                                                                  ;RHDB SHOULD HAVE 177773
6195                                                                  ;BUT CONTAINED WHAT IS
6196                                                                  ;GIVEN IN BAD RHDB
6197  020124                          73$:
6198
6199
6200  020124  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
6201                                          ;CHECK THAT RHDB HAS 177767
6202  020132  012737  177767  001124          MOV     #177767,$GDDAT  ;GET GOOD = 0
6203
6204  020140  017737  163736  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6205  020146  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6206                                                                  ;DATA WITH DATA READ FROM
6207                                                                  ;RHDB
6208  020154  001401                          BEQ     75$             ;BRANCH IF GOOD
6209  020156  104024                          ERROR   24
6210                                                                  ;AFTER SETTING "CLR" BIT #5
6211                                                                  ;IN RHCS2 TO INIT THE RH
6212                                                                  ;AND WRITING A PATTERN OF FLOATING
6213                                                                  ;ZEROS - 177776, 177775, 177773, 177767,
6214                                                                  ;177757, 177737, 177677, 177577
6215                                                                  ;INTO RHDB TO FILL IT
6216                                                                  ;THEN ON READING RHDB THE
6217                                                                  ;FOURTH TIME
6218                                                                  ;RHDB SHOULD HAVE 177757
6219                                                                  ;BUT CONTAINED WHAT IS
6220                                                                  ;GIVEN IN BAD RHDB
6221  020160                          75$:
6222
6223
6224  020160  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
6225                                          ;CHECK THAT RHDB HAS 177757
6226  020166  012737  177757  001124          MOV     #177757,$GDDAT  ;GET GOOD = 0
6227
6228  020174  017737  163702  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6229  020202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6230                                                                  ;DATA WITH DATA READ FROM
6231                                                                  ;RHDB
```

```
6232  020210  001401                        BEQ     77$             ;BRANCH IF GOOD
6233  020212  104024                        ERROR   24
6234                                                                ;AFTER SETTING "CLR" BIT #5
6235                                                                ;IN RHCS2 TO INIT THE RH
6236                                                                ;AND WRITING A PATTERN OF FLOATING
6237                                                                ;ZEROS - 177776, 177775, 177773, 177767,
6238                                                                ;177757, 177737, 177677, 177577
6239                                                                ;INTO RHDB TO FILL IT
6240                                                                ;THEN ON READING RHDB THE
6241                                                                ;FIFTH TIME
6242                                                                ;RHDB SHOULD HAVE 177757
6243                                                                ;BUT CONTAINED WHAT IS
6244                                                                ;GIVEN IN BAD RHDB
6245  020214                        77$:
6246
6247
6248  020214  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
6249                                          ;CHECK THAT RHD3 HAS 177737
6250  020222  012737  177737  001124          MOV     #177737,$GDDAT  ;GET GOOD = 0
6251
6252  020230  017737  163646  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6253  020236  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6254                                                                ;DATA WITH DATA READ FROM
6255                                                                ;RHDB
6256  020244  001401                        BEQ     79$             ;BRANCH IF GOOD
6257  020246  104024                        ERROR   24
6258                                                                ;AFTER SETTING "CLR" BIT #5
6259                                                                ;IN RHCS2 TO INIT THE RH
6260                                                                ;AND WRITING A PATTERN OF FLOATING
6261                                                                ;ZEROS - 177776, 177775, 177773, 177767,
6262                                                                ;177757, 177737, 177677, 177577
6263                                                                ;INTO RHDB TO FILL IT
6264                                                                ;THEN ON READING RHDB THE
6265                                                                ;SIXTH TIME
6266                                                                ;RHDB SHOULD HAVE 177737
6267                                                                ;BUT CONTAINED WHAT IS
6268                                                                ;GIVEN IN BAD RHDB
6269  020250                        79$:
6270
6271
6272  020250  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
6273                                          ;CHECK THAT RHDB HAS 177677
6274  020256  012737  177677  001124          MOV     #177677,$GDDAT  ;GET GOOD = 0
6275
6276  020264  017737  163612  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6277  020272  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6278                                                                ;DATA WITH DATA READ FROM
6279                                                                ;RHDB
6280  020300  001401                        BEQ     81$             ;BRANCH IF GOOD
6281  020302  104024                        ERROR   24
6282                                                                ;AFTER SETTING "CLR" BIT #5
6283                                                                ;IN RHCS2 TO INIT THE RH
6284                                                                ;AND WRITING A PATTERN OF FLOATING
6285                                                                ;ZEROS - 177776, 177775, 177773, 177767,
6286                                                                ;177757, 177737, 177677, 177577
6287                                                                ;INTO RHDB TO FILL IT
```

```
                                                        ;THEN ON READING RHDB THE
                                                        ;SEVENTH TIME
                                                        ;RHDB SHOULD HAVE 177677
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
020304                          81$:

020304  012737  000010  004664        MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                                      ;CHECK THAT RHDB HAS 177577
020312  012737  177577  001124        MOV     #177577,$GDDAT  ;GET GOOD = 0

020320  017737  163556  001126        MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020326  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020334  001401                      BEQ     83$             ;BRANCH IF GOOD
020336  104024                      ERROR   24
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF FLOATING
                                                        ;ZEROS - 177776, 177775, 177773, 177767,
                                                        ;177757, 177737, 177677, 177577
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;EIGHTH TIME
                                                        ;RHDB SHOULD HAVE 177577
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
020340                          83$:


                                      ;CHECK THAT RHCS2 HAS IR
020340  012737  000100  001124        MOV     #IR,$GDDAT      ;GET GOOD = 100
020346  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

020354  017737  163510  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
020362  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
020370  001401                      BEQ     85$             ;BRANCH IF GOOD
020372  104025                      ERROR   25
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF FLOATING
                                                        ;ZEROS - 177776, 177775, 177773, 177767,
                                                        ;177757, 177737, 177677, 177577
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;TOTAL 8 TIMES
                                                        ;RHCS2 SHOULD HAVE IR
                                                        ;=100
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
020374                          85$:
```

```
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
020374  012737  004200  001124        MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

020402  017737  163452  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
020410  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHCS1
020416  001401                        BEQ     87$             ;BRANCH IF GOOD
020420  104026                        ERROR   26

                                                              ;AFTER SETTING "CLR" BIT #5
                                                              ;IN RHCS2 TO INIT THE RH
                                                              ;AND WRITING A PATTERN OF FLOATING
                                                              ;ZEROS - 177776, 177775, 177773, 177767,
                                                              ;177757, 177737, 177677, 177577
                                                              ;INTO RHDB TO FILL IT
                                                              ;THEN ON READING RHDB THE
                                                              ;TOTAL 8 TIMES
                                                              ;RHCS1 SHOULD HAVE DVA!RDY
                                                              ;=4200
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHCS1
020422                          87$:
                                        ;CHECK THAT RHCS3 HAS 0
020422  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

020430  017737  163476  001126        MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
020436  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHCS3
020444  001401                        BEQ     89$             ;BRANCH IF GOOD
020446  104027                        ERROR   27

                                                              ;AFTER SETTING "CLR" BIT #5
                                                              ;IN RHCS2 TO INIT THE RH
                                                              ;AND WRITING A PATTERN OF FLOATING
                                                              ;ZEROS - 177776, 177775, 177773, 177767,
                                                              ;177757, 177737, 177677, 177577
                                                              ;INTO RHDB TO FILL IT
                                                              ;THEN ON READING RHDB THE
                                                              ;TOTAL 8 TIMES
                                                              ;RHCS3 SHOULD HAVE 0
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHCS3
020450                          89$:
                                        ;CHECK THAT RHBA HAS 0
020450  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

020456  017737  163402  001126        MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
020464  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHBA
020472  001401                        BEQ     91$             ;BRANCH IF GOOD
020474  104030                        ERROR   30

                                                              ;AFTER SETTING "CLR" BIT #5
                                                              ;IN RHCS2 TO INIT THE RH
                                                              ;AND WRITING A PATTERN OF FLOATING
                                                              ;ZEROS - 177776, 177775, 177773, 177767,
```

```
                                                                ;177757, 177737, 177677, 177577
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
020476                                  91$:
                                                ;CHECK THAT RHBAE HAS 0
020476  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

020504  017737  163420  001126                  MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
020512  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
020520  001401                                  BEQ     93$             ;BRANCH IF GOOD
020522  104031                                  ERROR   31

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS - 177776, 177775, 177773, 177767,
                                                                ;177757, 177737, 177677, 177577
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
020524                                  93$:
                                                ;CHECK THAT RHWC HAS 0
020524  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

020532  017737  163324  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
020540  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
020546  001401                                  BEQ     95$             ;BRANCH IF GOOD
020550  104032                                  ERROR   32

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS - 177776, 177775, 177773, 177767,
                                                                ;177757, 177737, 177677, 177577
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
020552                                  95$:


                                ;;*****************************************************************
                                ;*TEST 20        SILO TEST 8 (FLOATING ZEROS IN UPPER BYTE)

                                ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
```

```
                                        ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
                                        ;*      TOGETHER WITH UNIT NUMBER
                                        ;*      EIGHT WORDS, A PATTERN OF FLOATING ZEROS IN UPPER BYTE
                                        ;*      177377, 177677, 175777, 173777, 167777, 157777, 137777, 77777
                                        ;*      IS WRITTEN INTO RHDB
                                        ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
                                        ;*      INSTRUCTION CALLED "WAT"
                                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
                                        ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
                                        ;*      THEN RHDB IS READ AND COMPARED TO HAVE THE
                                        ;*      RIGHT VALUE AFTER EACH OF THE EIGHT READS.
                                        ;*      THEN RHCS2 IS CHECKED TO HAVE "IR"
                                        ;*      AND UNIT NUMBER
                                        ;*      THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
                                        ;*      CHECKED TO HAVE THE APPROPIATE VALUE
                                        ;*****************************************************************
020552  000004                  TST20:  SCOPE
020554  012706  001000                  MOV     #STACK,SP       ;RESET STACK
020560  012737  000020  004656          MOV     #20,@#TSTNM     ;SAVE TEST NUMBER

020566  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS2 HAS IR
020572  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
020600  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

020606  017737  163256  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
020614  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
020622  001401                          BEQ     65$             ;BRANCH IF GOOD
020624  104006                          ERROR   6

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND HAVING DONE NOTHING
                                                                ;ELSE
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
020626                          65$:


                                        ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
                                        ;A PATTERN OF FLOATING ZEROS IN UPPER BYTE
                                        ;DATA IS - 177377, 176777, 175777, 173777, 167777,
                                        ;157777, 137777, 77777
020626  012701  177377                  MOV     #177377,R1      ;GETTING READY TO FLOAT ZEROS
                                                                ;IN UPPER BYTE
020632  012702  000010                  MOV     #8.,R2          ;COUNTER -8 DATA WORDS
020636  010177  163240          1$:     MOV     R1,@RHDB        ;WRITE INTO RHDB
020642  000261                          SEC                     ;SET CARRY
020644  006101                          ROL     R1              ;GET ZERO ONE BIT LEFT
```

```
6512  020646  005302                          DEC     R2              ;COUNT TO 8
6513  020650  001372                          BNE     1$              ;BRANCH IF 8 NOT DONE
6514                                          ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
6515  020652  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
6516  020654  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
6517  020656  000200                          OR                      ;RHCS2 REGISTER
6518                                                                  ;IF ERROR OCCURS HERE
6519                                                                  ;IT MEANS "OR" DID NOT
6520                                                                  ;SET FOR THE FULL COUNT
6521                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
6522                                                                  ;THIS TIME IS APPROXIMATELY
6523                                                                  ;LARGER THAN 500 MILLISECONDS
6524
6525
6526                                          ;CHECK THAT RHCS2 HAS OR
6527  020660  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
6528  020666  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
6529
6530  020674  017737  163170  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
6531  020702  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6532                                                                  ;DATA WITH DATA READ FROM
6533                                                                  ;RHCS2
6534  020710  001401                          BEQ     67$             ;BRANCH IF GOOD
6535  020712  104023                          ERROR   23
6536                                                                  ;AFTER SETTING "CLR" BIT #5
6537                                                                  ;IN RHCS2 TO INIT THE RH
6538                                                                  ;AND HAVING WRITTEN A
6539                                                                  ;PATTERN OF FLOATING ZEROS IN UPPER BYTE
6540                                                                  ;177377, 176777, 175777, 173777,
6541                                                                  ;167777, 157777, 137777, 77777
6542                                                                  ;INTO RHDB TO FILL IT
6543                                                                  ;RHCS2 SHOULD HAVE OR
6544                                                                  ;=200
6545                                                                  ;TOGETHER WITH UNIT NUMBER
6546                                                                  ;BUT CONTAINED WHAT IS
6547                                                                  ;GIVEN IN BAD RHCS2
6548  020714                          67$:
6549
6550
6551
6552  020714  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
6553                                          ;CHECK THAT RHDB HAS 177377
6554  020722  012737  177377  001124          MOV     #177377,$GDDAT  ;GET GOOD = 0
6555
6556  020730  017737  163146  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6557  020736  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6558                                                                  ;DATA WITH DATA READ FROM
6559                                                                  ;RHDB
6560  020744  001401                          BEQ     69$             ;BRANCH IF GOOD
6561  020746  104024                          ERROR   24
6562                                                                  ;AFTER SETTING "CLR" BIT #5
6563                                                                  ;IN RHCS2 TO INIT THE RH
6564                                                                  ;AND WRITING A PATTERN OF FLOATING
6565                                                                  ;ZEROS IN UPPER BYTE
6566                                                                  ;177377, 176777, 175777, 173777,
6567                                                                  ;167777, 157777, 137777, 77777
```

```
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;FIRST TIME
                                                                ;RHDB SHOULD HAVE 177377
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
020750                                  69$:

020750  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 176777
020756  012737  176777  001124          MOV     #176777,$GDDAT  ;GET GOOD = 0

020764  017737  163112  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020772  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
021000  001401                          BEQ     71$             ;BRANCH IF GOOD
021002  104034                          ERROR   34
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS IN UPPER BYTE
                                                                ;177377, 176777, 175777, 173777,
                                                                ;167777, 157777, 137777, 77777
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;SECOND TIME
                                                                ;RHDB SHOULD HAVE 176777
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
021004                                  71$:

021004  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 175777
021012  012737  175777  001124          MOV     #175777,$GDDAT  ;GET GOOD = 0

021020  017737  163056  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
021026  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
021034  001401                          BEQ     73$             ;BRANCH IF GOOD
021036  104024                          ERROR   24
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS IN UPPER BYTE
                                                                ;177377, 176777, 175777, 173777,
                                                                ;167777, 157777, 137777, 77777
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;THIRD TIME
                                                                ;RHDB SHOULD HAVE 175777
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
```

```
6624  021040                          73$:
6625
6626
6627  021040  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
6628                                          ;CHECK THAT RHDB HAS 173777
6629  021046  012737  173777  001124          MOV     #173777,$GDDAT  ;GET GOOD = 0
6630
6631  021054  017737  163022  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6632  021062  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6633                                                                  ;DATA WITH DATA READ FROM
6634                                                                  ;RHDB
6635  021070  001401                          BEQ     75$             ;BRANCH IF GOOD
6636  021072  104024                          ERROR   24
6637                                                                  ;AFTER SETTING "CLR" BIT #5
6638                                                                  ;IN RHCS2 TO INIT THE RH
6639                                                                  ;AND WRITING A PATTERN OF FLOATING
6640                                                                  ;ZEROS IN UPPER BYTE
6641                                                                  ;177377, 176777, 175777, 173777,
6642                                                                  ;167777, 157777, 137777, 77777
6643                                                                  ;INTO RHDB TO FILL IT
6644                                                                  ;THEN ON READING RHDB THE
6645                                                                  ;FOURTH TIME
6646                                                                  ;RHDB SHOULD HAVE 173777
6647                                                                  ;BUT CONTAINED WHAT IS
6648                                                                  ;GIVEN IN BAD RHDB
6649  021074                          75$:
6650
6651
6652  021074  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
6653                                          ;CHECK THAT RHDB HAS 167777
6654  021102  012737  167777  001124          MOV     #167777,$GDDAT  ;GET GOOD = 0
6655
6656  021110  017737  162766  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6657  021116  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6658                                                                  ;DATA WITH DATA READ FROM
6659                                                                  ;RHDB
6660  021124  001401                          BEQ     77$             ;BRANCH IF GOOD
6661  021126  104024                          ERROR   24
6662                                                                  ;AFTER SETTING "CLR" BIT #5
6663                                                                  ;IN RHCS2 TO INIT THE RH
6664                                                                  ;AND WRITING A PATTERN OF FLOATING
6665                                                                  ;ZEROS IN UPPER BYTE
6666                                                                  ;177377, 176777, 175777, 173777,
6667                                                                  ;167777, 157777, 137777, 77777
6668                                                                  ;INTO RHDB TO FILL IT
6669                                                                  ;THEN ON READING RHDB THE
6670                                                                  ;FIFTH TIME
6671                                                                  ;RHDB SHOULD HAVE 167777
6672                                                                  ;BUT CONTAINED WHAT IS
6673                                                                  ;GIVEN IN BAD RHDB
6674  021130                          77$:
6675
6676
6677  021130  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
6678                                          ;CHECK THAT RHDB HAS 157777
6679  021136  012737  157777  001124          MOV     #157777,$GDDAT  ;GET GOOD = 0
```

```
                                                        ;READ RHDB FOR COMPARISON
021144  017737  162732  001126        MOV     @RHDB,$BDDAT
021152  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021160  001401                        BEQ     79$             ;BRANCH IF GOOD
021162  104024                        ERROR   24
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF FLOATING
                                                        ;ZEROS IN UPPER BYTE
                                                        ;177377, 176777, 175777, 173777,
                                                        ;167777, 157777, 137777, 77777
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;SIXTH TIME
                                                        ;RHDB SHOULD HAVE 157777
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
021164                  79$:

021164  012737  000007  004664        MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
                        ;CHECK THAT RHDB HAS 137777
021172  012737  137777  001124        MOV     #137777,$GDDAT  ;GET GOOD = 0

021200  017737  162676  001126        MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
021206  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021214  001401                        BEQ     81$             ;BRANCH IF GOOD
021216  104024                        ERROR   24
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND WRITING A PATTERN OF FLOATING
                                                        ;ZEROS IN UPPER BYTE
                                                        ;177377, 176777, 175777, 173777,
                                                        ;167777, 157777, 137777, 77777
                                                        ;INTO RHDB TO FILL IT
                                                        ;THEN ON READING RHDB THE
                                                        ;SEVENTH TIME
                                                        ;RHDB SHOULD HAVE 137777
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
021220                  81$:

021220  012737  000010  004664        MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                        ;CHECK THAT RHDB HAS 77777
021226  012737  077777  001124        MOV     #77777,$GDDAT   ;GET GOOD = 0

021234  017737  162642  001126        MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
021242  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021250  001401                        BEQ     83$             ;BRANCH IF GOOD
```

```
6736  021252  104024                          ERROR   24
6737                                                    ;AFTER SETTING "CLR" BIT #5
6738                                                    ;IN RHCS2 TO INIT THE RH
6739                                                    ;AND WRITING A PATTERN OF FLOATING
6740                                                    ;ZEROS IN UPPER BYTE
6741                                                    ;177377, 176777, 175777, 173777,
6742                                                    ;167777, 157777, 137777, 77777
6743                                                    ;INTO RHDB TO FILL IT
6744                                                    ;THEN ON READING RHDB THE
6745                                                    ;EIGHTH TIME
6746                                                    ;RHDB SHOULD HAVE 77777
6747                                                    ;BUT CONTAINED WHAT IS
6748                                                    ;GIVEN IN BAD RHDB
6749  021254                          83$:
6750
6751
6752                                    ;CHECK THAT RHCS2 HAS IR
6753  021254  012737  000100  001124    MOV     #IR,$GDDAT      ;GET GOOD = 100
6754  021262  053737  005010  001124    BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
6755
6756  021270  017737  162574  001126    MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
6757  021276  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6758                                                    ;DATA WITH DATA READ FROM
6759                                                    ;RHCS2
6760  021304  001401                      BEQ     85$             ;BRANCH IF GOOD
6761  021306  104025                      ERROR   25
6762                                                    ;AFTER SETTING "CLR" BIT #5
6763                                                    ;IN RHCS2 TO INIT THE RH
6764                                                    ;AND WRITING A PATTERN OF FLOATING
6765                                                    ;ZEROS IN UPPER BYTE
6766                                                    ;177377, 176777, 175777, 173777,
6767                                                    ;167777, 157777, 137777, 77777
6768                                                    ;INTO RHDB TO FILL IT
6769                                                    ;THEN ON READING RHDB THE
6770                                                    ;TOTAL 8 TIMES
6771                                                    ;RHCS2 SHOULD HAVE IR
6772                                                    ;=100
6773                                                    ;TOGETHER WITH UNIT NUMBER
6774                                                    ;BUT CONTAINED WHAT IS
6775                                                    ;GIVEN IN BAD RHCS2
6776  021310                          85$:
6777                                    ;CHECK THAT RHCS1 HAS DVA!RDY
6778  021310  012737  004200  001124    MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
6779
6780  021316  017737  162536  001126    MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
6781  021324  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6782                                                    ;DATA WITH DATA READ FROM
6783                                                    ;RHCS1
6784  021332  001401                      BEQ     87$             ;BRANCH IF GOOD
6785  021334  104026                      ERROR   26
6786                                                    ;AFTER SETTING "CLR" BIT #5
6787                                                    ;IN RHCS2 TO INIT THE RH
6788                                                    ;AND WRITING A PATTERN OF FLOATING
6789                                                    ;ZEROS IN UPPER BYTE
6790                                                    ;177377, 176777, 175777, 173777,
6791                                                    ;167777, 157777, 137777, 77777
```

```
                                                                        ;INTO RHDB TO FILL IT
                                                                        ;THEN ON READING RHDB THE
                                                                        ;TOTAL 8 TIMES
                                                                        ;RHCS1 SHOULD HAVE DVA!RDY
                                                                        ;=4200
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS1
021336                                  87$:
                                                ;CHECK THAT RHCS3 HAS 0
021336  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

021344  017737  162562  001126                  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
021352  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS3
021360  001401                                  BEQ     89$             ;BRANCH IF GOOD
021362  104027                                  ERROR   27

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;AND WRITING A PATTERN OF FLOATING
                                                                        ;ZEROS IN UPPER BYTE
                                                                        ;177377, 176777, 175777, 173777,
                                                                        ;167777, 157777, 137777, 77777
                                                                        ;INTO RHDB TO FILL IT
                                                                        ;THEN ON READING RHDB THE
                                                                        ;TOTAL 8 TIMES
                                                                        ;RHCS3 SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS3
021364                                  89$:
                                                ;CHECK THAT RHBA HAS 0
021364  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

021372  017737  162466  001126                  MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
021400  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBA
021406  001401                                  BEQ     91$             ;BRANCH IF GOOD
021410  104030                                  ERROR   30

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;AND WRITING A PATTERN OF FLOATING
                                                                        ;ZEROS IN UPPER BYTE
                                                                        ;177377, 176777, 175777, 173777,
                                                                        ;167777, 157777, 137777, 77777
                                                                        ;INTO RHDB TO FILL IT
                                                                        ;THEN ON READING RHDB THE
                                                                        ;TOTAL 8 TIMES
                                                                        ;RHBA SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBA
021412                                  91$:
                                                ;CHECK THAT RHBAE HAS 0
021412  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

021420  017737  162504  001126                  MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
```

```
021426  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
021434  001401                          BEQ     93$             ;BRANCH IF GOOD
021436  104031                          ERROR   31

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS IN UPPER BYTE
                                                                ;177377, 176777, 175777, 173777,
                                                                ;167777, 157777, 137777, 77777
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
021440                          93$:
                                        ;CHECK THAT RHWC HAS 0
021440  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

021446  017737  162410  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
021454  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
021462  001401                          BEQ     95$             ;BRANCH IF GOOD
021464  104032                          ERROR   32

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;AND WRITING A PATTERN OF FLOATING
                                                                ;ZEROS IN UPPER BYTE
                                                                ;177377, 176777, 175777, 173777,
                                                                ;167777, 157777, 137777, 77777
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;TOTAL 8 TIMES
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
021466                          95$:


;;*******************************************************************
;*TEST 21        RHCS1 - MCPE BIT #13 (PARITY LINE = 0)

;*      AN RH CLEAR (SET BIT #5 IN RHCS2) IS GIVEN TO
;*      CLEAR ALL DEVICE REGISTERS
;*      THEN RHER1 (ERROR REGISTER 1) IS CHECKED TO HAVE
;*      ZEROS
;*      SET "PAT" (BIT #4 IN RHCS2) TO INVERT PARITY CHECKING
;*      READ ANY DEVICE REGISTER - HERE RHER1
;*      READ AND CHECK RHCS1 TO CONTAIN SC (BIT #15)
;*      AND MCPE (BIT #13)
;*      WRITE "1" INTO TRE (BIT #14 IN RHCS1)
;*      READ RHCS1 SC, MCPE, AND RDY SHOULD BE SET
;*      GIVE AN RH CLEAR (CLR - BIT #5 IN RHCS2)
```

```
                                ;*      CHECK RHCS1 TO HAVE RDY
                                ;*      CHECK RHCS2 TO HAVE ONLY IR AND UNIT NUMBER
                                ;*      CHECK RHCS3, RHBA, RHBAE, RHWC TO HAVE APPROPIATE
                                ;*      VALUES.
                                ;;*****************************************************************
021466  000004                TST21:  SCOPE
021470  012706  001000                MOV     #STACK,SP       ;RESET STACK
021474  012737  000021  004656        MOV     #21,@#TSTNM     ;SAVE TEST NUMBER

021502  004737  040312                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                              ;SETUP UNIT NUBER
                                                              ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                ;CHECK THAT RHER1 HAS 0
021506  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

021514  017737  162354  001126        MOV     @RHER1,$BDDAT   ;READ RHER1 FOR COMPARISON
021522  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHER1
021530  001401                        BEQ     65$             ;BRANCH IF GOOD
021532  104033                        ERROR   33
                                                              ;AFTER SETTING "CLR" BIT #15
                                                              ;IN RHCS2 TO INIT THE RH
                                                              ;AND HAVING DONE NOTHING ELSE
                                                              ;RHER1 SHOULD HAVE 0
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHER1
021534                          65$:


                                ;TO INVERT THE PARITY CHECKING ON THE CONTROL BUS
                                ;"PAT" BIT #4 IN RHCS2 IS SET
021534  052777  000020  162326        BIS     #PAT,@RHCS2     ;SET PAT = 20 IN RHCS2


                                ;CHECK THAT RHER1 HAS 0
021542  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

021550  017737  162320  001126        MOV     @RHER1,$BDDAT   ;READ RHER1 FOR COMPARISON
021556  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHER1
021564  001401                        BEQ     67$             ;BRANCH IF GOOD
021566  104034                        ERROR   34

                                                              ;AFTER CLEARING THE RH AND
                                                              ;DEVICE BY AN RH CLEAR
                                                              ;AND READING RHER1 WITH
                                                              ;WRONG PARITY CHECKING
                                                              ;RHER1 SHOULD HAVE 0
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHER1
021570                          67$:


                                ;HAVING READ RHER1 WITH WRONG PARITY BEING
                                ;CHECKED MCPE - BIT #13 AND SC BIT #15 IN RHCS1 SHOULD BE SET
```

```
                                        ;CHECK THAT RHCS1 HAS 4200
021570  012737  004200  001124        MOV     #4200,$GDDAT    ;GET GOOD = 124200

021576  017737  162256  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
021604  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHCS1
021612  001401                        BEQ     69$             ;BRANCH IF GOOD
021614  104035                        ERROR   35
                                                              ;AFTER CLEARING THE RH AND
                                                              ;DEVICE BY AN RH CLEAR
                                                              ;AND READING RHER1 WITH
                                                              ;WRONG PARITY CHECKING
                                                              ;RHCS1 SHOULD HAVE 4200
                                                              ;=124200
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHCS1
021616                          69$:

                                        ;WRITE A ONE INTO TRE (RHCS1 - BIT #14)
                                        ;THIS SHOULD NOT CLEAR MCPE OR SC
021616  012777  040000  162234        MOV     #TRE,@RHCS1     ;WRITE "1" INTO TRE IN RHCS1


                                        ;CHECK THAT RHCS1 HAS 104200
021624  012737  104200  001124        MOV     #104200,$GDDAT  ;GET GOOD = 124200

021632  017737  162222  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
021640  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHCS1
021646  001401                        BEQ     71$             ;BRANCH IF GOOD
021650  104036                        ERROR   36
                                                              ;AFTER CLEARING THE RH AND
                                                              ;DEVICE REGISTERS BY AN
                                                              ;RH CLEAR
                                                              ;RHER1 WAS CHECKED TO HAVE
                                                              ;ZEROS
                                                              ;WRONG PARITY CHECKING WAS
                                                              ;SET BY "PAT" BIT IN RHCS2
                                                              ;RHER1 WAS READ TO SET
                                                              ;MCPE AND SC IN RHCS1
                                                              ;ON WRITING "1" INTO TRE
                                                              ;RHCS1 SHOULD HAVE 104200
                                                              ;=124200
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHCS1
021652                          71$:

                                        ;NOW ERRORS WILL BE CLEARED BY RH CLEAR
                                        ;SET "CLR" BIT #5 IN RHCS2
021652  012777  000040  162210        MOV     #CLR,@RHCS2     ;CLEAR BY RH INIT
021660  013777  005010  162202        MOV     UNIT,@RHCS2     ;REINSTATE UNIT NUMBER
```

```
                                                ;CHECK THAT RHCS1 HAS DVA!RDY
021666  012737  004200  001124                MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

021674  017737  162160  001126                MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
021702  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                      ;DATA WITH DATA READ FROM
                                                                      ;RHCS1
021710  001401                                BEQ     73$             ;BRANCH IF GOOD
021712  104037                                ERROR   37
                                                                      ;AFTER CLEARING THE RH AND
                                                                      ;DEVICE REGISTERS BY AN
                                                                      ;RH CLEAR
                                                                      ;RHER1 WAS CHECKED TO HAVE
                                                                      ;ZEROS
                                                                      ;WRONG PARITY CHECKING WAS
                                                                      ;SET BY "PAT" BIT IN RHCS2
                                                                      ;RHER1 WAS READ TO SET
                                                                      ;MCPE AND SC IN RHCS1
                                                                      ;ON WRITING "1" INTO TRE
                                                                      ;THEN SETTING "CLR" IN RHCS1
                                                                      ;RHCS1 SHOULD HAVE DVA!RDY
                                                                      ;=4200
                                                                      ;BUT CONTAINED WHAT IS
                                                                      ;GIVEN IN BAD RHCS1
021714                                  73$:
                                                ;CHECK THAT RHCS2 HAS IR
021714  012737  000100  001124                MOV     #IR,$GDDAT      ;GET GOOD = 100
021722  053737  005010  001124                BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

021730  017737  162134  001126                MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
021736  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                      ;DATA WITH DATA READ FROM
                                                                      ;RHCS2
021744  001401                                BEQ     75$             ;BRANCH IF GOOD
021746  104040                                ERROR   40
                                                                      ;AFTER CLEARING THE RH AND
                                                                      ;DEVICE REGISTERS BY AN
                                                                      ;RH CLEAR
                                                                      ;RHER1 WAS CHECKED TO HAVE
                                                                      ;ZEROS
                                                                      ;WRONG PARITY CHECKING WAS
                                                                      ;SET BY "PAT" BIT IN RHCS2
                                                                      ;RHER1 WAS READ TO SET
                                                                      ;MCPE AND SC IN RHCS1
                                                                      ;ON WRITING "1" INTO TRE
                                                                      ;THEN SETTING "CLR" IN RHCS1
                                                                      ;RHCS2 SHOULD HAVE IR
                                                                      ;=100
                                                                      ;TOGETHER WITH UNIT NUMBER
                                                                      ;BUT CONTAINED WHAT IS
                                                                      ;GIVEN IN BAD RHCS2
021750                                  75$:
                                                ;CHECK THAT RHCS3 HAS 0
021750  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0
```

```
7072  021756  017737  162150  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7073  021764  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7074                                                                  ;DATA WITH DATA READ FROM
7075                                                                  ;RHCS3
7076  021772  001401                          BEQ     77$             ;BRANCH IF GOOD
7077  021774  104041                          ERROR   41
7078                                                                  ;AFTER CLEARING THE RH AND
7079                                                                  ;DEVICE REGISTERS BY AN
7080                                                                  ;RH CLEAR
7081                                                                  ;RHER1 WAS CHECKED TO HAVE
7082                                                                  ;ZEROS
7083                                                                  ;WRONG PARITY CHECKING WAS
7084                                                                  ;SET BY "PAT" BIT IN RHCS2
7085                                                                  ;RHER1 WAS READ TO SET
7086                                                                  ;MCPE AND SC IN RHCS1
7087                                                                  ;ON WRITING "1" INTO TRE
7088                                                                  ;THEN SETTING "CLR" IN RHCS1
7089                                                                  ;RHCS3 SHOULD HAVE 0
7090                                                                  ;BUT CONTAINED WHAT IS
7091                                                                  ;GIVEN IN BAD RHCS3
7092  021776                          77$:
7093                                          ;CHECK THAT RHBA HAS 0
7094  021776  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
7095
7096  022004  017737  162054  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
7097  022012  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7098                                                                  ;DATA WITH DATA READ FROM
7099                                                                  ;RHBA
7100  022020  001401                          BEQ     79$             ;BRANCH IF GOOD
7101  022022  104042                          ERROR   42
7102                                                                  ;AFTER CLEARING THE RH AND
7103                                                                  ;DEVICE REGISTERS BY AN
7104                                                                  ;RH CLEAR
7105                                                                  ;RHER1 WAS CHECKED TO HAVE
7106                                                                  ;ZEROS
7107                                                                  ;WRONG PARITY CHECKING WAS
7108                                                                  ;SET BY "PAT" BIT IN RHCS2
7109                                                                  ;RHER1 WAS READ TO SET
7110                                                                  ;MCPE AND SC IN RHCS1
7111                                                                  ;ON WRITING "1" INTO TRE
7112                                                                  ;THEN SETTING "CLR" IN RHCS1
7113                                                                  ;RHBA SHOULD HAVE 0
7114                                                                  ;BUT CONTAINED WHAT IS
7115                                                                  ;GIVEN IN BAD RHBA
7116  022024                          79$:
7117                                          ;CHECK THAT RHBAE HAS 0
7118  022024  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
7119
7120  022032  017737  162072  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
7121  022040  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7122                                                                  ;DATA WITH DATA READ FROM
7123                                                                  ;RHBAE
7124  022046  001401                          BEQ     81$             ;BRANCH IF GOOD
7125  022050  104043                          ERROR   43
7126                                                                  ;AFTER CLEARING THE RH AND
7127                                                                  ;DEVICE REGISTERS BY AN
```

```
                                                                ;RH CLEAR
                                                                ;RHER1 WAS CHECKED TO HAVE
                                                                ;ZEROS
                                                                ;WRONG PARITY CHECKING WAS
                                                                ;SET BY "PAT" BIT IN RHCS2
                                                                ;RHER1 WAS READ TO SET
                                                                ;MCPE AND SC IN RHCS1
                                                                ;ON WRITING "1" INTO TRE
                                                                ;THEN SETTING "CLR" IN RHCS1
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
022052                                  81$:
                                                ;CHECK THAT RHWC HAS 0
022052  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

022060  017737  161776  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
022066  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
022074  001401                          BEQ     83$             ;BRANCH IF GOOD
022076  104044                          ERROR   44

                                                                ;AFTER CLEARING THE RH AND
                                                                ;DEVICE REGISTERS BY AN
                                                                ;RH CLEAR
                                                                ;RHER1 WAS CHECKED TO HAVE
                                                                ;ZEROS
                                                                ;WRONG PARITY CHECKING WAS
                                                                ;SET BY "PAT" BIT IN RHCS2
                                                                ;RHER1 WAS READ TO SET
                                                                ;MCPE AND SC IN RHCS1
                                                                ;ON WRITING "1" INTO TRE
                                                                ;THEN SETTING "CLR" IN RHCS1
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
022100                                  83$:


                                        ;;*****************************************************************
                                        ;*TEST 22       RHCS1 - MCPE BIT #13 (PARITY LINE = 1)

                                        ;*      AN RH CLEAR (SET BIT #5 IN RHCS2) IS GIVEN TO CLEAR
                                        ;*      ALL DEVICE REGISTERS
                                        ;*      WRITE A ONE INTO DISK ADDRESS REGISTER (RHDA)
                                        ;*      (IN TAPE DRIVES CALLED REGISTER)
                                        ;*      SET "PAT" (RHCS2 BIT #4) THIS WILL INVERT THE PARITY CHECKING
                                        ;*      READ RHDA AND COMPARE IT HAS ONE IN IT
                                        ;*      THIS READING SHOULD SET SC, MCPE IN RHCS1
                                        ;*      CHECK RHCS1 TO HAVE ONLY RDY SET
                                        ;*      CHECK RHCS2, RHCS3, RHBA, RHBAE, RHWC TO HAVE APPROPIATE
                                        ;*      VALUES
                                        ;;*****************************************************************
022100  000004                        TST22:  SCOPE
022102  012706  001000                        MOV     #STACK,SP       ;RESET STACK
022106  012737  000022  004656                MOV     #22,@#TSTNM     ;SAVE TEST NUMBER
```

```
7184
7185  022114  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
7186                                                                  ;SETUP UNIT NUBER
7187                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
7188
7189                                          ;WRITE A ONE INTO RHDA - DISK ADDRESS REGISTER
7190                                          ;THIS REGISTER IN TAPE DRIVES IS CALLED "RHFC FRAME COUNT"
7191                                          ;AFTER WRITING THIS "1" THEN ON READING THIS REGISTER
7192                                          ;THE CONTROL PARITY LINE WILL HAVE ZERO
7193  022120  012777  000001  161740          MOV     #BIT0,@RHDA     ;WRITE "1" INTO RHDA
7194                                                                  ;IN TAPE DRIVES CALLED RHFC
7195
7196                                          ;INVERT THE PARITY TO BE CHECKED BY SETTING "PAT" BIT #4 IN RHCS2
7197  022126  052777  000020  161734          BIS     #PAT,@RHCS2     ;SET PAT IN RHCS2
7198
7199
7200                                          ;CHECK THAT RHDA HAS BIT0
7201  022134  012737  000001  001124          MOV     #BIT0,$GDDAT    ;GET GOOD = 1
7202
7203  022142  017737  161720  001126          MOV     @RHDA,$BDDAT    ;READ RHDA FOR COMPARISON
7204  022150  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7205                                                                  ;DATA WITH DATA READ FROM
7206                                                                  ;RHDA
7207  022156  001401                          BEQ     65$             ;BRANCH IF GOOD
7208  022160  104045                          ERROR   45
7209                                                                  ;AFTER AN RH CLEAR "1"
7210                                                                  ;WAS WRITTEN INTO ADDRESS REGISTER
7211                                                                  ;(FRAME COUNT IN TAPE)
7212                                                                  ;PAT IN RHCS2 WAS SET
7213                                                                  ;THEN THE ADDRESS REGISTER
7214                                                                  ;WAS READ BACK
7215                                                                  ;RHDA SHOULD HAVE BIT0
7216                                                                  ;=1
7217                                                                  ;BUT CONTAINED WHAT IS
7218                                                                  ;GIVEN IN BAD RHDA
7219  022162                          65$:
7220
7221                                          ;CHECK THAT RHCS1 HAS 4200
7222  022162  012737  004200  001124          MOV     #4200,$GDDAT    ;GET GOOD = 124000
7223
7224  022170  017737  161664  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
7225  022176  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7226                                                                  ;DATA WITH DATA READ FROM
7227                                                                  ;RHCS1
7228  022204  001401                          BEQ     67$             ;BRANCH IF GOOD
7229  022206  104046                          ERROR   46
7230                                                                  ;AFTER AN RH CLEAR "1"
7231                                                                  ;WAS WRITTEN INTO ADDRESS REGISTER
7232                                                                  ;(FRAME COUNT IN TAPE)
7233                                                                  ;PAT IN RHCS2 WAS SET
7234                                                                  ;THEN THE ADDRESS REGISTER
7235                                                                  ;WAS READ BACK
7236                                                                  ;RHCS1 SHOULD HAVE 4200
7237                                                                  ;=124000
7238                                                                  ;BUT CONTAINED WHAT IS
7239                                                                  ;GIVEN IN BAD RHCS1
```

```
7240  022210                          67$:
7241                                        ;CHECK THAT RHCS2 HAS IR!PAT
7242  022210  012737  000120  001124        MOV     #IR!PAT,$GDDAT  ;GET GOOD = 120
7243  022216  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7244
7245  022224  017737  161640  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7246  022232  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7247                                                                ;DATA WITH DATA READ FROM
7248                                                                ;RHCS2
7249  022240  001401                        BEQ     69$             ;BRANCH IF GOOD
7250  022242  104047                        ERROR   47
7251                                                                ;AFTER AN RH CLEAR "1"
7252                                                                ;WAS WRITTEN INTO ADDRESS REGISTER
7253                                                                ;(FRAME COUNT IN TAPE)
7254                                                                ;PAT IN RHCS2 WAS SET
7255                                                                ;THEN THE ADDRESS REGISTER
7256                                                                ;WAS READ BACK
7257                                                                ;RHCS2 SHOULD HAVE IR!PAT
7258                                                                ;=120
7259                                                                ;TOGETHER WITH UNIT NUMBER
7260                                                                ;BUT CONTAINED WHAT IS
7261                                                                ;GIVEN IN BAD RHCS2
7262  022244                          69$:
7263                                        ;CHECK THAT RHCS3 HAS 0
7264  022244  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
7265
7266  022252  017737  161654  001126        MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7267  022260  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7268                                                                ;DATA WITH DATA READ FROM
7269                                                                ;RHCS3
7270  022266  001401                        BEQ     71$             ;BRANCH IF GOOD
7271  022270  104050                        ERROR   50
7272                                                                ;AFTER AN RH CLEAR "1"
7273                                                                ;WAS WRITTEN INTO ADDRESS REGISTER
7274                                                                ;(FRAME COUNT IN TAPE)
7275                                                                ;PAT IN RHCS2 WAS SET
7276                                                                ;THEN THE ADDRESS REGISTER
7277                                                                ;WAS READ BACK
7278                                                                ;RHCS3 SHOULD HAVE 0
7279                                                                ;BUT CONTAINED WHAT IS
7280                                                                ;GIVEN IN BAD RHCS3
7281  022272                          71$:
7282                                        ;CHECK THAT RHBA HAS 0
7283  022272  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
7284
7285  022300  017737  161560  001126        MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
7286  022306  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7287                                                                ;DATA WITH DATA READ FROM
7288                                                                ;RHBA
7289  022314  001401                        BEQ     73$             ;BRANCH IF GOOD
7290  022316  104051                        ERROR   51
7291                                                                ;AFTER AN RH CLEAR "1"
7292                                                                ;WAS WRITTEN INTO ADDRESS REGISTER
7293                                                                ;(FRAME COUNT IN TAPE)
7294                                                                ;PAT IN RHCS2 WAS SET
7295                                                                ;THEN THE ADDRESS REGISTER
```

```
7296                                                                    ;WAS READ BACK
7297                                                                    ;RHBA SHOULD HAVE 0
7298                                                                    ;BUT CONTAINED WHAT IS
7299                                                                    ;GIVEN IN BAD RHBA
7300  022320                                  73$:
7301                                                  ;CHECK THAT RHBAE HAS 0
7302  022320  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
7303
7304  022326  017737  161576  001126                  MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
7305  022334  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7306                                                                          ;DATA WITH DATA READ FROM
7307                                                                          ;RHBAE
7308  022342  001401                                  BEQ     75$             ;BRANCH IF GOOD
7309  022344  104052                                  ERROR   52
7310                                                                    ;AFTER AN RH CLEAR "1"
7311                                                                    ;WAS WRITTEN INTO ADDRESS REGISTER
7312                                                                    ;(FRAME COUNT IN TAPE)
7313                                                                    ;PAT IN RHCS2 WAS SET
7314                                                                    ;THEN THE ADDRESS REGISTER
7315                                                                    ;WAS READ BACK
7316                                                                    ;RHBAE SHOULD HAVE 0
7317                                                                    ;BUT CONTAINED WHAT IS
7318                                                                    ;GIVEN IN BAD RHBAE
7319  022346                                  75$:
7320                                                  ;CHECK THAT RHWC HAS 0
7321  022346  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
7322
7323  022354  017737  161502  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
7324  022362  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7325                                                                          ;DATA WITH DATA READ FROM
7326                                                                          ;RHWC
7327  022370  001401                                  BEQ     77$             ;BRANCH IF GOOD
7328  022372  104053                                  ERROR   53
7329                                                                    ;AFTER AN RH CLEAR "1"
7330                                                                    ;WAS WRITTEN INTO ADDRESS REGISTER
7331                                                                    ;(FRAME COUNT IN TAPE)
7332                                                                    ;PAT IN RHCS2 WAS SET
7333                                                                    ;THEN THE ADDRESS REGISTER
7334                                                                    ;WAS READ BACK
7335                                                                    ;RHWC SHOULD HAVE 0
7336                                                                    ;BUT CONTAINED WHAT IS
7337                                                                    ;GIVEN IN BAD RHWC
7338  022374                                  77$:
7339
7340
7341                                          ;;*****************************************************************
7342                                          ;*TEST 23       TEST DUPLICATED A16 (RHCS1 BIT #8)
7343
7344                                          ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
7345                                          ;*      MOVE 400 (ONE INTO A16) IN RHCS1
7346                                          ;*      READ RHCS1 TO CONTAIN 600 (BIT #8 AND RDY)
7347                                          ;*      READ RHBAE TO CONTAIN "1" (BIT #0 HIGH)
7348                                          ;*
7349                                          ;*
7350                                          ;*      MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #0)
7351                                          ;*      READ RHBAE TO CONTAIN 0
```

```
7352                                    ;*      READ RHCS1 TO CONTAIN ONLY RDY (BIT #8 IN RHCS1 IS ZERO)
7353                                    ;*
7354                                    ;*
7355                                    ;;*****************************************************************
7356  022374  000004                    TST23:  SCOPE
7357  022376  012706  001000                    MOV     #STACK,SP       ;RESET STACK
7358  022402  012737  000023  004656            MOV     #23,@#TSTNM     ;SAVE TEST NUMBER
7359
7360  022410  004737  040312                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
7361                                                                    ;SETUP UNIT NUBER
7362                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
7363
7364                                    ;WRITE "1" INTO A16 IN RHCS1
7365  022414  012777  000400  161436            MOV     #A16,@RHCS1     ;MOVE 400 INTO RHCS1 .
7366
7367
7368
7369                                    ;CHECK THAT RHCS1 HAS A16!DVA!RDY
7370  022422  012737  004600  001124            MOV     #A16!DVA!RDY,$GDDAT     ;GET GOOD = 4600
7371
7372  022430  017737  161424  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
7373  022436  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7374                                                                    ;DATA WITH DATA READ FROM
7375                                                                    ;RHCS1
7376  022444  001401                            BEQ     65$             ;BRANCH IF GOOD
7377  022446  104054                            ERROR   54
7378                                                                    ;AFTER SETTING "CLR" BIT #5
7379                                                                    ;IN RHCS2 TO INIT THE RH
7380                                                                    ;A16 WAS WRITTEN INTO RHCS1
7381                                                                    ;RHCS1 WAS READ
7382                                                                    ;RHCS1 SHOULD HAVE A16!DVA!RDY
7383                                                                    ;=4600
7384                                                                    ;BUT CONTAINED WHAT IS
7385                                                                    ;GIVEN IN BAD RHCS1
7386  022450                            65$:
7387
7388
7389                                    ;CHECK THAT RHBAE HAS BIT0
7390  022450  012737  000001  001124            MOV     #BIT0,$GDDAT    ;GET GOOD = 1
7391
7392  022456  017737  161446  001126            MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
7393  022464  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7394                                                                    ;DATA WITH DATA READ FROM
7395                                                                    ;RHBAE
7396  022472  001401                            BEQ     67$             ;BRANCH IF GOOD
7397  022474  104055                            ERROR   55
7398                                                                    ;AFTER SETTING "CLR" BIT #5
7399                                                                    ;IN RHCS2 TO INIT THE RH
7400                                                                    ;A16 WAS WRITTEN INTO RHCS1
7401                                                                    ;RHCS1 WAS READ
7402                                                                    ;THEN RHBAE WAS READ
7403                                                                    ;RHBAE SHOULD HAVE BIT0
7404                                                                    ;=1
7405                                                                    ;BUT CONTAINED WHAT IS
7406                                                                    ;GIVEN IN BAD RHBAE
7407  022476                            67$:
```

```
7408
7409                                        ;NOW MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #0)
7410  022476  005077  161426                CLR     @RHBAE          ;WRITE ZERO INTO RHBAE
7411
7412
7413                                        ;CHECK THAT RHBAE HAS 0
7414  022502  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
7415
7416  022510  017737  161414  001126        MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
7417  022516  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7418                                                                ;DATA WITH DATA READ FROM
7419                                                                ;RHBAE
7420  022524  001401                        BEQ     69$             ;BRANCH IF GOOD
7421  022526  104056                        ERROR   56
7422                                                                ;AFTER SETTING "CLR" BIT #5
7423                                                                ;IN RHCS2 TO INIT THE RH
7424                                                                ;A16 WAS WRITTEN INTO RHCS1
7425                                                                ;RHCS1 WAS READ
7426                                                                ;RHBAE WAS READ
7427                                                                ;THEN ZERO WAS WRITTEN
7428                                                                ;INTO RHBAE
7429                                                                ;ON READING RHBAE
7430                                                                ;RHBAE SHOULD HAVE 0
7431                                                                ;BUT CONTAINED WHAT IS
7432                                                                ;GIVEN IN BAD RHBAE
7433  022530                            69$:
7434
7435
7436                                        ;CHECK THAT RHCS1 HAS DVA!RDY
7437  022530  012737  004200  001124        MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
7438
7439  022536  017737  161316  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
7440  022544  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7441                                                                ;DATA WITH DATA READ FROM
7442                                                                ;RHCS1
7443  022552  001401                        BEQ     71$             ;BRANCH IF GOOD
7444  022554  104057                        ERROR   57
7445                                                                ;AFTER SETTING "CLR" BIT #5
7446                                                                ;IN RHCS2 TO INIT THE RH
7447                                                                ;A16 WAS WRITTEN INTO RHCS1
7448                                                                ;RHCS1 WAS READ
7449                                                                ;RHBAE WAS READ
7450                                                                ;THEN ZERO WAS WRITTEN
7451                                                                ;INTO RHBAE
7452                                                                ;RHBAE WAS READ
7453                                                                ;ON READING RHCS1
7454                                                                ;RHCS1 SHOULD HAVE DVA!RDY
7455                                                                ;=4200
7456                                                                ;BUT CONTAINED WHAT IS
7457                                                                ;GIVEN IN BAD RHCS1
7458  022556                            71$:
7459
7460
7461                                        ;;*****************************************************************
7462                                        ;*TEST 24        TEST DUPLICATED A17 (RHCS1 BIT #9)
7463
```

```
                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                ;*      MOVE 400 (ONE INTO A17) IN RHCS1
                                ;*      READ RHCS1 TO CONTAIN 1200 (BIT #9 AND RDY)
                                ;*      READ RHBAE TO CONTAIN "2" (BIT #1 HIGH)
                                ;*
                                ;*
                                ;*      MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #1)
                                ;*      READ RHBAE TO CONTAIN 2
                                ;*      READ RHCS1 TO CONTAIN ONLY RDY (BIT #9 IN RHCS1 IS ZERO)
                                ;*
                                ;*
                                ;;*****************************************************************
022556  000004                  TST24:  SCOPE
022560  012706  001000                  MOV     #STACK,SP       ;RESET STACK
022564  012737  000024  004656          MOV     #24,@#TSTNM     ;SAVE TEST NUMBER

022572  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;WRITE "1" INTO A17 IN RHCS1
022576  012777  001000  161254          MOV     #A17,@RHCS1     ;MOVE 1200 INTO RHCS1


                                        ;CHECK THAT RHCS1 HAS A17!DVA!RDY
022604  012737  005200  001124          MOV     #A17!DVA!RDY,$GDDAT     ;GET GOOD = 4600

022612  017737  161242  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022620  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
022626  001401                          BEQ     65$             ;BRANCH IF GOOD
022630  104060                          ERROR   60

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;A17 WAS WRITTEN INTO RHCS1
                                                                ;RHCS1 WAS READ
                                                                ;RHCS1 SHOULD HAVE A17!DVA!RDY
                                                                ;=4600
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
022632                          65$:


                                        ;CHECK THAT RHBAE HAS BIT1
022632  012737  000002  001124          MOV     #BIT1,$GDDAT    ;GET GOOD = 2

022640  017737  161264  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
022646  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
022654  001401                          BEQ     67$             ;BRANCH IF GOOD
022656  104061                          ERROR   61

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
```

```
                                                        ;A17 WAS WRITTEN INTO RHCS1
                                                        ;RHCS1 WAS READ
                                                        ;THEN RHBAE WAS READ
                                                        ;RHBAE SHOULD HAVE BIT1
                                                        ;=2
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHBAE
022660                          67$:

                                        ;NOW MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #0)
022660  005077  161244                  CLR     @RHBAE          ;WRITE ZERO INTO RHBAE

                                        ;CHECK THAT RHBAE HAS 0
022664  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

022672  017737  161232  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
022700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
022706  001401                          BEQ     69$             ;BRANCH IF GOOD
022710  104062                          ERROR   62
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A17 WAS WRITTEN INTO RHCS1
                                                        ;RHCS1 WAS READ
                                                        ;RHBAE WAS READ
                                                        ;THEN ZERO WAS WRITTEN
                                                        ;INTO RHBAE
                                                        ;ON READING RHBAE
                                                        ;RHBAE SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHBAE
022712                          69$:


                                        ;CHECK THAT RHCS1 HAS DVA!RDY
022712  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

022720  017737  161134  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022726  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
022734  001401                          BEQ     71$             ;BRANCH IF GOOD
022736  104063                          ERROR   63

                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A17 WAS WRITTEN INTO RHCS1
                                                        ;RHCS1 WAS READ
                                                        ;RHBAE WAS READ
                                                        ;THEN ZERO WAS WRITTEN
                                                        ;INTO RHBAE
                                                        ;RHBAE WAS READ
                                                        ;ON READING RHCS1
                                                        ;RHCS1 SHOULD HAVE DVA!RDY
                                                        ;=4200
```

```
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS1
022740                          71$:


                                ;;**************************************************************
                                ;*TEST 25       TEST DUPLICATED IE (RHCS1 BIT #6)

                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                ;*      MOVE 100 (ONE INTO IE) IN RHCS1
                                ;*      READ RHCS1 TO CONTAIN 300 (BIT #6 AND RDY)
                                ;*      READ RHCS3 TO CONTAIN "100" (BIT #6 HIGH)
                                ;*
                                ;*
                                ;*      MOVE 0 INTO RHCS3 (WRITING 0 INTO RHCS3 BIT #6)
                                ;*      READ RHCS3 TO CONTAIN 100
                                ;*      READ RHCS1 TO CONTAIN ONLY RDY (BIT #6 IN RHCS1 IS ZERO)
                                ;*
                                ;*
                                ;;**************************************************************
022740  000004                  TST25:  SCOPE
022742  012706  001000                  MOV     #STACK,SP       ;RESET STACK
022746  012737  000025  004656          MOV     #25,@#TSTNM     ;SAVE TEST NUMBER

022754  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;WRITE "1" INTO IE IN RHCS1
022760  012777  000100  161072          MOV     #IE,@RHCS1      ;MOVE 100 INTO RHCS1


                                        ;CHECK THAT RHCS1 HAS IE!DVA!RDY
022766  012737  004300  001124          MOV     #IE!DVA!RDY,$GDDAT      ;GET GOOD = 4300

022774  017737  161060  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
023002  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
023010  001401                          BEQ     65$             ;BRANCH IF GOOD
023012  104064                          ERROR   64
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IE WAS WRITTEN INTO RHCS1
                                                                ;RHCS1 WAS READ
                                                                ;RHCS1 SHOULD HAVE IE!DVA!RDY
                                                                ;=4300
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
023014                          65$:


                                        ;CHECK THAT RHCS3 HAS IE
023014  012737  000100  001124          MOV     #IE,$GDDAT      ;GET GOOD = 1
```

```
 7632  023022  017737  161104  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 7633  023030  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 7634                                                                  ;DATA WITH DATA READ FROM
 7635                                                                  ;RHCS3
 7636  023036  001401                          BEQ     67$             ;BRANCH IF GOOD
 7637  023040  104065                          ERROR   65
 7638                                                                  ;AFTER SETTING "CLR" BIT #5
 7639                                                                  ;IN RHCS2 TO INIT THE RH
 7640                                                                  ;IE WAS WRITTEN INTO RHCS1
 7641                                                                  ;RHCS1 WAS READ
 7642                                                                  ;THEN RHCS3 WAS READ
 7643                                                                  ;RHCS3 SHOULD HAVE IE
 7644                                                                  ;=1
 7645                                                                  ;BUT CONTAINED WHAT IS
 7646                                                                  ;GIVEN IN BAD RHCS3
 7647  023042                          67$:
 7648
 7649                                          ;NOW MOVE 0 INTO RHCS3 (WRITING 0 INTO RHCS3 BIT #1)
 7650  023042  005077  161064                  CLR     @RHCS3          ;WRITE ZERO INTO RHCS3
 7651
 7652
 7653                                          ;CHECK THAT RHCS3 HAS 0
 7654  023046  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 7655
 7656  023054  017737  161052  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 7657  023062  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 7658                                                                  ;DATA WITH DATA READ FROM
 7659                                                                  ;RHCS3
 7660  023070  001401                          BEQ     69$             ;BRANCH IF GOOD
 7661  023072  104066                          ERROR   66
 7662                                                                  ;AFTER SETTING "CLR" BIT #5
 7663                                                                  ;IN RHCS2 TO INIT THE RH
 7664                                                                  ;IE WAS WRITTEN INTO RHCS1
 7665                                                                  ;RHCS1 WAS READ
 7666                                                                  ;RHCS3 WAS READ
 7667                                                                  ;THEN ZERO WAS WRITTEN
 7668                                                                  ;INTO RHCS3
 7669                                                                  ;ON READING RHCS3
 7670                                                                  ;RHCS3 SHOULD HAVE 0
 7671                                                                  ;BUT CONTAINED WHAT IS
 7672                                                                  ;GIVEN IN BAD RHCS3
 7673  023074                          69$:
 7674
 7675
 7676                                          ;CHECK THAT RHCS1 HAS DVA!RDY
 7677  023074  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
 7678
 7679  023102  017737  160752  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 7680  023110  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 7681                                                                  ;DATA WITH DATA READ FROM
 7682                                                                  ;RHCS1
 7683  023116  001401                          BEQ     71$             ;BRANCH IF GOOD
 7684  023120  104067                          ERROR   67
 7685                                                                  ;AFTER SETTING "CLR" BIT #5
 7686                                                                  ;IN RHCS2 TO INIT THE RH
 7687                                                                  ;IE WAS WRITTEN INTO RHCS1
```

```
                                                        ;RHCS1 WAS READ
                                                        ;RHCS3 WAS READ
                                                        ;THEN ZERO WAS WRITTEN
                                                        ;INTO RHCS3
                                                        ;RHCS3 WAS READ
                                                        ;ON READING RHCS1
                                                        ;RHCS1 SHOULD HAVE DVA!RDY
                                                        ;=4200
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS1
023122                          715$:


                                ;;*****************************************************************
                                ;*TEST 26       RHCS1 PROGRAM ERROR (PGE BIT #10)

                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                ;*      SET UP FOR A 10 WORD WRITE
                                ;*      SET "GO" (RHCS1 BIT #0) TWICE IN TWO SUCCESSIVE
                                ;*      INSTRUCTIONS
                                ;*      THIS SHOULD SET, SC AND TRE IN RHCS1
                                ;*      AND PGE SHOULD BE SET IN RHCS2
                                ;*      RHCS3, ARE CHECKED
                                ;*      THE NUMBER OF WORDS ARE LARGE ENOUGH SO THAT AFTER
                                ;*      ONE "BIS" INSTRUCTION TO SET "GO" IN RHCS1
                                ;*      THERE IS SUFFICIENT TIME TO GIVE ANOTHER "BIS" INSTRUCTION
                                ;*      TO SET "GO" BEFORE THE FIRST "GO" HAS TIME TO COMPLETE
                                ;*
                                ;;*****************************************************************
023122  000004                  TST26:  SCOPE
023124  012706  001000                  MOV     #STACK,SP       ;RESET STACK
023130  012737  000026  004656          MOV     #26,@#TSTNM     ;SAVE TEST NUMBER

023136  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;SET UP RH FOR A 10 WORD WRITE
023142  012777  177766  160712          MOV     #-10.,@RHWC     ;WORD COUNT REGISTER=10
023150  012777  004210  160706          MOV     #WRFROM,@RHBA   ;BUS ADDRESS REGISTER=WRITBUF
                                        ;WRITE 10 WORDS FROM "WRFROM" INTO
                                        ;WHAT EVER RH DEVICE IS AVAILABLE
023156  012737  177777  004644          MOV     #-1,NOGO        ;DO NOT GIVE GO IN COMMAND ROUTINE
023164  004077  161450                  JSR     R0,@COMND       ;GO TO DO COMMAND
023170  004154                          WRIDAT                  ;WRITE DATA
023172  052777  000001  160660          BIS     #GO,@RHCS1      ;SET GO
023200  052777  000001  160652          BIS     #GO,@RHCS1      ;SET GO WITHOUT TIME TO COMPLET LAST GO


                                        ;CHECK THAT RHCS1 HAS SC!DVA!TRE!61
023206  012737  144061  001124          MOV     #SC!DVA!TRE!61,$GDDAT   ;GET GOOD = 144000

023214  017737  160640  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
023222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
```

```
7744  023230  001401                          BEQ     65$             ;BRANCH IF GOOD
7745  023232  104070                          ERROR   70
7746                                                                  ;AFTER SETTING "CLR" BIT #5
7747                                                                  ;IN RHCS2 TO INIT THE RH
7748                                                                  ;AND GIVING TWO SUCCESSIVE
7749                                                                  ;"GO" WITHOUT GIVING TIME FOR
7750                                                                  ;THE FIRST TO COMPLETE
7751                                                                  ;RHCS1 SHOULD HAVE SC!DVA!TRE!61
7752                                                                  ;=144000
7753                                                                  ;BUT CONTAINED WHAT IS
7754                                                                  ;GIVEN IN BAD RHCS1
7755  023234                          65$:
7756                                          ;CHECK THAT RHCS2 HAS PGE!OR
7757  023234  012737  002200  001124          MOV     #PGE!OR,$GDDAT  ;GET GOOD = 2000
7758  023242  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7759
7760  023250  017737  160614  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7761  023256  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7762                                                                  ;DATA WITH DATA READ FROM
7763                                                                  ;RHCS2
7764  023264  001401                          BEQ     67$             ;BRANCH IF GOOD
7765  023266  104071                          ERROR   71
7766                                                                  ;AFTER SETTING "CLR" BIT #5
7767                                                                  ;IN RHCS2 TO INIT THE RH
7768                                                                  ;AND GIVING TWO SUCCESSIVE
7769                                                                  ;"GO" WITHOUT GIVING TIME FOR
7770                                                                  ;THE FIRST TO COMPLETE
7771                                                                  ;RHCS2 SHOULD HAVE PGE!OR
7772                                                                  ;=2000
7773                                                                  ;TOGETHER WITH UNIT NUMBER
7774                                                                  ;BUT CONTAINED WHAT IS
7775                                                                  ;GIVEN IN BAD RHCS2
7776  023270                          67$:
7777                                          ;CHECK THAT RHCS3 HAS DBL
7778  023270  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 0
7779
7780  023276  017737  160630  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7781  023304  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7782                                                                  ;DATA WITH DATA READ FROM
7783                                                                  ;RHCS3
7784  023312  001401                          BEQ     69$             ;BRANCH IF GOOD
7785  023314  104072                          ERROR   72
7786                                                                  ;AFTER SETTING "CLR" BIT #5
7787                                                                  ;IN RHCS2 TO INIT THE RH
7788                                                                  ;AND GIVING TWO SUCCESSIVE
7789                                                                  ;"GO" WITHOUT GIVING TIME FOR
7790                                                                  ;THE FIRST TO COMPLETE
7791                                                                  ;RHCS3 SHOULD HAVE DBL
7792                                                                  ;BUT CONTAINED WHAT IS
7793                                                                  ;GIVEN IN BAD RHCS3
7794  023316                          69$:
7795
7796
7797                                  ;;*****************************************************************
7798                                  ;*TEST 27       RHCS2 - BUS ADDRESS INHIBIT BIT #3
7799
```

```
                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #6)
                                        ;*      SET UP FOR A 2 WORD WRITE FROM A BUFFER TAGGED "WRFROM"
                                        ;*      SET BUS ADDRESS INHIBIT RHCS2 BIT #3 "BAI"
                                        ;*      AT THE END OF WRITE CHECK
                                        ;*      RHCS1 TO HAVE ONLY RDY AND COMMAND
                                        ;*      RHCS2 TO HAVE BAI
                                        ;*      RHCS3 TO HAVE 0
                                        ;*      RHBA TO HAVE ADDRESS OF "WRFROM"
                                        ;*      RHBAE AND RHWC TO HAVE ZEROS
                                        ;*
                                        ;*      NOW SET UP READ FOR THE SAME DATA WITH
                                        ;*      BAI SET TO READ INTO A BUFFER TAGGED "REINTO"
                                        ;*      AFTER READ CHECK
                                        ;*      RHCS1 TO HAVE ONLY RDY AND COMMAND
                                        ;*      RHCS2 TO HAVE BAI
                                        ;*      RHCS3 TO HAVE 0
                                        ;*      RHBA TO HAVE ADDRESS OF "REINTO"
                                        ;*      RHBAE AND RHWC TO HAVE ZEROS
                                        ;*      DATA IN REINTO BUFFER IS CHECKED WITH DATA IN
                                        ;*      WRFROM BUFFER
                                        ;*************************************************************
023316  000004                        ↑ST27:  SCOPE
023320  012706  001000                          MOV     #STACK,SP       ;RESET STACK
023324  012737  000027  004656                  MOV     #27,@#↑STNM     ;SAVE TEST NUMBER

023332  004737  040312                          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                                ;SET UP 10 WORD WRITE FROM BUFFER TAGGED "WRFROM"
                                                ;WITH BAI IN RHCS2
023336  012777  177766  160516                  MOV     #-10.,@RHWC     ;WORD COUNT REGISTER=10
023344  052777  000010  160516                  BIS     #BAI,@RHCS2     ;BUS ADDRESS INHIBIT IN RHCS2
023352  012777  004210  160504                  MOV     #WRFROM,@RHBA   ;BUS ADDRESS REGISTER=WRFROM
023360  004077  161254                          JSR     R0,@COMND       ;GO TO DO COMMAND
023364  004154                                  WRIDAT                  ;WRITE DATA


                                                ;CHECK THAT RHCS1 HAS DVA!RDY!60
023366  012737  004260  001124                  MOV     #DVA!RDY!60,$GDDAT      ;GET GOOD = 4252

023374  017737  160460  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
023402  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS1
023410  001401                                  BEQ     65$             ;BRANCH IF GOOD
023412  104076                                  ERROR   76

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A 2 WORD WRITE WAS DONE
                                                                        ;WITH BAI BIT IN RHCS2 SET
                                                                        ;AT END OF WRITE
                                                                        ;RHCS1 SHOULD HAVE DVA!RDY!60
                                                                        ;=4252
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS1
```

```
7856  023414                                  65$:
7857                                          ;CHECK THAT RHCS2 HAS BAI!IR
7858  023414  012737  000110  001124          MOV     #BAI!IR,$GDDAT  ;GET GOOD = 10
7859  023422  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7860
7861  023430  017737  160434  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7862  023436  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7863                                                                  ;DATA WITH DATA READ FROM
7864                                                                  ;RHCS2
7865  023444  001401                          BEQ     67$             ;BRANCH IF GOOD
7866  023446  104077                          ERROR   77
7867                                                                  ;AFTER SETTING "CLR" BIT #5
7868                                                                  ;IN RHCS2 TO INIT THE RH
7869                                                                  ;A 2 WORD WRITE WAS DONE
7870                                                                  ;WITH BAI BIT IN RHCS2 SET
7871                                                                  ;AT END OF WRITE
7872                                                                  ;RHCS2 SHOULD HAVE BAI!IR
7873                                                                  ;=10
7874                                                                  ;TOGETHER WITH UNIT NUMBER
7875                                                                  ;BUT CONTAINED WHAT IS
7876                                                                  ;GIVEN IN BAD RHCS2
7877  023450                                  67$:
7878                                          ;CHECK THAT RHCS3 HAS 0
7879  023450  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
7880
7881  023456  017737  160450  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7882  023464  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7883                                                                  ;DATA WITH DATA READ FROM
7884                                                                  ;RHCS3
7885  023472  001401                          BEQ     69$             ;BRANCH IF GOOD
7886  023474  104100                          ERROR   100
7887                                                                  ;AFTER SETTING "CLR" BIT #5
7888                                                                  ;IN RHCS2 TO INIT THE RH
7889                                                                  ;A 2 WORD WRITE WAS DONE
7890                                                                  ;WITH BAI BIT IN RHCS2 SET
7891                                                                  ;AT END OF WRITE
7892                                                                  ;RHCS3 SHOULD HAVE 0
7893                                                                  ;BUT CONTAINED WHAT IS
7894                                                                  ;GIVEN IN BAD RHCS3
7895  023476                                  69$:
7896                                          ;CHECK THAT RHBA HAS WRFROM
7897  023476  012737  004210  001124          MOV     #WRFROM,$GDDAT  ;GET GOOD = 0
7898
7899  023504  017737  160354  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
7900  023512  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7901                                                                  ;DATA WITH DATA READ FROM
7902                                                                  ;RHBA
7903  023520  001401                          BEQ     71$             ;BRANCH IF GOOD
7904  023522  104101                          ERROR   101
7905                                                                  ;AFTER SETTING "CLR" BIT #5
7906                                                                  ;IN RHCS2 TO INIT THE RH
7907                                                                  ;A 2 WORD WRITE WAS DONE
7908                                                                  ;WITH BAI BIT IN RHCS2 SET
7909                                                                  ;AT END OF WRITE
7910                                                                  ;RHBA SHOULD HAVE WRFROM
7911                                                                  ;BUT CONTAINED WHAT IS
```

```
                                                                        ;GIVEN IN BAD RHBA
023524                                  71$:
                                                ;CHECK THAT RHBAE HAS 0
023524  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0

023532  017737  160372  001126                MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
023540  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBAE
023546  001401                                BEQ     73$             ;BRANCH IF GOOD
023550  104102                                ERROR   102
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A 2 WORD WRITE WAS DONE
                                                                        ;WITH BAI BIT IN RHCS2 SET
                                                                        ;AT END OF WRITE
                                                                        ;RHBAE SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBAE
023552                                  73$:
                                                ;CHECK THAT RHWC HAS 0
023552  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0

023560  017737  160276  001126                MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
023566  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHWC
023574  001401                                BEQ     75$             ;BRANCH IF GOOD
023576  104103                                ERROR   103
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A 2 WORD WRITE WAS DONE
                                                                        ;WITH BAI BIT IN RHCS2 SET
                                                                        ;AT END OF WRITE
                                                                        ;RHWC SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHWC
023600                                  75$:


                                        ;;*****************************************************************
                                        ;*TEST 30       RHSC2 MDPE BIT #8 AND RHCS3 IPCKO BIT #0
                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        ;*      SET IPCKO (RHCS3 BIT #0)
                                        ;*      MOVE ALL ZEROS INTO RHDB ONCE
                                        ;*      THIS SHOULD SET TRE SC IN RHCS1
                                        ;*      READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
                                        ;*      CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15)
                                        ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ;*      ALL ERRORS ARE CLEARED
                                        ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                        ;;*****************************************************************
023600  000004                          TST30:  SCOPE
023602  012706  001000                        MOV     #STACK,SP       ;RESET STACK
```

```
7968  023606  012737  000030  004656          MOV     #30,@#TSTNM     ;SAVE TEST NUMBER
7969
7970  023614  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
7971                                                                  ;SETUP UNIT NUBER
7972                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
7973  023620  012737  000000  004662          MOV     #0,IP           ;IPCK BIT NO. FOR PRINTOUT
7974
7975  023626  052777  000001  160276          BIS     #IPCKO,@RHCS3 ;SET IPCKO IN RHCS3-BIT #0
7976  023634  005037  004210                  CLR     WRFROM          ;WRITE ZERO INTO LOCATION
7977  023640  013777  004210  160234          MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
7978
7979                                          ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
7980  023646  012737  144200  001124          MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200
7981
7982  023654  017737  160200  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
7983  023662  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7984                                                                  ;DATA WITH DATA READ FROM
7985                                                                  ;RHCS1
7986  023670  001401                          BEQ     65$             ;BRANCH IF GOOD
7987  023672  104140                          ERROR   140
7988                                                                  ;AFTER SETTING "CLR" BIT #5
7989                                                                  ;IN RHCS2 TO INIT THE RH
7990                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
7991                                                                  ;ZEROS WERE MOVED INTO RHDB
7992                                                                  ;RHCS1 SHOULD HAVE SC!TRE!DVA!RDY
7993                                                                  ;=144200
7994                                                                  ;BUT CONTAINED WHAT IS
7995                                                                  ;GIVEN IN BAD RHCS1
7996  023674                          65$:
7997  023674  017737  160202  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE
7998
7999
8000                                          ;CHECK THAT RHCS3 HAS 1
8001  023702  012737  000001  001124          MOV     #1,$GDDAT       ;GET GOOD = 1
8002
8003  023710  017737  160216  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8004  023716  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8005                                                                  ;DATA WITH DATA READ FROM
8006                                                                  ;RHCS3
8007  023724  001401                          BEQ     67$             ;BRANCH IF GOOD
8008  023726  104104                          ERROR   104
8009                                                                  ;AFTER SETTING "CLR" BIT #5
8010                                                                  ;IN RHCS2 TO INIT THE RH
8011                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8012                                                                  ;ZEROS WERE MOVED INTO
8013                                                                  ;RHDB
8014                                                                  ;RHDB WAS READ THEN
8015                                                                  ;RHCS3 SHOULD HAVE 1
8016                                                                  ;=1
8017                                                                  ;BUT CONTAINED WHAT IS
8018                                                                  ;GIVEN IN BAD RHCS3
8019  023730                          67$:
8020
8021                                          ;CHECK THAT RHCS2 HAS MDPE!IR
8022  023730  012737  000500  001124          MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
8023  023736  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
8024
8025  023744  017737  160120  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8026  023752  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8027                                                                  ;DATA WITH DATA READ FROM
8028                                                                  ;RHCS2
8029  023760  001401                          BEQ     69$             ;BRANCH IF GOOD
8030  023762  104105                          ERROR   105
8031                                                                  ;AFTER SETTING "CLR" BIT #5
8032                                                                  ;IN RHCS2 TO INIT THE RH
8033                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8034                                                                  ;ZEROS WERE MOVED INTO
8035                                                                  ;RHDB
8036                                                                  ;RHDB WAS READ THEN
8037                                                                  ;RHCS2 SHOULD HAVE MDPE!IR
8038                                                                  ;=500
8039                                                                  ;TOGETHER WITH UNIT NUMBER
8040                                                                  ;BUT CONTAINED WHAT IS
8041                                                                  ;GIVEN IN BAD RHCS2
8042  023764                          69$:
8043
8044
8045  023764  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
8046                                                                  ;SETUP UNIT NUBER
8047                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8048
8049
8050                                  ;CHECK THAT RHCS1 HAS RDY!DVA
8051  023770  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
8052
8053  023776  017737  160056  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8054  024004  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8055                                                                  ;DATA WITH DATA READ FROM
8056                                                                  ;RHCS1
8057  024012  001401                          BEQ     71$             ;BRANCH IF GOOD
8058  024014  104106                          ERROR   106
8059                                                                  ;AFTER SETTING "CLR" BIT #5
8060                                                                  ;IN RHCS2 TO INIT THE RH
8061                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8062                                                                  ;ZEROS WERE MOVED INTO
8063                                                                  ;RHDB
8064                                                                  ;RHDB WAS READ THEN
8065                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8066                                                                  ;WAS GIVEN THIS SHOULD
8067                                                                  ;CLEAR ALL ERRORS
8068                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
8069                                                                  ;=4200
8070                                                                  ;BUT CONTAINED WHAT IS
8071                                                                  ;GIVEN IN BAD RHCS1
8072  024016                          71$:
8073                                  ;CHECK THAT RHCS2 HAS IR
8074  024016  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
8075  024024  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8076
8077  024032  017737  160032  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8078  024040  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8079                                                                  ;DATA WITH DATA READ FROM
```

```
8080
8081  024046  001401                          BEQ     73$             ;RHCS2
8082  024050  104107                          ERROR   107             ;BRANCH IF GOOD
8083                                                                  ;AFTER SETTING "CLR" BIT #5
8084                                                                  ;IN RHCS2 TO INIT THE RH
8085                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8086                                                                  ;ZEROS WERE MOVED INTO
8087                                                                  ;RHDB
8088                                                                  ;RHDB WAS READ THEN
8089                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8090                                                                  ;WAS GIVEN THIS SHOULD
8091                                                                  ;CLEAR ALL ERRORS
8092                                                                  ;RHCS2 SHOULD HAVE IR
8093                                                                  ;=100
8094                                                                  ;TOGETHER WITH UNIT NUMBER
8095                                                                  ;BUT CONTAINED WHAT IS
8096                                                                  ;GIVEN IN BAD RHCS2
8097  024052                          73$:
8098                                          ;CHECK THAT RHCS3 HAS 0
8099  024052  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8100
8101  024060  017737  160046  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8102  024066  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8103                                                                  ;DATA WITH DATA READ FROM
8104                                                                  ;RHCS3
8105  024074  001401                          BEQ     75$             ;BRANCH IF GOOD
8106  024076  104110                          ERROR   110
8107                                                                  ;AFTER SETTING "CLR" BIT #5
8108                                                                  ;IN RHCS2 TO INIT THE RH
8109                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8110                                                                  ;ZEROS WERE MOVED INTO
8111                                                                  ;RHDB
8112                                                                  ;RHDB WAS READ THEN
8113                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8114                                                                  ;WAS GIVEN THIS SHOULD
8115                                                                  ;CLEAR ALL ERRORS
8116                                                                  ;RHCS3 SHOULD HAVE 0
8117                                                                  ;BUT CONTAINED WHAT IS
8118                                                                  ;GIVEN IN BAD RHCS3
8119  024100                          75$:
8120                                          ;CHECK THAT RHBA HAS 0
8121  024100  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8122
8123  024106  017737  157752  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
8124  024114  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8125                                                                  ;DATA WITH DATA READ FROM
8126                                                                  ;RHBA
8127  024122  001401                          BEQ     77$             ;BRANCH IF GOOD
8128  024124  104111                          ERROR   111
8129                                                                  ;AFTER SETTING "CLR" BIT #5
8130                                                                  ;IN RHCS2 TO INIT THE RH
8131                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8132                                                                  ;ZEROS WERE MOVED INTO
8133                                                                  ;RHDB
8134                                                                  ;RHDB WAS READ THEN
8135                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
```

```
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHBA SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBA
024126                                  77$:
                                                ;CHECK THAT RHBAE HAS 0
024126  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

024134  017737  157770  001126                  MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
024142  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBAE
024150  001401                                  BEQ     79$             ;BRANCH IF GOOD
024152  104112                                  ERROR   112

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;IPCKO BIT #0 RHCS3 WAS SET
                                                                        ;ZEROS WERE MOVED INTO
                                                                        ;RHDB
                                                                        ;RHDB WAS READ THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHBAE SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBAE
024154                                  79$:
                                                ;CHECK THAT RHWC HAS 0
024154  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

024162  017737  157674  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
024170  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHWC
024176  001401                                  BEQ     81$             ;BRANCH IF GOOD
024200  104113                                  ERROR   113

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;IPCKO BIT #0 RHCS3 WAS SET
                                                                        ;ZEROS WERE MOVED INTO
                                                                        ;RHDB
                                                                        ;RHDB WAS READ THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHWC SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHWC
024202                                  81$:


                                        ;;*****************************************************************
                                        ;*TEST 31       RHSC2 MDPE BIT #8 AND RHCS3 IPCK1 BIT #1
                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
```

```
                                ;*      SET IPCK1 (RHCS3 BIT #1)
                                ;*      MOVE ALL ZEROS INTO RHDB ONCE
                                ;*      THIS SHOULD SET TRE SC IN RHCS1
                                ;*      READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
                                ;*      CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15)
                                ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                                ;*      ALL ERRORS ARE CLEARED
                                ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                ;;*************************************************************
024202  000004                  TST31:  SCOPE
024204  012706  001000                  MOV     #STACK,SP       ;RESET STACK
024210  012737  000031  004656          MOV     #31,@#TSTNM     ;SAVE TEST NUMBER

024216  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
024222  012737  000001  004662          MOV     #1,IP           ;IPCK BIT NO. FOR PRINTOUT

024230  052777  000002  157674          BIS     #IPCK1,@RHCS3 ;SET IPCK1 IN RHCS3-BIT #1
024236  005037  004210                  CLR     WRFROM          ;WRITE ZERO INTO LOCATION
024242  013777  004210  157632          MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE

                                        ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
024250  012737  144200  001124          MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200

024256  017737  157576  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
024264  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
024272  001401                          BEQ     65$             ;BRANCH IF GOOD
024274  104140                          ERROR   140

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK1 BIT #1 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO RHDB
                                                                ;RHCS1 SHOULD HAVE SC!TRE!DVA!RDY
                                                                ;=144200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
024276                  65$:
024276  017737  157600  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE


                                        ;CHECK THAT RHCS3 HAS 2
024304  012737  000002  001124          MOV     #2,$GDDAT       ;GET GOOD = 2

024312  017737  157614  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
024320  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
024326  001401                          BEQ     67$             ;BRANCH IF GOOD
024330  104104                          ERROR   104

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK1 BIT #1 RHCS3 WAS SET
```

```
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;RHCS3 SHOULD HAVE 2
                                                                ;=2
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
024332                          67$:

                                        ;CHECK THAT RHCS2 HAS MDPE!IR
024332  012737  000500  001124        MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
024340  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

024346  017737  157516  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
024354  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
024362  001401                        BEQ     69$             ;BRANCH IF GOOD
024364  104105                        ERROR   105

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK1 BIT #1 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;RHCS2 SHOULD HAVE MDPE!IR
                                                                ;=500
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
024366                          69$:

024366  004737  040312                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS1 HAS RDY!DVA
024372  012737  004200  001124        MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200

024400  017737  157454  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
024406  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
024414  001401                        BEQ     71$             ;BRANCH IF GOOD
024416  104106                        ERROR   106

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK1 BIT #1 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
```

```
                                                                ;RHCS1 SHOULD HAVE RDY!DVA
                                                                ;=4200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
024420                                  71$:
                                                ;CHECK THAT RHCS2 HAS IR
024420  012737  000100  001124            MOV     #IR,$GDDAT    ;GET GOOD = 100
024426  053737  005010  001124            BIS     UNIT,$GDDAT   ;INCLUDE UNIT NUMBER

024434  017737  157430  001126            MOV     @RHCS2,$BDDAT ;READ RHCS2 FOR COMPARISON
024442  023737  001124  001126            CMP     $GDDAT,$BDDAT ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
024450  001401                            BEQ     73$           ;BRANCH IF GOOD
024452  104107                            ERROR   107
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK1 BIT #1 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
024454                                  73$:
                                                ;CHECK THAT RHCS3 HAS 0
024454  012737  000000  001124            MOV     #0,$GDDAT     ;GET GOOD = 0

024462  017737  157444  001126            MOV     @RHCS3,$BDDAT ;READ RHCS3 FOR COMPARISON
024470  023737  001124  001126            CMP     $GDDAT,$BDDAT ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
024476  001401                            BEQ     75$           ;BRANCH IF GOOD
024500  104110                            ERROR   110
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK1 BIT #1 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHCS3 SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
024502                                  75$:
                                                ;CHECK THAT RHBA HAS 0
024502  012737  000000  001124            MOV     #0,$GDDAT     ;GET GOOD = 0

024510  017737  157350  001126            MOV     @RHBA,$BDDAT  ;READ RHBA FOR COMPARISON
```

```
8360  024516  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8361                                                                    ;DATA WITH DATA READ FROM
8362                                                                    ;RHBA
8363  024524  001401                          BEQ     77$             ;BRANCH IF GOOD
8364  024526  104111                          ERROR   111
8365                                                                    ;AFTER SETTING "CLR" BIT #5
8366                                                                    ;IN RHCS2 TO INIT THE RH
8367                                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8368                                                                    ;ZEROS WERE MOVED INTO
8369                                                                    ;RHDB
8370                                                                    ;RHDB WAS READ THEN
8371                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8372                                                                    ;WAS GIVEN THIS SHOULD
8373                                                                    ;CLEAR ALL ERRORS
8374                                                                    ;RHBA SHOULD HAVE 0
8375                                                                    ;BUT CONTAINED WHAT IS
8376                                                                    ;GIVEN IN BAD RHBA
8377  024530                          77$:
8378                                          ;CHECK THAT RHBAE HAS 0
8379  024530  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8380
8381  024536  017737  157366  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
8382  024544  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8383                                                                    ;DATA WITH DATA READ FROM
8384                                                                    ;RHBAE
8385  024552  001401                          BEQ     79$             ;BRANCH IF GOOD
8386  024554  104112                          ERROR   112
8387                                                                    ;AFTER SETTING "CLR" BIT #5
8388                                                                    ;IN RHCS2 TO INIT THE RH
8389                                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8390                                                                    ;ZEROS WERE MOVED INTO
8391                                                                    ;RHDB
8392                                                                    ;RHDB WAS READ THEN
8393                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8394                                                                    ;WAS GIVEN THIS SHOULD
8395                                                                    ;CLEAR ALL ERRORS
8396                                                                    ;RHBAE SHOULD HAVE 0
8397                                                                    ;BUT CONTAINED WHAT IS
8398                                                                    ;GIVEN IN BAD RHBAE
8399  024556                          79$:
8400                                          ;CHECK THAT RHWC HAS 0
8401  024556  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8402
8403  024564  017737  157272  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
8404  024572  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8405                                                                    ;DATA WITH DATA READ FROM
8406                                                                    ;RHWC
8407  024600  001401                          BEQ     81$             ;BRANCH IF GOOD
8408  024602  104113                          ERROR   113
8409                                                                    ;AFTER SETTING "CLR" BIT #5
8410                                                                    ;IN RHCS2 TO INIT THE RH
8411                                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8412                                                                    ;ZEROS WERE MOVED INTO
8413                                                                    ;RHDB
8414                                                                    ;RHDB WAS READ THEN
8415                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
```

```
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
024604                                  81$:


                                        ;;**************************************************************
                                        ;*TEST 32      RHSC2 MDPE BIT #8 AND RHCS3 IPCK2 BIT #2
                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        ;*      SET IPCK2 (RHCS3 BIT #2)
                                        ;*      MOVE ALL ZEROS INTO RHDB TWICE
                                        ;*      THIS SHOULD NOT SET TRE SC IN RHCS1
                                        ;*      READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
                                        ;*      CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15)
                                        ;*      READ RHDB A SECOND TIME. CHECK RHCS1, RHCS2, RHCS3
                                        ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ;*      ALL ERRORS ARE CLEARED
                                        ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                        ;;**************************************************************
024604  000004                          TST32:  SCOPE
024606  012706  001000                          MOV     #STACK,SP       ;RESET STACK
024612  012737  000032  004656                  MOV     #32,@#TSTNM     ;SAVE TEST NUMBER

024620  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
024624  012737  000002  004662          MOV     #2,IP           ;IPCK BIT NO. FOR PRINTOUT

024632  052777  000004  157272          BIS     #IPCK2,@RHCS3 ;SET IPCK2 IN RHCS3-BIT #2
024640  005037  004210                  CLR     WRFROM          ;WRITE ZERO INTO LOCATION
024644  013777  004210  157230          MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
024652  013777  004210  157222          MOV     WRFROM,@RHDB    ;WRITE ZERO SECOND TIME

                                        ;CHECK THAT RHCS1 HAS DVA!RDY
024660  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

024666  017737  157166  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
024674  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
024702  001401                          BEQ     65$             ;BRANCH IF GOOD
024704  104140                          ERROR   140

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK2 BIT #2 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO RHDB
                                                                ;TWICE
                                                                ;RHCS1 SHOULD HAVE DVA!RDY
                                                                ;=4200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
024706                                  65$:
```

```
8472  024706  017737  157170  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE
8473
8474
8475                                          ;CHECK THAT RHCS3 HAS 4
8476  024714  012737  000004  001124          MOV     #4,$GDDAT       ;GET GOOD = 4
8477
8478  024722  017737  157204  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8479  024730  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8480                                                                  ;DATA WITH DATA READ FROM
8481                                                                  ;RHCS3
8482  024736  001401                          BEQ     67$             ;BRANCH IF GOOD
8483  024740  104104                          ERROR   104
8484                                                                  ;AFTER SETTING "CLR" BIT #5
8485                                                                  ;IN RHCS2 TO INIT THE RH
8486                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
8487                                                                  ;ZEROS WERE MOVED INTO
8488                                                                  ;RHDB
8489                                                                  ;RHDB WAS READ THEN
8490                                                                  ;RHCS3 SHOULD HAVE 4
8491                                                                  ;=4
8492                                                                  ;BUT CONTAINED WHAT IS
8493                                                                  ;GIVEN IN BAD RHCS3
8494  024742                          67$:
8495
8496                                          ;CHECK THAT RHCS2 HAS MDPE!OR!IR
8497  024742  012737  000700  001124          MOV     #MDPE!OR!IR,$GDDAT      ;GET GOOD = 700
8498  024750  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8499
8500  024756  017737  157106  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8501  024764  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8502                                                                  ;DATA WITH DATA READ FROM
8503                                                                  ;RHCS2
8504  024772  001401                          BEQ     69$             ;BRANCH IF GOOD
8505  024774  104105                          ERROR   105
8506                                                                  ;AFTER SETTING "CLR" BIT #5
8507                                                                  ;IN RHCS2 TO INIT THE RH
8508                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
8509                                                                  ;ZEROS WERE MOVED INTO
8510                                                                  ;RHDB
8511                                                                  ;RHDB WAS READ THEN
8512                                                                  ;RHCS2 SHOULD HAVE MDPE!OR!IR
8513                                                                  ;=700
8514                                                                  ;TOGETHER WITH UNIT NUMBER
8515                                                                  ;BUT CONTAINED WHAT IS
8516                                                                  ;GIVEN IN BAD RHCS2
8517  024776                          69$:
8518
8519  024776  017737  157100  001202          MOV     @RHDB,$TMP2     ;READ RHDB SECOND TIME
8520                                          ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
8521  025004  012737  144200  001124          MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200
8522
8523  025012  017737  157042  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8524  025020  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8525                                                                  ;DATA WITH DATA READ FROM
8526                                                                  ;RHCS1
8527  025026  001401                          BEQ     71$             ;BRANCH IF GOOD
```

```
8528  025030  104140                          ERROR   140
8529                                                      ;AFTER SETTING "CLR" BIT #5
8530                                                      ;IN RHCS2 TO INIT THE RH
8531                                                      ;IPCK2 BIT #2 RHCS3 WAS SET
8532                                                      ;ZEROS WERE MOVED INTO
8533                                                      ;RHDB
8534                                                      ;RHDB WAS READ THEN
8535                                                      ;RHCS1 SHOULD HAVE SC!TRE!DVA!RDY
8536                                                      ;=144200
8537                                                      ;BUT CONTAINED WHAT IS
8538                                                      ;GIVEN IN BAD RHCS1
8539  025032                          71$:
8540                                      ;CHECK THAT RHCS2 HAS MDPE!IR
8541  025032  012737  000500  001124      MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
8542  025040  053737  005010  001124      BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8543
8544  025046  017737  157016  001126      MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8545  025054  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8546                                                      ;DATA WITH DATA READ FROM
8547                                                      ;RHCS2
8548  025062  001401                      BEQ     73$             ;BRANCH IF GOOD
8549  025064  104105                      ERROR   105
8550                                                      ;AFTER SETTING "CLR" BIT #5
8551                                                      ;IN RHCS2 TO INIT THE RH
8552                                                      ;IPCK2 BIT #2 RHCS3 WAS SET
8553                                                      ;ZEROS WERE MOVED INTO
8554                                                      ;RHDB
8555                                                      ;RHDB WAS READ THEN
8556                                                      ;RHCS2 SHOULD HAVE MDPE!IR
8557                                                      ;=500
8558                                                      ;TOGETHER WITH UNIT NUMBER
8559                                                      ;BUT CONTAINED WHAT IS
8560                                                      ;GIVEN IN BAD RHCS2
8561  025066                          73$:
8562                                      ;CHECK THAT RHCS3 HAS 4
8563  025066  012737  000004  001124      MOV     #4,$GDDAT       ;GET GOOD = 4
8564
8565  025074  017737  157032  001126      MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8566  025102  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8567                                                      ;DATA WITH DATA READ FROM
8568                                                      ;RHCS3
8569  025110  001401                      BEQ     75$             ;BRANCH IF GOOD
8570  025112  104104                      ERROR   104
8571                                                      ;AFTER SETTING "CLR" BIT #5
8572                                                      ;IN RHCS2 TO INIT THE RH
8573                                                      ;IPCK2 BIT #2 RHCS3 WAS SET
8574                                                      ;ZEROS WERE MOVED INTO
8575                                                      ;RHDB
8576                                                      ;RHDB WAS READ THEN
8577                                                      ;RHCS3 SHOULD HAVE 4
8578                                                      ;=4
8579                                                      ;BUT CONTAINED WHAT IS
8580                                                      ;GIVEN IN BAD RHCS3
8581  025114                          75$:
8582
8583  025114  004737  040312              JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
```

```
8584                                                        ;SETUP UNIT NUBER
8585                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8586
8587
8588                                    ;CHECK THAT RHCS1 HAS RDY!DVA
8589  025120  012737  004200  001124    MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
8590
8591  025126  017737  156726  001126    MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8592  025134  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8593                                                        ;DATA WITH DATA READ FROM
8594                                                        ;RHCS1
8595  025142  001401                    BEQ     77$             ;BRANCH IF GOOD
8596  025144  104106                    ERROR   106
8597                                                        ;AFTER SETTING "CLR" BIT #5
8598                                                        ;IN RHCS2 TO INIT THE RH
8599                                                        ;IPCK2 BIT #2 RHCS3 WAS SET
8600                                                        ;ZEROS WERE MOVED INTO
8601                                                        ;RHDB
8602                                                        ;RHDB WAS READ THEN
8603                                                        ;AN RH CLEAR (RHCS2 BIT #5)
8604                                                        ;WAS GIVEN THIS SHOULD
8605                                                        ;CLEAR ALL ERRORS
8606                                                        ;RHCS1 SHOULD HAVE RDY!DVA
8607                                                        ;=4200
8608                                                        ;BUT CONTAINED WHAT IS
8609                                                        ;GIVEN IN BAD RHCS1
8610  025146                        77$:
8611                                    ;CHECK THAT RHCS2 HAS IR
8612  025146  012737  000100  001124    MOV     #IR,$GDDAT      ;GET GOOD = 100
8613  025154  053737  005010  001124    BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8614
8615  025162  017737  156702  001126    MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8616  025170  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8617                                                        ;DATA WITH DATA READ FROM
8618                                                        ;RHCS2
8619  025176  001401                    BEQ     79$             ;BRANCH IF GOOD
8620  025200  104107                    ERROR   107
8621                                                        ;AFTER SETTING "CLR" BIT #5
8622                                                        ;IN RHCS2 TO INIT THE RH
8623                                                        ;IPCK2 BIT #2 RHCS3 WAS SET
8624                                                        ;ZEROS WERE MOVED INTO
8625                                                        ;RHDB
8626                                                        ;RHDB WAS READ THEN
8627                                                        ;AN RH CLEAR (RHCS2 BIT #5)
8628                                                        ;WAS GIVEN THIS SHOULD
8629                                                        ;CLEAR ALL ERRORS
8630                                                        ;RHCS2 SHOULD HAVE IR
8631                                                        ;=100
8632                                                        ;TOGETHER WITH UNIT NUMBER
8633                                                        ;BUT CONTAINED WHAT IS
8634                                                        ;GIVEN IN BAD RHCS2
8635  025202                        79$:
8636                                    ;CHECK THAT RHCS3 HAS 0
8637  025202  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD = 0
8638
8639  025210  017737  156716  001126    MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
```

```
 8640  025216  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8641                                                                  ;DATA WITH DATA READ FROM ——
 8642                                                                  ;RHCS3
 8643  025224  001401                          BEQ     81$             ;BRANCH IF GOOD
 8644  025226  104110                          ERROR   110
 8645                                                                  ;AFTER SETTING "CLR" BIT #5
 8646                                                                  ;IN RHCS2 TO INIT THE RH
 8647                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
 8648                                                                  ;ZEROS WERE MOVED INTO
 8649                                                                  ;RHDB
 8650                                                                  ;RHDB WAS READ THEN
 8651                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 8652                                                                  ;WAS GIVEN THIS SHOULD
 8653                                                                  ;CLEAR ALL ERRORS
 8654                                                                  ;RHCS3 SHOULD HAVE 0
 8655                                                                  ;BUT CONTAINED WHAT IS
 8656                                                                  ;GIVEN IN BAD RHCS3
 8657  025230                          81$:
 8658                                          ;CHECK THAT RHBA HAS 0
 8659  025230  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 8660
 8661  025236  017737  156622  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
 8662  025244  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8663                                                                  ;DATA WITH DATA READ FROM
 8664                                                                  ;RHBA
 8665  025252  001401                          BEQ     83$             ;BRANCH IF GOOD
 8666  025254  104111                          ERROR   111
 8667                                                                  ;AFTER SETTING "CLR" BIT #5
 8668                                                                  ;IN RHCS2 TO INIT THE RH
 8669                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
 8670                                                                  ;ZEROS WERE MOVED INTO
 8671                                                                  ;RHDB
 8672                                                                  ;RHDB WAS READ THEN
 8673                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 8674                                                                  ;WAS GIVEN THIS SHOULD
 8675                                                                  ;CLEAR ALL ERRORS
 8676                                                                  ;RHBA SHOULD HAVE 0
 8677                                                                  ;BUT CONTAINED WHAT IS
 8678                                                                  ;GIVEN IN BAD RHBA
 8679  025256                          83$:
 8680                                          ;CHECK THAT RHBAE HAS 0
 8681  025256  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 8682
 8683  025264  017737  156640  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
 8684  025272  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8685                                                                  ;DATA WITH DATA READ FROM
 8686                                                                  ;RHBAE
 8687  025300  001401                          BEQ     85$             ;BRANCH IF GOOD
 8688  025302  104112                          ERROR   112
 8689                                                                  ;AFTER SETTING "CLR" BIT #5
 8690                                                                  ;IN RHCS2 TO INIT THE RH
 8691                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
 8692                                                                  ;ZEROS WERE MOVED INTO
 8693                                                                  ;RHDB
 8694                                                                  ;RHDB WAS READ THEN
 8695                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
```

```
8696                                                    ;WAS GIVEN THIS SHOULD
8697                                                    ;CLEAR ALL ERRORS
8698                                                    ;RHBAE SHOULD HAVE 0
8699                                                    ;BUT CONTAINED WHAT IS
8700                                                    ;GIVEN IN BAD RHBAE
8701  025304                            85$:
8702                                          ;CHECK THAT RHWC HAS 0
8703  025304  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
8704
8705  025312  017737  156544  001126        MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
8706  025320  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8707                                                    ;DATA WITH DATA READ FROM
8708                                                    ;RHWC
8709  025326  001401                        BEQ     87$             ;BRANCH IF GOOD
8710  025330  104113                        ERROR   113
8711                                                    ;AFTER SETTING "CLR" BIT #5
8712                                                    ;IN RHCS2 TO INIT THE RH
8713                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
8714                                                    ;ZEROS WERE MOVED INTO
8715                                                    ;RHDB
8716                                                    ;RHDB WAS READ THEN
8717                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8718                                                    ;WAS GIVEN THIS SHOULD
8719                                                    ;CLEAR ALL ERRORS
8720                                                    ;RHWC SHOULD HAVE 0
8721                                                    ;BUT CONTAINED WHAT IS
8722                                                    ;GIVEN IN BAD RHWC
8723  025332                            87$:
8724
8725
8726
8727                                    ;;**************************************************************
8728                                    ;*TEST 33       RHSC2 MDPE BIT #8 AND RHCS3 IPCK3 BIT #3
8729                                    ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
8730                                    ;*      SET IPCK3 (RHCS3 BIT #3)
8731                                    ;*      MOVE ALL ZEROS INTO RHDB TWICE
8732                                    ;*      THIS SHOULD NOT SET TRE SC IN RHCS1
8733                                    ;*      READ RHDB ONCE THIS SHOULD SET MDPE (RHCS2 BIT #8)
8734                                    ;*      CHECK RHCS1 FOR RDY (BIT #7), TRE (BIT #14), SC (BIT #15)
8735                                    ;*      READ RHDB A SECOND TIME. CHECK RHCS1, RHCS2, RHCS3
8736                                    ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
8737                                    ;*      ALL ERRORS ARE CLEARED
8738                                    ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC
8739
8740                                    ;;**************************************************************
8741  025332  000004                    ↑ST33:  SCOPE
8742  025334  012706  001000                    MOV     #STACK,SP       ;RESET STACK
8743  025340  012737  000033  004656            MOV     #33,@#↑STNM     ;SAVE TEST NUMBER
8744
8745  025346  004737  040312                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
8746                                                                    ;SETUP UNIT NUBER
8747                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8748  025352  012737  000003  004662            MOV     #3,IP           ;IPCK BIT NO. FOR PRINTOUT
8749
8750  025360  052777  000010  156544            BIS     #IPCK3,@RHCS3 ;SET IPCK3 IN RHCS3-BIT #3
8751  025366  005037  004210                    CLR     WRFROM          ;WRITE ZERO INTO LOCATION
```

```
8752  025372  013777  004210  156502          MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
8753  025400  013777  004210  156474          MOV     WRFROM,@RHDB    ;WRITE ZERO SECOND TIME
8754
8755                                          ;CHECK THAT RHCS1 HAS DVA!RDY
8756  025406  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
8757
8758  025414  017737  156440  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8759  025422  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8760                                                                  ;DATA WITH DATA READ FROM
8761                                                                  ;RHCS1
8762  025430  001401                          BEQ     65$             ;BRANCH IF GOOD
8763  025432  104140                          ERROR   140
8764                                                                  ;AFTER SETTING "CLR" BIT #5
8765                                                                  ;IN RHCS2 TO INIT THE RH
8766                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8767                                                                  ;ZEROS WERE MOVED INTO RHDB
8768                                                                  ;TWICE
8769                                                                  ;RHCS1 SHOULD HAVE DVA!RDY
8770                                                                  ;=4200
8771                                                                  ;BUT CONTAINED WHAT IS
8772                                                                  ;GIVEN IN BAD RHCS1
8773  025434                          65$:
8774  025434  017737  156442  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE
8775
8776
8777                                          ;CHECK THAT RHCS3 HAS 10
8778  025442  012737  000010  001124          MOV     #10,$GDDAT      ;GET GOOD = 10
8779
8780  025450  017737  156456  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8781  025456  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8782                                                                  ;DATA WITH DATA READ FROM
8783                                                                  ;RHCS3
8784  025464  001401                          BEQ     67$             ;BRANCH IF GOOD
8785  025466  104104                          ERROR   104
8786                                                                  ;AFTER SETTING "CLR" BIT #5
8787                                                                  ;IN RHCS2 TO INIT THE RH
8788                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8789                                                                  ;ZEROS WERE MOVED INTO
8790                                                                  ;RHDB
8791                                                                  ;RHDB WAS READ THEN
8792                                                                  ;RHCS3 SHOULD HAVE 10
8793                                                                  ;=10
8794                                                                  ;BUT CONTAINED WHAT IS
8795                                                                  ;GIVEN IN BAD RHCS3
8796  025470                          67$:
8797
8798                                          ;CHECK THAT RHCS2 HAS MDPE!OR!IR
8799  025470  012737  000700  001124          MOV     #MDPE!OR!IR,$GDDAT      ;GET GOOD = 700
8800  025476  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8801
8802  025504  017737  156360  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8803  025512  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8804                                                                  ;DATA WITH DATA READ FROM
8805                                                                  ;RHCS2
8806  025520  001401                          BEQ     69$             ;BRANCH IF GOOD
8807  025522  104105                          ERROR   105
```

```
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;RHCS2 SHOULD HAVE MDPE!OR!IR
                                                                ;=700
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
025524                                  69$:

025524  017737  156352  001202        MOV     @RHDB,$TMP2     ;READ RHDB SECOND TIME
                                      ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
025532  012737  144200  001124        MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200

025540  017737  156314  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
025546  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
025554  001401                        BEQ     71$             ;BRANCH IF GOOD
025556  104140                        ERROR   140
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;RHCS1 SHOULD HAVE SC!TRE!DVA!RDY
                                                                ;=144200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
025560                                  71$:
                                      ;CHECK THAT RHCS2 HAS MDPE!IR
025560  012737  000500  001124        MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
025566  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

025574  017737  156270  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
025602  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
025610  001401                        BEQ     73$             ;BRANCH IF GOOD
025612  104105                        ERROR   105
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;RHCS2 SHOULD HAVE MDPE!IR
                                                                ;=500
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
025614                                  73$:
```

```
8864                                                    ;CHECK THAT RHCS3 HAS 10
8865  025614  012737  000010  001124                    MOV     #10,$GDDAT      ;GET GOOD = 10
8866
8867  025622  017737  156304  001126                    MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8868  025630  023737  001124  001126                    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8869                                                                            ;DATA WITH DATA READ FROM
8870                                                                            ;RHCS3
8871  025636  001401                                    BEQ     75$             ;BRANCH IF GOOD
8872  025640  104104                                    ERROR   104
8873                                                                            ;AFTER SETTING "CLR" BIT #5
8874                                                                            ;IN RHCS2 TO INIT THE RH
8875                                                                            ;IPCK3 BIT #3 RHCS3 WAS SET
8876                                                                            ;ZEROS WERE MOVED INTO
8877                                                                            ;RHDB
8878                                                                            ;RHDB WAS READ THEN
8879                                                                            ;RHCS3 SHOULD HAVE 10
8880                                                                            ;=10
8881                                                                            ;BUT CONTAINED WHAT IS
8882                                                                            ;GIVEN IN BAD RHCS3
8883  025642                                75$:
8884
8885  025642  004737  040312                            JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
8886                                                                            ;SETUP UNIT NUBER
8887                                                                            ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8888
8889
8890                                                    ;CHECK THAT RHCS1 HAS RDY!DVA
8891  025646  012737  004200  001124                    MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
8892
8893  025654  017737  156200  001126                    MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8894  025662  023737  001124  001126                    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8895                                                                            ;DATA WITH DATA READ FROM
8896                                                                            ;RHCS1
8897  025670  001401                                    BEQ     77$             ;BRANCH IF GOOD
8898  025672  104106                                    ERROR   106
8899                                                                            ;AFTER SETTING "CLR" BIT #5
8900                                                                            ;IN RHCS2 TO INIT THE RH
8901                                                                            ;IPCK3 BIT #3 RHCS3 WAS SET
8902                                                                            ;ZEROS WERE MOVED INTO
8903                                                                            ;RHDB
8904                                                                            ;RHDB WAS READ THEN
8905                                                                            ;AN RH CLEAR (RHCS2 BIT #5)
8906                                                                            ;WAS GIVEN THIS SHOULD
8907                                                                            ;CLEAR ALL ERRORS
8908                                                                            ;RHCS1 SHOULD HAVE RDY!DVA
8909                                                                            ;=4200
8910                                                                            ;BUT CONTAINED WHAT IS
8911                                                                            ;GIVEN IN BAD RHCS1
8912  025674                                77$:
8913                                                    ;CHECK THAT RHCS2 HAS IR
8914  025674  012737  000100  001124                    MOV     #IR,$GDDAT      ;GET GOOD = 100
8915  025702  053737  005010  001124                    BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8916
8917  025710  017737  156154  001126                    MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8918  025716  023737  001124  001126                    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8919                                                                            ;DATA WITH DATA READ FROM
```

```
8920
8921  025724  001401                        BEQ     79$             ;RHCS2
8922  025726  104107                        ERROR   107             ;BRANCH IF GOOD
8923                                                                ;AFTER SETTING "CLR" BIT #5
8924                                                                ;IN RHCS2 TO INIT THE RH
8925                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
8926                                                                ;ZEROS WERE MOVED INTO
8927                                                                ;RHDB
8928                                                                ;RHDB WAS READ THEN
8929                                                                ;AN RH CLEAR (RHCS2 BIT #5)
8930                                                                ;WAS GIVEN THIS SHOULD
8931                                                                ;CLEAR ALL ERRORS
8932                                                                ;RHCS2 SHOULD HAVE IR
8933                                                                ;=100
8934                                                                ;TOGETHER WITH UNIT NUMBER
8935                                                                ;BUT CONTAINED WHAT IS
8936                                                                ;GIVEN IN BAD RHCS2
8937  025730                            79$:
8938                                    ;CHECK THAT RHCS3 HAS 0
8939  025730  012737  000000  001124      MOV     #0,$GDDAT       ;GET GOOD = 0
8940
8941  025736  017737  156170  001126      MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8942  025744  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8943                                                                ;DATA WITH DATA READ FROM
8944                                                                ;RHCS3
8945  025752  001401                        BEQ     81$             ;BRANCH IF GOOD
8946  025754  104110                        ERROR   110
8947                                                                ;AFTER SETTING "CLR" BIT #5
8948                                                                ;IN RHCS2 TO INIT THE RH
8949                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
8950                                                                ;ZEROS WERE MOVED INTO
8951                                                                ;RHDB
8952                                                                ;RHDB WAS READ THEN
8953                                                                ;AN RH CLEAR (RHCS2 BIT #5)
8954                                                                ;WAS GIVEN THIS SHOULD
8955                                                                ;CLEAR ALL ERRORS
8956                                                                ;RHCS3 SHOULD HAVE 0
8957                                                                ;BUT CONTAINED WHAT IS
8958                                                                ;GIVEN IN BAD RHCS3
8959  025756                            81$:
8960                                    ;CHECK THAT RHBA HAS 0
8961  025756  012737  000000  001124      MOV     #0,$GDDAT       ;GET GOOD = 0
8962
8963  025764  017737  156074  001126      MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
8964  025772  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8965                                                                ;DATA WITH DATA READ FROM
8966                                                                ;RHBA
8967  026000  001401                        BEQ     83$             ;BRANCH IF GOOD
8968  026002  104111                        ERROR   111
8969                                                                ;AFTER SETTING "CLR" BIT #5
8970                                                                ;IN RHCS2 TO INIT THE RH
8971                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
8972                                                                ;ZEROS WERE MOVED INTO
8973                                                                ;RHDB
8974                                                                ;RHDB WAS READ THEN
8975                                                                ;AN RH CLEAR (RHCS2 BIT #5)
```

```
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
026004                                  83$:
                                                ;CHECK THAT RHBAE HAS 0
026004  012737  000000  001124                 MOV     #0,$GDDAT      ;GET GOOD = 0

026012  017737  156112  001126                 MOV     @RHBAE,$BDDAT  ;READ RHBAE FOR COMPARISON
026020  023737  001124  001126                 CMP     $GDDAT,$BDDAT  ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
026026  001401                                BEQ     85$            ;BRANCH IF GOOD
026030  104112                                ERROR   112

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
026032                                  85$:
                                                ;CHECK THAT RHWC HAS 0
026032  012737  000000  001124                 MOV     #0,$GDDAT      ;GET GOOD = 0

026040  017737  156016  001126                 MOV     @RHWC,$BDDAT   ;READ RHWC FOR COMPARISON
026046  023737  001124  001126                 CMP     $GDDAT,$BDDAT  ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
026054  001401                                BEQ     87$            ;BRANCH IF GOOD
026056  104113                                ERROR   113

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;IPCK3 BIT #3 RHCS3 WAS SET
                                                                ;ZEROS WERE MOVED INTO
                                                                ;RHDB
                                                                ;RHDB WAS READ THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
026060                                  87$:


                                ;;*****************************************************************
                                ;*TEST 34       RHCS2-WCE BIT #14 AND RHCS3-WCE OW BIT #12
```

```
                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        ;*      WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                        ;*      SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                        ;*      ODD WORD BOUNDARY
                                        ;*      DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                        ;*      ODD WORD BOUNDARY
                                        ;*      THIS SHOULD SET WCE OW (RHCS3 BIT #12)
                                        ;*      AND WCE (RHCS2 BIT #14)
                                        ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                                        ;*      ALL ERRORS ARE CLEARED
                                        ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                        ;;*****************************************************************
026060  000004                          TST34:  SCOPE
026062  012706  001000                          MOV     #STACK,SP       ;RESET STACK
026066  012737  000034  004656                  MOV     #34,@#TSTNM     ;SAVE TEST NUMBER

026074  004737  040312                          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                                ;WRITE ONE WORD OF ALL ONES ON THE DEVICE
026100  012737  177777  004210                  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
026106  012777  177767  155746                  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026114  012777  004210  155742                  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

026122  004077  156512                          JSR     R0,@COMND       ;GO TO DO COMMAND
026126  004154                                  WRIDAT                  ;WRITE DATA

                                                ;SET UP FOR WRITE CHECK
026130  005037  003002                          CLR     BUFOW           ;WRITE DATA FOR WRITE CHECK ERROR
026134  012777  177767  155720                  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026142  012777  003002  155714                  MOV     #BUFOW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
                                                                        ;FROM ODD WORD BOUNDRY

                                                ;WRITE CHECK ON ODD WORD BOUNDARY
026150  004077  156464                          JSR     R0,@COMND       ;GO TO DO COMMAND
026154  004150                                  WRCHEK                  ;WRITE CHECK DATA


                                                ;CHECK THAT RHCS2 HAS WCE!OR!IR
026156  012737  040300  001124                  MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
026164  053737  005010  001124                  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

026172  017737  155672  001126                  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
026200  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS2
026206  001401                                  BEQ     65$             ;BRANCH IF GOOD
026210  104114                                  ERROR   114
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
```

```
                                                        ;A WRITE CHECK WAS DONE
                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                        ;RHCS2 SHOULD HAVE WCE!OR!IR
                                                        ;=40300
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
026212                          65$:

                                        ;CHECK THAT RHCS3 HAS DBL!WCEOW
026212  012737  012000  001124          MOV     #DBL!WCEOW,$GDDAT       ;GET GOOD = 12000

026220  017737  155706  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
026226  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
026234  001401                          BEQ     67$             ;BRANCH IF GOOD
026236  104115                          ERROR   115
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;ONE WORD OF ALL ONES IS
                                                        ;WRITTEN ON THE DEVICE
                                                        ;A WRITE CHECK WAS DONE
                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                        ;RHCS3 SHOULD HAVE DBL!WCEOW
                                                        ;=12000
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS3
026240                          67$:


026240  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS1 HAS RDY!DVA
026244  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200

026252  017737  155602  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
026260  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
026266  001401                          BEQ     69$             ;BRANCH IF GOOD
026270  104116                          ERROR   116
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;ONE WORD OF ALL ONES IS
                                                        ;WRITTEN ON THE DEVICE
                                                        ;A WRITE CHECK WAS DONE
                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                        ;WAS GIVEN THIS SHOULD
                                                        ;CLEAR ALL ERRORS
                                                        ;RHCS1 SHOULD HAVE RDY!DVA
                                                        ;=4200
```

```
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
026272                                  69$:
                        ;CHECK THAT RHCS2 HAS IR
026272  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
026300  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

026306  017737  155556  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
026314  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
026322  001401                          BEQ     71$             ;BRANCH IF GOOD
026324  104117                          ERROR   117
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
026326                                  71$:
                        ;CHECK THAT RHCS3 HAS 0
026326  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

026334  017737  155572  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
026342  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
026350  001401                          BEQ     73$             ;BRANCH IF GOOD
026352  104120                          ERROR   120
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHCS3 SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
026354                                  73$:
                        ;CHECK THAT RHBA HAS 0
026354  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

026362  017737  155476  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
026370  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
```

```
                                                                ;RHBA
026376  001401                        BEQ     75$             ;BRANCH IF GOOD
026400  104121                        ERROR   121
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
026402                          75$:
                                ;CHECK THAT RHBAE HAS 0
026402  012737  000000  001124  MOV     #0,$GDDAT               ;GET GOOD = 0

026410  017737  155514  001126  MOV     @RHBAE,$BDDAT           ;READ RHBAE FOR COMPARISON
026416  023737  001124  001126  CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
026424  001401                        BEQ     77$             ;BRANCH IF GOOD
026426  104122                        ERROR   122
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
026430                          77$:
                                ;CHECK THAT RHWC HAS 0
026430  012737  000000  001124  MOV     #0,$GDDAT               ;GET GOOD = 0

026436  017737  155420  001126  MOV     @RHWC,$BDDAT            ;READ RHWC FOR COMPARISON
026444  023737  001124  001126  CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
026452  001401                        BEQ     79$             ;BRANCH IF GOOD
026454  104123                        ERROR   123
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
```

```
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
026456                          79$:


                        ;;**************************************************************
                        ;*TEST 35       RHCS2-WCE BIT #14 AND RHCS3-WCE OW BIT #12

                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                        ;*      WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                        ;*      SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                        ;*      ODD WORD BOUNDARY
                        ;*      DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                        ;*      ODD WORD BOUNDARY
                        ;*      THIS SHOULD SET WCE OW (RHCS3 BIT #12)
                        ;*      AND WCE (RHCS2 BIT #14)
                        ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                        ;*      ALL ERRORS ARE CLEARED
                        ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                        ;;**************************************************************
026456  000004          TST35:  SCOPE
026460  012706  001000          MOV     #STACK,SP       ;RESET STACK
026464  012737  000035  004656  MOV     #35,@#TSTNM     ;SAVE TEST NUMBER

026472  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                        ;WRITE ONE WORD OF ALL ONES ON THE DEVICE
026476  012737  177777  004210  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
026504  012777  177767  155350  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026512  012777  004210  155344  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

026520  004077  156114          JSR     R0,@COMND       ;GO TO DO COMMAND
026524  004154                  WRIDAT                  ;WRITE DATA

                        ;SET UP FOR WRITE CHECK
026526  005037  003002          CLR     BUFOW           ;WRITE DATA FOR WRITE CHECK ERROR
026532  012777  177767  155322  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026540  012777  003002  155316  MOV     #BUFOW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
                                                        ;FROM ODD WORD BOUNDRY
                        ;WRITE CHECK ON ODD WORD BOUNDARY
026546  004077  156066          JSR     R0,@COMND       ;GO TO DO COMMAND
026552  004150                  WRCHEK                  ;WRITE CHECK DATA


                        ;CHECK THAT RHCS2 HAS WCE!OR!IR
026554  012737  040300  001124  MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
026562  053737  005010  001124  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

026570  017737  155274  001126  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
```

```
9312  026576  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9313                                                                    ;DATA WITH DATA READ FROM
9314                                                                    ;RHCS2
9315  026604  001401                          BEQ     65$             ;BRANCH IF GOOD
9316  026606  104114                          ERROR   114
9317                                                                    ;AFTER SETTING "CLR" BIT #5
9318                                                                    ;IN RHCS2 TO INIT THE RH
9319                                                                    ;ONE WORD OF ALL ONES IS
9320                                                                    ;WRITTEN ON THE DEVICE
9321                                                                    ;A WRITE CHECK WAS DONE
9322                                                                    ;ON AN ODD WORD BOUNDARY THEN
9323                                                                    ;RHCS2 SHOULD HAVE WCE!OR!IR
9324                                                                    ;=40300
9325                                                                    ;TOGETHER WITH UNIT NUMBER
9326                                                                    ;BUT CONTAINED WHAT IS
9327                                                                    ;GIVEN IN BAD RHCS2
9328  026610                          65$:
9329
9330                                          ;CHECK THAT RHCS3 HAS DBL!WCEOW
9331  026610  012737  012000  001124          MOV     #DBL!WCEOW,$GDDAT       ;GET GOOD = 12000
9332
9333  026616  017737  155310  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
9334  026624  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9335                                                                    ;DATA WITH DATA READ FROM
9336                                                                    ;RHCS3
9337  026632  001401                          BEQ     67$             ;BRANCH IF GOOD
9338  026634  104115                          ERROR   115
9339                                                                    ;AFTER SETTING "CLR" BIT #5
9340                                                                    ;IN RHCS2 TO INIT THE RH
9341                                                                    ;ONE WORD OF ALL ONES IS
9342                                                                    ;WRITTEN ON THE DEVICE
9343                                                                    ;A WRITE CHECK WAS DONE
9344                                                                    ;ON AN ODD WORD BOUNDARY THEN
9345                                                                    ;RHCS3 SHOULD HAVE DBL!WCEOW
9346                                                                    ;=12000
9347                                                                    ;BUT CONTAINED WHAT IS
9348                                                                    ;GIVEN IN BAD RHCS3
9349  026636                          67$:
9350
9351
9352  026636  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
9353                                                                    ;SETUP UNIT NUBER
9354                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
9355
9356
9357                                          ;CHECK THAT RHCS1 HAS RDY!DVA
9358  026642  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
9359
9360  026650  017737  155204  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
9361  026656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9362                                                                    ;DATA WITH DATA READ FROM
9363                                                                    ;RHCS1
9364  026664  001401                          BEQ     69$             ;BRANCH IF GOOD
9365  026666  104116                          ERROR   116
9366                                                                    ;AFTER SETTING "CLR" BIT #5
9367                                                                    ;IN RHCS2 TO INIT THE RH
```

```
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHCS1 SHOULD HAVE RDY!DVA
                                                                        ;=4200
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS1
026670                                  69$:
                                        ;CHECK THAT RHCS2 HAS IR
026670  012737  000100  001124          MOV     #IR,$GDDAT              ;GET GOOD = 100
026676  053737  005010  001124          BIS     UNIT,$GDDAT             ;INCLUDE UNIT NUMBER

026704  017737  155160  001126          MOV     @RHCS2,$BDDAT           ;READ RHCS2 FOR COMPARISON
026712  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS2
026720  001401                          BEQ     71$                     ;BRANCH IF GOOD
026722  104117                          ERROR   117
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHCS2 SHOULD HAVE IR
                                                                        ;=100
                                                                        ;TOGETHER WITH UNIT NUMBER
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS2
026724                                  71$:
                                        ;CHECK THAT RHCS3 HAS 0
026724  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

026732  017737  155174  001126          MOV     @RHCS3,$BDDAT           ;READ RHCS3 FOR COMPARISON
026740  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS3
026746  001401                          BEQ     73$                     ;BRANCH IF GOOD
026750  104120                          ERROR   120
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHCS3 SHOULD HAVE 0
```

```
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
026752                                  73$:
                                                ;CHECK THAT RHBA HAS 0
026752  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

026760  017737  155100  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
026766  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBA
026774  001401                          BEQ     75$             ;BRANCH IF GOOD
026776  104121                          ERROR   121
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
027000                                  75$:
                                                ;CHECK THAT RHBAE HAS 0
027000  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027006  017737  155116  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
027014  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
027022  001401                          BEQ     77$             ;BRANCH IF GOOD
027024  104122                          ERROR   122
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN ODD WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
027026                                  77$:
                                                ;CHECK THAT RHWC HAS 0
027026  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027034  017737  155022  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
027042  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
027050  001401                          BEQ     79$             ;BRANCH IF GOOD
027052  104123                          ERROR   123
```

```
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;ONE WORD OF ALL ONES IS
                                                        ;WRITTEN ON THE DEVICE
                                                        ;A WRITE CHECK WAS DONE
                                                        ;ON AN ODD WORD BOUNDARY THEN
                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                        ;WAS GIVEN THIS SHOULD
                                                        ;CLEAR ALL ERRORS
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
027054                          79$:


                                ;;*****************************************************************
                                ;*TEST 36       RHCS2-WCE BIT #14 AND RHCS3-WCE EW BIT #11

                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                ;*      WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                ;*      SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                ;*      EVEN WORD BOUNDARY
                                ;*      DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                ;*      EVEN WORD BOUNDARY
                                ;*      THIS SHOULD SET WCE EW (RHCS3 BIT #11)
                                ;*      AND WCE (RHCS2 BIT #14)
                                ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                                ;*      ALL ERRORS ARE CLEARED
                                ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                ;;*****************************************************************
027054  000004          TST36:  SCOPE
027056  012706  001000          MOV     #STACK,SP       ;RESET STACK
027062  012737  000036  004656  MOV     #36,@#TSTNM     ;SAVE TEST NUMBER

027070  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                ;WRITE ONE WORD OF ALL ONES ON THE DEVICE
027074  012737  177777  004210  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
027102  012777  177767  154752  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
027110  012777  004210  154746  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

027116  004077  155516          JSR     R0,@COMND       ;GO TO DO COMMAND
027122  004154                  WRIDAT                  ;WRITE DATA

                                ;SET UP FOR WRITE CHECK
027124  005037  003000          CLR     BUFEW           ;WRITE DATA FOR WRITE CHECK ERROR
027130  012777  177767  154724  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
027136  012777  003000  154720  MOV     #BUFEW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
                                                        ;FROM EVEN WORD BOUNDRY

                                ;WRITE CHECK ON EVEN WORD BOUNDARY
027144  004077  155470          JSR     R0,@COMND       ;GO TO DO COMMAND
027150  004150                  WRCHEK                  ;WRITE CHECK DATA
```

```
                                        ;CHECK THAT RHCS2 HAS WCE!OR!IR
027152  012737  040300  001124          MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
027160  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

027166  017737  154676  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
027174  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
027202  001401                          BEQ     65$             ;BRANCH IF GOOD
027204  104114                          ERROR   114
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN EVEN WORD BOUNDARY THEN
                                                                ;RHCS2 SHOULD HAVE WCE!OR!IR
                                                                ;=40300
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
027206                          65$:

                                        ;CHECK THAT RHCS3 HAS DBL!WCEEW
027206  012737  006000  001124          MOV     #DBL!WCEEW,$GDDAT       ;GET GOOD = 6000

027214  017737  154712  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
027222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
027230  001401                          BEQ     67$             ;BRANCH IF GOOD
027232  104115                          ERROR   115
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN EVEN WORD BOUNDARY THEN
                                                                ;RHCS3 SHOULD HAVE DBL!WCEEW
                                                                ;=6000
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
027234                          67$:

027234  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS1 HAS RDY!DVA
027240  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
```

```
9592
9593  027246  017737  154606  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
9594  027254  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9595                                                                  ;DATA WITH DATA READ FROM
9596                                                                  ;RHCS1
9597  027262  001401                          BEQ     69$             ;BRANCH IF GOOD
9598  027264  104116                          ERROR   116
9599                                                                  ;AFTER SETTING "CLR" BIT #5
9600                                                                  ;IN RHCS2 TO INIT THE RH
9601                                                                  ;ONE WORD OF ALL ONES IS
9602                                                                  ;WRITTEN ON THE DEVICE
9603                                                                  ;A WRITE CHECK WAS DONE
9604                                                                  ;ON AN EVEN WORD BOUNDARY THEN
9605                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
9606                                                                  ;WAS GIVEN THIS SHOULD
9607                                                                  ;CLEAR ALL ERRORS
9608                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
9609                                                                  ;=4200
9610                                                                  ;BUT CONTAINED WHAT IS
9611                                                                  ;GIVEN IN BAD RHCS1
9612  027266                          69$:
9613                                          ;CHECK THAT RHCS2 HAS IR
9614  027266  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
9615  027274  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
9616
9617  027302  017737  154562  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
9618  027310  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9619                                                                  ;DATA WITH DATA READ FROM
9620                                                                  ;RHCS2
9621  027316  001401                          BEQ     71$             ;BRANCH IF GOOD
9622  027320  104117                          ERROR   117
9623                                                                  ;AFTER SETTING "CLR" BIT #5
9624                                                                  ;IN RHCS2 TO INIT THE RH
9625                                                                  ;ONE WORD OF ALL ONES IS
9626                                                                  ;WRITTEN ON THE DEVICE
9627                                                                  ;A WRITE CHECK WAS DONE
9628                                                                  ;ON AN EVEN WORD BOUNDARY THEN
9629                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
9630                                                                  ;WAS GIVEN THIS SHOULD
9631                                                                  ;CLEAR ALL ERRORS
9632                                                                  ;RHCS2 SHOULD HAVE IR
9633                                                                  ;=100
9634                                                                  ;TOGETHER WITH UNIT NUMBER
9635                                                                  ;BUT CONTAINED WHAT IS
9636                                                                  ;GIVEN IN BAD RHCS2
9637  027322                          71$:
9638                                          ;CHECK THAT RHCS3 HAS 0
9639  027322  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
9640
9641  027330  017737  154576  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
9642  027336  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9643                                                                  ;DATA WITH DATA READ FROM
9644                                                                  ;RHCS3
9645  027344  001401                          BEQ     73$             ;BRANCH IF GOOD
9646  027346  104120                          ERROR   120
9647                                                                  ;AFTER SETTING "CLR" BIT #5
```

```
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN EVEN WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHCS3 SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS3
027350                                  73$:
                                                ;CHECK THAT RHBA HAS 0
027350  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

027356  017737  154502  001126          MOV     @RHBA,$BDDAT            ;READ RHBA FOR COMPARISON
027364  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBA
027372  001401                          BEQ     75$                     ;BRANCH IF GOOD
027374  104121                          ERROR   121

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN EVEN WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHBA SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBA
027376                                  75$:
                                                ;CHECK THAT RHBAE HAS 0
027376  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

027404  017737  154520  001126          MOV     @RHBAE,$BDDAT           ;READ RHBAE FOR COMPARISON
027412  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBAE
027420  001401                          BEQ     77$                     ;BRANCH IF GOOD
027422  104122                          ERROR   122

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN EVEN WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHBAE SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBAE
027424                                  77$:
```

N14


```
                                        ;CHECK THAT RHWC HAS 0
027424  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027432  017737  154424  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
027440  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
027446  001401                          BEQ     79$             ;BRANCH IF GOOD
027450  104123                          ERROR   123

                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;ONE WORD OF ALL ONES IS
                                                                ;WRITTEN ON THE DEVICE
                                                                ;A WRITE CHECK WAS DONE
                                                                ;ON AN EVEN WORD BOUNDARY THEN
                                                                ;AN RH CLEAR (RHCS2 BIT #5)
                                                                ;WAS GIVEN THIS SHOULD
                                                                ;CLEAR ALL ERRORS
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
027452                          79$:


                                ;;**************************************************************
                                ;*TEST 37       RHCS2-WCE BIT #14 AND RHCS3-WCE EW BIT #11

                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                ;*      WRITE NINE WORD OF ALL NINES ON TO THE DEVICE
                                ;*      SET UP TO DO WRITE CHECK ON THAT WORD FROM AN
                                ;*      EVEN WORD BOUNDARY
                                ;*      DO A ONE WORD WRITE CHECK WITH RHBA FROM AN
                                ;*      EVEN WORD BOUNDARY
                                ;*      THIS SHOULD SET WCE EW (RHCS3 BIT #11)
                                ;*      AND WCE (RHCS2 BIT #14)
                                ;*      "CLR" (RHCS2 BIT #5) IS GIVEN
                                ;*      ALL ERRORS ARE CLEARED
                                ;*      CHECK RHCS1, RHCS2, RHCS3, RHBA, RHBAE, RHWC

                                ;;**************************************************************
027452  000004                  TST37:  SCOPE
027454  012706  001000                  MOV     #STACK,SP       ;RESET STACK
027460  012737  000037  004656          MOV     #37,@#TSTNM     ;SAVE TEST NUMBER

027466  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;WRITE ONE WORD OF ALL ONES ON THE DEVICE
027472  012737  177777  004210          MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
027500  012777  177767  154354          MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
027506  012777  004210  154350          MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

027514  004077  155120                  JSR     R0,@COMND       ;GO TO DO COMMAND
027520  004154                          WRIDAT                  ;WRITE DATA
```

```
 9760
 9761                                          ;SET UP FOR WRITE CHECK
 9762  027522  005037  003000                  CLR     BUFEW           ;WRITE DATA FOR WRITE CHECK ERROR
 9763  027526  012777  177767  154326          MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
 9764  027534  012777  003000  154322          MOV     #BUFEW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
 9765                                          ;FROM EVEN WORD BOUNDRY
 9766                                          ;WRITE CHECK ON EVEN WORD BOUNDARY
 9767  027542  004077  155072                  JSR     R0,@COMND       ;GO TO DO COMMAND
 9768  027546  004150                          WRCHEK                  ;WRITE CHECK DATA
 9769
 9770
 9771
 9772
 9773                                          ;CHECK THAT RHCS2 HAS WCE!OR!IR
 9774  027550  012737  040300  001124          MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
 9775  027556  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 9776
 9777  027564  017737  154300  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 9778  027572  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9779                                          ;DATA WITH DATA READ FROM
 9780                                          ;RHCS2
 9781  027600  001401                          BEQ     65$             ;BRANCH IF GOOD
 9782  027602  104114                          ERROR   114
 9783                                          ;AFTER SETTING "CLR" BIT #5
 9784                                          ;IN RHCS2 TO INIT THE RH
 9785                                          ;ONE WORD OF ALL ONES IS
 9786                                          ;WRITTEN ON THE DEVICE
 9787                                          ;A WRITE CHECK WAS DONE
 9788                                          ;ON AN EVEN WORD BOUNDARY THEN
 9789                                          ;RHCS2 SHOULD HAVE WCE!OR!IR
 9790                                          ;=40300
 9791                                          ;TOGETHER WITH UNIT NUMBER
 9792                                          ;BUT CONTAINED WHAT IS
 9793                                          ;GIVEN IN BAD RHCS2
 9794  027604                          65$:
 9795
 9796                                          ;CHECK THAT RHCS3 HAS DBL!WCEEW
 9797  027604  012737  006000  001124          MOV     #DBL!WCEEW,$GDDAT       ;GET GOOD = 6000
 9798
 9799  027612  017737  154314  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 9800  027620  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9801                                          ;DATA WITH DATA READ FROM
 9802                                          ;RHCS3
 9803  027626  001401                          BEQ     67$             ;BRANCH IF GOOD
 9804  027630  104115                          ERROR   115
 9805                                          ;AFTER SETTING "CLR" BIT #5
 9806                                          ;IN RHCS2 TO INIT THE RH
 9807                                          ;ONE WORD OF ALL ONES IS
 9808                                          ;WRITTEN ON THE DEVICE
 9809                                          ;A WRITE CHECK WAS DONE
 9810                                          ;ON AN EVEN WORD BOUNDARY THEN
 9811                                          ;RHCS3 SHOULD HAVE DBL!WCEEW
 9812                                          ;=6000
 9813                                          ;BUT CONTAINED WHAT IS
 9814                                          ;GIVEN IN BAD RHCS3
 9815  027632                          67$:
```

```
                                        ;CHECK THAT RHCS1 HAS RDY!DVA
027636  012737  004200  001124        MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200

027644  017737  154210  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
027652  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHCS1
027660  001401                        BEQ     69$             ;BRANCH IF GOOD
027662  104116                        ERROR   116
                                                              ;AFTER SETTING "CLR" BIT #5
                                                              ;IN RHCS2 TO INIT THE RH
                                                              ;ONE WORD OF ALL ONES IS
                                                              ;WRITTEN ON THE DEVICE
                                                              ;A WRITE CHECK WAS DONE
                                                              ;ON AN EVEN WORD BOUNDARY THEN
                                                              ;AN RH CLEAR (RHCS2 BIT #5)
                                                              ;WAS GIVEN THIS SHOULD
                                                              ;CLEAR ALL ERRORS
                                                              ;RHCS1 SHOULD HAVE RDY!DVA
                                                              ;=4200
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHCS1
027664                          69$:
                                        ;CHECK THAT RHCS2 HAS IR
027664  012737  000100  001124        MOV     #IR,$GDDAT      ;GET GOOD = 100
027672  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

027700  017737  154164  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
027706  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                              ;DATA WITH DATA READ FROM
                                                              ;RHCS2
027714  001401                        BEQ     71$             ;BRANCH IF GOOD
027716  104117                        ERROR   117
                                                              ;AFTER SETTING "CLR" BIT #5
                                                              ;IN RHCS2 TO INIT THE RH
                                                              ;ONE WORD OF ALL ONES IS
                                                              ;WRITTEN ON THE DEVICE
                                                              ;A WRITE CHECK WAS DONE
                                                              ;ON AN EVEN WORD BOUNDARY THEN
                                                              ;AN RH CLEAR (RHCS2 BIT #5)
                                                              ;WAS GIVEN THIS SHOULD
                                                              ;CLEAR ALL ERRORS
                                                              ;RHCS2 SHOULD HAVE IR
                                                              ;=100
                                                              ;TOGETHER WITH UNIT NUMBER
                                                              ;BUT CONTAINED WHAT IS
                                                              ;GIVEN IN BAD RHCS2
027720                          71$:
                                        ;CHECK THAT RHCS3 HAS 0
```

```
 9872  027720  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9873
 9874  027726  017737  154200  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 9875  027734  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9876                                                                  ;DATA WITH DATA READ FROM
 9877                                                                  ;RHCS3
 9878  027742  001401                          BEQ     73$             ;BRANCH IF GOOD
 9879  027744  104120                          ERROR   120
 9880                                                                  ;AFTER SETTING "CLR" BIT #5
 9881                                                                  ;IN RHCS2 TO INIT THE RH
 9882                                                                  ;ONE WORD OF ALL ONES IS
 9883                                                                  ;WRITTEN ON THE DEVICE
 9884                                                                  ;A WRITE CHECK WAS DONE
 9885                                                                  ;ON AN EVEN WORD BOUNDARY THEN
 9886                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 9887                                                                  ;WAS GIVEN THIS SHOULD
 9888                                                                  ;CLEAR ALL ERRORS
 9889                                                                  ;RHCS3 SHOULD HAVE 0
 9890                                                                  ;BUT CONTAINED WHAT IS
 9891                                                                  ;GIVEN IN BAD RHCS3
 9892  027746                          73$:
 9893                                          ;CHECK THAT RHBA HAS 0
 9894  027746  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9895
 9896  027754  017737  154104  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
 9897  027762  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9898                                                                  ;DATA WITH DATA READ FROM
 9899                                                                  ;RHBA
 9900  027770  001401                          BEQ     75$             ;BRANCH IF GOOD
 9901  027772  104121                          ERROR   121
 9902                                                                  ;AFTER SETTING "CLR" BIT #5
 9903                                                                  ;IN RHCS2 TO INIT THE RH
 9904                                                                  ;ONE WORD OF ALL ONES IS
 9905                                                                  ;WRITTEN ON THE DEVICE
 9906                                                                  ;A WRITE CHECK WAS DONE
 9907                                                                  ;ON AN EVEN WORD BOUNDARY THEN
 9908                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 9909                                                                  ;WAS GIVEN THIS SHOULD
 9910                                                                  ;CLEAR ALL ERRORS
 9911                                                                  ;RHBA SHOULD HAVE 0
 9912                                                                  ;BUT CONTAINED WHAT IS
 9913                                                                  ;GIVEN IN BAD RHBA
 9914  027774                          75$:
 9915                                          ;CHECK THAT RHBAE HAS 0
 9916  027774  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9917
 9918  030002  017737  154122  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
 9919  030010  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9920                                                                  ;DATA WITH DATA READ FROM
 9921                                                                  ;RHBAE
 9922  030016  001401                          BEQ     77$             ;BRANCH IF GOOD
 9923  030020  104122                          ERROR   122
 9924                                                                  ;AFTER SETTING "CLR" BIT #5
 9925                                                                  ;IN RHCS2 TO INIT THE RH
 9926                                                                  ;ONE WORD OF ALL ONES IS
 9927                                                                  ;WRITTEN ON THE DEVICE
```

```
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN EVEN WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHBAE SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBAE
030022                                  77$:
                                                ;CHECK THAT RHWC HAS 0
030022  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

030030  017737  154026  001126          MOV     @RHWC,$BDDAT            ;READ RHWC FOR COMPARISON
030036  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHWC
030044  001401                          BEQ     79$                     ;BRANCH IF GOOD
030046  104123                          ERROR   123

                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;ONE WORD OF ALL ONES IS
                                                                        ;WRITTEN ON THE DEVICE
                                                                        ;A WRITE CHECK WAS DONE
                                                                        ;ON AN EVEN WORD BOUNDARY THEN
                                                                        ;AN RH CLEAR (RHCS2 BIT #5)
                                                                        ;WAS GIVEN THIS SHOULD
                                                                        ;CLEAR ALL ERRORS
                                                                        ;RHWC SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHWC
030050                                  79$:



                                        ;;*****************************************************************
                                        ;*TEST 40       TEST DBL (RHCS3 BIT #10) TEST A

                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        ;*      SET UP FOR A NINE WORD WRITE FROM AN EVEN WORD BOUNDARY
                                        ;*      DO A NINE WORD WRITE FROM AN EVEN WORD BOUNDARY
                                        ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                        ;*      CHECK RHCS3
                                        ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                        ;;*****************************************************************
030050  000004                        TST40:  SCOPE
030052  012706  001003                        MOV     #STACK,SP       ;RESET STACK
030056  012737  000040  004656                MOV     #40,@#TSTNM     ;SAVE TEST NUMBER

030064  004737  040312                        JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                                ;SET UP FOR A NINE WORD WRITE FROM AN EVEN WORD BOUNDARY
030070  012701  000003                        MOV     #3,R1           ;COUNT OF THREE
```

```
 9984  030074  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
 9985                                                                  ;FROM AN EVEN WORD BOUNDARY
 9986  030100  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
 9987  030104  005301                          DEC     R1              ;COUNT
 9988  030106  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
 9989  030110  012777  003000  153746          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
 9990                                                                  ;FROM AN EVEN WORD BOUNDARY
 9991  030116  012777  177767  153736          MOV #-9.,@RHWC ;WORD COUNT NINE
 9992  030124  004077  154510                  JSR     R0,@COMND       ;GO TO DO COMMAND
 9993  030130  004154                          WRIDAT                  ;WRITE DATA
 9994
 9995
 9996                                          ;CHECK THAT RHCS3 HAS 0
 9997  030132  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9998
 9999  030140  017737  153766  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10000  030146  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10001                                                                 ;DATA WITH DATA READ FROM
10002                                                                 ;RHCS3
10003  030154  001401                          BEQ     65$             ;BRANCH IF GOOD
10004  030156  104124                          ERROR   124
10005                                                                 ;AFTER SETTING "CLR" BIT #5
10006                                                                 ;IN RHCS2 TO INIT THE RH
10007                                                                 ;A NINE WORD WRITE FROM AN
10008                                                                 ;EVEN WORD BOUNDARY WAS DONE
10009                                                                 ;THEN
10010                                                                 ;RHCS3 SHOULD HAVE 0
10011                                                                 ;BUT CONTAINED WHAT IS
10012                                                                 ;GIVEN IN BAD RHCS3
10013  030160                           65$:
10014  030160  042777  000076  153672          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10015                                          ;CHECK THAT RHCS1 HAS RDY!DVA
10016  030166  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10017
10018  030174  017737  153660  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10019  030202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10020                                                                 ;DATA WITH DATA READ FROM
10021                                                                 ;RHCS1
10022  030210  001401                          BEQ     67$             ;BRANCH IF GOOD
10023  030212  104125                          ERROR   125
10024                                                                 ;AFTER SETTING "CLR" BIT #5
10025                                                                 ;IN RHCS2 TO INIT THE RH
10026                                                                 ;A NINE WORD WRITE FROM AN
10027                                                                 ;EVEN WORD BOUNDARY WAS DONE
10028                                                                 ;THEN
10029                                                                 ;RHCS1 SHOULD HAVE RDY!DVA
10030                                                                 ;=4200
10031                                                                 ;BUT CONTAINED WHAT IS
10032                                                                 ;GIVEN IN BAD RHCS1
10033  030214                           67$:
10034                                          ;CHECK THAT RHCS2 HAS IR
10035  030214  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10036  030222  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10037
10038  030230  017737  153634  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10039  030236  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
10040
10041
10042  030244  001401                    BEQ     69$             ;DATA WITH DATA READ FROM
10043  030246  104126                    ERROR   126             ;RHCS2
10044                                                            ;BRANCH IF GOOD
10045                                                            ;AFTER SETTING "CLR" BIT #5
10046                                                            ;IN RHCS2 TO INIT THE RH
10047                                                            ;A NINE WORD WRITE FROM AN
10048                                                            ;EVEN WORD BOUNDARY WAS DONE
10049                                                            ;THEN
10050                                                            ;RHCS2 SHOULD HAVE IR
10051                                                            ;=100
10052                                                            ;TOGETHER WITH UNIT NUMBER
10053                                                            ;BUT CONTAINED WHAT IS
10054  030250                       69$:                        ;GIVEN IN BAD RHCS2
10055                                       ;CHECK THAT RHWC HAS 0
10056  030250  012737  000000  001124      MOV     #0,$GDDAT       ;GET GOOD = 0
10057
10058  030256  017737  153600  001126      MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10059  030264  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10060                                                            ;DATA WITH DATA READ FROM
10061                                                            ;RHWC
10062  030272  001401                    BEQ     71$             ;BRANCH IF GOOD
10063  030274  104131                    ERROR   131
10064                                                            ;AFTER SETTING "CLR" BIT #5
10065                                                            ;IN RHCS2 TO INIT THE RH
10066                                                            ;A NINE WORD WRITE FROM AN
10067                                                            ;EVEN WORD BOUNDARY WAS DONE
10068                                                            ;THEN
10069                                                            ;RHWC SHOULD HAVE 0
10070                                                            ;BUT CONTAINED WHAT IS
10071                                                            ;GIVEN IN BAD RHWC
10072  030276                       71$:
10073
10074
10075
10076                                ;;*****************************************************************
10077                                ;*TEST 41       TEST DBL (RHCS3 BIT #10) TEST B
10078
10079                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10080                                ;*      SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10081                                ;*      DO A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10082                                ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
10083                                ;*      CHECK RHCS3
10084                                ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10085
10086                                ;;*****************************************************************
10087  030276  000004                TST41:  SCOPE
10088  030300  012706  001000                MOV     #STACK,SP       ;RESET STACK
10089  030304  012737  000041  004656        MOV     #41,@#TSTNM     ;SAVE TEST NUMBER
10090
10091  030312  004737  040312                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10092                                                                ;SETUP UNIT NUBER
10093                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10094
10095                                ;SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
```

```
10096  030316  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
10097  030322  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10098                                                                  ;FROM AN EVEN WORD BOUNDARY
10099  030326  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10100  030332  005301                          DEC     R1              ;COUNT
10101  030334  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
10102  030336  012777  003000  153520          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10103                                                                  ;FROM AN EVEN WORD BOUNDARY
10104  030344  012777  177766  153510          MOV #-10.,@RHWC ;WORD COUNT TEN
10105  030352  004077  154262                  JSR     R0,@COMND       ;GO TO DO COMMAND
10106  030356  004154                          WRIDAT                  ;WRITE DATA
10107
10108
10109                                          ;CHECK THAT RHCS3 HAS DBL
10110  030360  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
10111
10112  030366  017737  153540  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10113  030374  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10114                                                                  ;DATA WITH DATA READ FROM
10115                                                                  ;RHCS3
10116  030402  001401                          BEQ     65$             ;BRANCH IF GOOD
10117  030404  104124                          ERROR   124
10118                                                                  ;AFTER SETTING "CLR" BIT #5
10119                                                                  ;IN RHCS2 TO INIT THE RH
10120                                                                  ;A TEN WORD WRITE FROM AN
10121                                                                  ;EVEN WORD BOUNDARY WAS DONE
10122                                                                  ;THEN
10123                                                                  ;RHCS3 SHOULD HAVE DBL
10124                                                                  ;=2000
10125                                                                  ;BUT CONTAINED WHAT IS
10126                                                                  ;GIVEN IN BAD RHCS3
10127  030406                          65$:
10128  030406  042777  000076  153444          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10129                                          ;CHECK THAT RHCS1 HAS RDY!DVA
10130  030414  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10131
10132  030422  017737  153432  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10133  030430  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10134                                                                  ;DATA WITH DATA READ FROM
10135                                                                  ;RHCS1
10136  030436  001401                          BEQ     67$             ;BRANCH IF GOOD
10137  030440  104125                          ERROR   125
10138                                                                  ;AFTER SETTING "CLR" BIT #5
10139                                                                  ;IN RHCS2 TO INIT THE RH
10140                                                                  ;A TEN WORD WRITE FROM AN
10141                                                                  ;EVEN WORD BOUNDARY WAS DONE
10142                                                                  ;THEN
10143                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
10144                                                                  ;=4200
10145                                                                  ;BUT CONTAINED WHAT IS
10146                                                                  ;GIVEN IN BAD RHCS1
10147  030442                          67$:
10148                                          ;CHECK THAT RHCS2 HAS IR
10149  030442  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10150  030450  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10151
```

```
10152  030456  017737  153406  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10153  030464  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10154                                                                  ;DATA WITH DATA READ FROM
10155                                                                  ;RHCS2
10156  030472  001401                          BEQ     69$             ;BRANCH IF GOOD
10157  030474  104126                          ERROR   126
10158                                                                  ;AFTER SETTING "CLR" BIT #5
10159                                                                  ;IN RHCS2 TO INIT THE RH
10160                                                                  ;A TEN WORD WRITE FROM AN
10161                                                                  ;EVEN WORD BOUNDARY WAS DONE
10162                                                                  ;THEN
10163                                                                  ;RHCS2 SHOULD HAVE IR
10164                                                                  ;=100
10165                                                                  ;TOGETHER WITH UNIT NUMBER
10166                                                                  ;BUT CONTAINED WHAT IS
10167                                                                  ;GIVEN IN BAD RHCS2
10168  030476                          69$:
10169                                          ;CHECK THAT RHWC HAS 0
10170  030476  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10171
10172  030504  017737  153352  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10173  030512  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10174                                                                  ;DATA WITH DATA READ FROM
10175                                                                  ;RHWC
10176  030520  001401                          BEQ     71$             ;BRANCH IF GOOD
10177  030522  104131                          ERROR   131
10178                                                                  ;AFTER SETTING "CLR" BIT #5
10179                                                                  ;IN RHCS2 TO INIT THE RH
10180                                                                  ;A TEN WORD WRITE FROM AN
10181                                                                  ;EVEN WORD BOUNDARY WAS DONE
10182                                                                  ;THEN
10183                                                                  ;RHWC SHOULD HAVE 0
10184                                                                  ;BUT CONTAINED WHAT IS
10185                                                                  ;GIVEN IN BAD RHWC
10186  030524                          71$:
10187
10188
10189
10190                                  ;;*****************************************************************
10191                                  ;*TEST 42        TEST DBL (RHCS3 BIT #10) TEST C
10192
10193                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10194                                  ;*      SET UP FOR A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
10195                                  ;*      DO A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
10196                                  ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10197                                  ;*      CHECK RHCS3
10198                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10199
10200                                  ;;*****************************************************************
10201  030524  000004                  TST42:  SCOPE
10202  030526  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10203  030532  012737  000042  004656          MOV     #42,@#TSTNM     ;SAVE TEST NUMBER
10204
10205  030540  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10206                                                                  ;SETUP UNIT NUBER
10207                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
```

```
10208
10209                                          ;SET UP FOR A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
10210  030544  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
10211  030550  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
10212                                                                  ;FROM AN ODD WORD BOUNDARY
10213  030554  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10214  030560  005301                          DEC     R1              ;COUNT
10215  030562  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
10216  030564  012777  003002  153272          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
10217                                                                  ;FROM AN ODD WORD BOUNDARY
10218  030572  012777  177766  153262          MOV #-10.,@RHWC ;WORD COUNT TEN
10219  030600  004077  154034                  JSR     R0,@COMND       ;GO TO DO COMMAND
10220  030604  004154                          WRIDAT                  ;WRITE DATA
10221
10222
10223                                          ;CHECK THAT RHCS3 HAS 0
10224  030606  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10225
10226  030614  017737  153312  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10227  030622  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10228                                                                  ;DATA WITH DATA READ FROM
10229                                                                  ;RHCS3
10230  030630  001401                          BEQ     65$             ;BRANCH IF GOOD
10231  030632  104124                          ERROR   124
10232                                                                  ;AFTER SETTING "CLR" BIT #5
10233                                                                  ;IN RHCS2 TO INIT THE RH
10234                                                                  ;A TEN WORD WRITE FROM AN
10235                                                                  ;ODD WORD BOUNDARY WAS DONE
10236                                                                  ;THEN
10237                                                                  ;RHCS3 SHOULD HAVE 0
10238                                                                  ;BUT CONTAINED WHAT IS
10239                                                                  ;GIVEN IN BAD RHCS3
10240  030634                          65$:
10241  030634  042777  000076  153216          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10242                                          ;CHECK THAT RHCS1 HAS RDY!DVA
10243  030642  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10244
10245  030650  017737  153204  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10246  030656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10247                                                                  ;DATA WITH DATA READ FROM
10248                                                                  ;RHCS1
10249  030664  001401                          BEQ     67$             ;BRANCH IF GOOD
10250  030666  104125                          ERROR   125
10251                                                                  ;AFTER SETTING "CLR" BIT #5
10252                                                                  ;IN RHCS2 TO INIT THE RH
10253                                                                  ;A TEN WORD WRITE FROM AN
10254                                                                  ;ODD WORD BOUNDARY WAS DONE
10255                                                                  ;THEN
10256                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
10257                                                                  ;=4200
10258                                                                  ;BUT CONTAINED WHAT IS
10259                                                                  ;GIVEN IN BAD RHCS1
10260  030670                          67$:
10261                                          ;CHECK THAT RHCS2 HAS IR
10262  030670  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10263  030676  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
10264
10265  030704  017737  153160  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10266  030712  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10267                                                                  ;DATA WITH DATA READ FROM
10268                                                                  ;RHCS2
10269  030720  001401                          BEQ     69$             ;BRANCH IF GOOD
10270  030722  104126                          ERROR   126
10271                                                                  ;AFTER SETTING "CLR" BIT #5
10272                                                                  ;IN RHCS2 TO INIT THE RH
10273                                                                  ;A TEN WORD WRITE FROM AN
10274                                                                  ;ODD WORD BOUNDARY WAS DONE
10275                                                                  ;THEN
10276                                                                  ;RHCS2 SHOULD HAVE IR
10277                                                                  ;=100
10278                                                                  ;TOGETHER WITH UNIT NUMBER
10279                                                                  ;BUT CONTAINED WHAT IS
10280                                                                  ;GIVEN IN BAD RHCS2
10281  030724                          69$:
10282                                          ;CHECK THAT RHWC HAS 0
10283  030724  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10284
10285  030732  017737  153124  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10286  030740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10287                                                                  ;DATA WITH DATA READ FROM
10288                                                                  ;RHWC
10289  030746  001401                          BEQ     71$             ;BRANCH IF GOOD
10290  030750  104131                          ERROR   131
10291                                                                  ;AFTER SETTING "CLR" BIT #5
10292                                                                  ;IN RHCS2 TO INIT THE RH
10293                                                                  ;A TEN WORD WRITE FROM AN
10294                                                                  ;ODD WORD BOUNDARY WAS DONE
10295                                                                  ;THEN
10296                                                                  ;RHWC SHOULD HAVE 0
10297                                                                  ;BUT CONTAINED WHAT IS
10298                                                                  ;GIVEN IN BAD RHWC
10299  030752                          71$:
10300
10301
10302
10303                                  ;;*****************************************************************
10304                                  ;*TEST 43       TEST DBL (RHCS3 BIT #10) TEST D
10305
10306                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10307                                  ;*      SET UP FOR A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10308                                  ;*      DO A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10309                                  ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10310                                  ;*      CHECK RHCS3
10311                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10312
10313                                  ;;*****************************************************************
10314  030752  000004                  TST43:  SCOPE
10315  030754  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10316  030760  012737  000043  004656          MOV     #43,@#TSTNM     ;SAVE TEST NUMBER
10317
10318  030766  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10319                                                                  ;SETUP UNIT NUBER
```

```
10320                                                           ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10321
10322                                           ;SET UP FOR A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10323  030772  012701  000003                   MOV     #3,R1           ;COUNT OF THREE
10324  030776  012702  003000                   MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10325                                                                   ;FROM AN EVEN WORD BOUNDARY
10326  031002  012722  052525           1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10327  031006  005301                           DEC     R1              ;COUNT
10328  031010  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
10329  031012  012777  003000  153044           MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10330                                                                   ;FROM AN EVEN WORD BOUNDARY
10331  031020  012777  177765  153034           MOV #-11.,@RHWC ;WORD COUNT ELEVEN
10332  031026  004077  153606                   JSR     R0,@COMND       ;GO TO DO COMMAND
10333  031032  004154                           WRIDAT                  ;WRITE DATA
10334
10335
10336                                           ;CHECK THAT RHCS3 HAS 0
10337  031034  012737  000000  001124           MOV     #0,$GDDAT       ;GET GOOD = 0
10338
10339  031042  017737  153064  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10340  031050  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10341                                                                   ;DATA WITH DATA READ FROM
10342                                                                   ;RHCS3
10343  031056  001401                           BEQ     65$             ;BRANCH IF GOOD
10344  031060  104124                           ERROR   124
10345                                                                   ;AFTER SETTING "CLR" BIT #5
10346                                                                   ;IN RHCS2 TO INIT THE RH
10347                                                                   ;A ELEVEN WORD WRITE FROM AN
10348                                                                   ;EVEN WORD BOUNDARY WAS DONE
10349                                                                   ;THEN
10350                                                                   ;RHCS3 SHOULD HAVE 0
10351                                                                   ;BUT CONTAINED WHAT IS
10352                                                                   ;GIVEN IN BAD RHCS3
10353  031062                           65$:
10354  031062  042777  000076  152770           BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10355                                           ;CHECK THAT RHCS1 HAS RDY!DVA
10356  031070  012737  004200  001124           MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10357
10358  031076  017737  152756  001126           MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10359  031104  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10360                                                                   ;DATA WITH DATA READ FROM
10361                                                                   ;RHCS1
10362  031112  001401                           BEQ     67$             ;BRANCH IF GOOD
10363  031114  104125                           ERROR   125
10364                                                                   ;AFTER SETTING "CLR" BIT #5
10365                                                                   ;IN RHCS2 TO INIT THE RH
10366                                                                   ;A ELEVEN WORD WRITE FROM AN
10367                                                                   ;EVEN WORD BOUNDARY WAS DONE
10368                                                                   ;THEN
10369                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
10370                                                                   ;=4200
10371                                                                   ;BUT CONTAINED WHAT IS
10372                                                                   ;GIVEN IN BAD RHCS1
10373  031116                           67$:
10374                                           ;CHECK THAT RHCS2 HAS IR
10375  031116  012737  000100  001124           MOV     #IR,$GDDAT      ;GET GOOD = 100
```

```
10376  031124  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10377
10378  031132  017737  152732  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10379  031140  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10380                                                                  ;DATA WITH DATA READ FROM
10381                                                                  ;RHCS2
10382  031146  001401                          BEQ     69$             ;BRANCH IF GOOD
10383  031150  104126                          ERROR   126
10384                                                                  ;AFTER SETTING "CLR" BIT #5
10385                                                                  ;IN RHCS2 TO INIT THE RH
10386                                                                  ;A ELEVEN WORD WRITE FROM AN
10387                                                                  ;EVEN WORD BOUNDARY WAS DONE
10388                                                                  ;THEN
10389                                                                  ;RHCS2 SHOULD HAVE IR
10390                                                                  ;=100
10391                                                                  ;TOGETHER WITH UNIT NUMBER
10392                                                                  ;BUT CONTAINED WHAT IS
10393                                                                  ;GIVEN IN BAD RHCS2
10394  031152                          69$:
10395                                          ;CHECK THAT RHWC HAS 0
10396  031152  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10397
10398  031160  017737  152676  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10399  031166  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10400                                                                  ;DATA WITH DATA READ FROM
10401                                                                  ;RHWC
10402  031174  001401                          BEQ     71$             ;BRANCH IF GOOD
10403  031176  104131                          ERROR   131
10404                                                                  ;AFTER SETTING "CLR" BIT #5
10405                                                                  ;IN RHCS2 TO INIT THE RH
10406                                                                  ;A ELEVEN WORD WRITE FROM AN
10407                                                                  ;EVEN WORD BOUNDARY WAS DONE
10408                                                                  ;THEN
10409                                                                  ;RHWC SHOULD HAVE 0
10410                                                                  ;BUT CONTAINED WHAT IS
10411                                                                  ;GIVEN IN BAD RHWC
10412  031200                          71$:
10413
10414
10415
10416                                  ;;*****************************************************************
10417                                  ;*TEST 44        TEST DBL (RHCS3 BIT #10) TEST E
10418
10419                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10420                                  ;*      SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10421                                  ;*      WITH BAI IN RHCS2 BIT #3 SET
10422                                  ;*      DO A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10423                                  ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10424                                  ;*      CHECK RHCS3
10425                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10426
10427                                  ;;*****************************************************************
10428  031200  000004                  TST44.  SCOPE
10429  031202  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10430  031206  012737  000044  004656          MOV     #44,@#TSTNM     ;SAVE TEST NUMBER
10431
```

```
10432  031214  004737  040312                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10433                                                                     ;SETUP UNIT NUBER
10434                                                                     ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10435
10436                                             ;SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10437  031220  012701  000003                    MOV     #3,R1           ;COUNT OF THREE
10438  031224  012702  003000                    MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10439                                                                     ;FROM AN EVEN WORD BOUNDARY
10440  031230  012722  052525          1$:       MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10441  031234  005301                            DEC     R1              ;COUNT
10442  031236  001374                            BNE     1$              ;BRANCH IF THREE NOT DONE
10443  031240  012777  003000  152616            MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10444                                                                     ;FROM AN EVEN WORD BOUNDARY
10445  031246  012777  177766  152606            MOV #-10.,@RHWC ;WORD COUNT TEN
10446  031254  052777  000010  152606  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
10447  031262  004077  153352                    JSR     R0,@COMND       ;GO TO DO COMMAND
10448  031266  004154                            WRIDAT                  ;WRITE DATA
10449
10450
10451                                             ;CHECK THAT RHCS3 HAS 0
10452  031270  012737  000000  001124            MOV     #0,$GDDAT       ;GET GOOD = 0
10453
10454  031276  017737  152630  001126            MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10455  031304  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10456                                                                     ;DATA WITH DATA READ FROM
10457                                                                     ;RHCS3
10458  031312  001401                            BEQ     65$             ;BRANCH IF GOOD
10459  031314  104124                            ERROR   124
10460                                                                     ;AFTER SETTING "CLR" BIT #5
10461                                                                     ;IN RHCS2 TO INIT THE RH
10462                                                                     ;A TEN WORD WRITE FROM AN
10463                                                                     ;EVEN WORD BOUNDARY WAS DONE
10464                                                                     ;WITH BAI IN RHCS2 SET
10465                                                                     ;THEN
10466                                                                     ;RHCS3 SHOULD HAVE 0
10467                                                                     ;BUT CONTAINED WHAT IS
10468                                                                     ;GIVEN IN BAD RHCS3
10469  031316                           65$:
10470  031316  042777  000076  152534            BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10471                                             ;CHECK THAT RHCS1 HAS RDY!DVA
10472  031324  012737  004200  001124            MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10473
10474  031332  017737  152522  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10475  031340  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10476                                                                     ;DATA WITH DATA READ FROM
10477                                                                     ;RHCS1
10478  031346  001401                            BEQ     67$             ;BRANCH IF GOOD
10479  031350  104125                            ERROR   125
10480                                                                     ;AFTER SETTING "CLR" BIT #5
10481                                                                     ;IN RHCS2 TO INIT THE RH
10482                                                                     ;A TEN WORD WRITE FROM AN
10483                                                                     ;EVEN WORD BOUNDARY WAS DONE
10484                                                                     ;WITH BAI IN RHCS2 SET
10485                                                                     ;THEN
10486                                                                     ;RHCS1 SHOULD HAVE RDY!DVA
10487                                                                     ;=4200
```

```
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS1
031352                          67$:
                                        ;CHECK THAT RHCS2 HAS IR!BAI
031352  012737  000110  001124        MOV     #IR!BAI,$GDDAT  ;GET GOOD = 110
031360  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
031366  017737  152476  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
031374  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
031402  001401                        BEQ     69$             ;BRANCH IF GOOD
031404  104126                        ERROR   126
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TEN WORD WRITE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;WITH BAI IN RHCS2 SET
                                                        ;THEN
                                                        ;RHCS2 SHOULD HAVE IR!BAI
                                                        ;=110
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
031406                          69$:
                                        ;CHECK THAT RHWC HAS 0
031406  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
031414  017737  152442  001126        MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
031422  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
031430  001401                        BEQ     71$             ;BRANCH IF GOOD
031432  104131                        ERROR   131
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TEN WORD WRITE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;WITH BAI IN RHCS2 SET
                                                        ;THEN
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
031434                          71$:


                        ;;**************************************************************
                        ;*TEST 45        TEST DBL (RHCS3 BIT #10) TEST F

                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                        ;*      SET UP FOR A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
                        ;*      DO A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
                        ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
                        ;*      CHECK RHCS3
                        ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
```

```
10544
10545                                   ;;*****************************************************************
031434  000004                  TST45:  SCOPE
031436  012706  001000                  MOV     #STACK,SP       ;RESET STACK
031442  012737  000045  004656          MOV     #45,@#TSTNM     ;SAVE TEST NUMBER
10549
031450  004737  040312'                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10551                                                           ;SETUP UNIT NUBER
10553                                                           ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10553
10554                                   ;SET UP FOR A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
031454  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
031460  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
10557                                                           ;FROM AN ODD WORD BOUNDARY
031464  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
031470  005301                          DEC     R1              ;COUNT
031472  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
031474  012777  003002  152362          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
10562                                                           ;FROM AN ODD WORD BOUNDARY
031502  012777  177765  152352          MOV #-11.,@RHWC ;WORD COUNT ELEVEN
031510  004077  153124                  JSR     R0,@COMND       ;GO TO DO COMMAND
031514  004154                          WRIDAT                  ;WRITE DATA
10566
10567
10568                                   ;CHECK THAT RHCS3 HAS DBL
031516  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
10570
031524  017737  152402  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
031532  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10573                                                           ;DATA WITH DATA READ FROM
10574                                                           ;RHCS3
031540  001401                          BEQ     65$             ;BRANCH IF GOOD
031542  104124                          ERROR   124
10577                                                           ;AFTER SETTING "CLR" BIT #5
10578                                                           ;IN RHCS2 TO INIT THE RH
10579                                                           ;A ELEVEN WORD WRITE FROM AN
10580                                                           ;ODD WORD BOUNDARY WAS DONE
10581                                                           ;THEN
10582                                                           ;RHCS3 SHOULD HAVE DBL
10583                                                           ;=2000
10584                                                           ;BUT CONTAINED WHAT IS
10585                                                           ;GIVEN IN BAD RHCS3
031544                          65$:
031544  042777  000076  152306          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10588                                   ;CHECK THAT RHCS1 HAS RDY!DVA
031552  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10590
031560  017737  152274  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
031566  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10593                                                           ;DATA WITH DATA READ FROM
10594                                                           ;RHCS1
031574  001401                          BEQ     67$             ;BRANCH IF GOOD
031576  104125                          ERROR   125
10597                                                           ;AFTER SETTING "CLR" BIT #5
10598                                                           ;IN RHCS2 TO INIT THE RH
10599                                                           ;A ELEVEN WORD WRITE FROM AN
```

```
10600                                                                   ;ODD WORD BOUNDARY WAS DONE
10601                                                                   ;THEN
10602                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
10603                                                                   ;=4200
10604                                                                   ;BUT CONTAINED WHAT IS
10605                                                                   ;GIVEN IN BAD RHCS1
10606  031600                                67$:
10607                                              ;CHECK THAT RHCS2 HAS IR
10608  031600  012737  000100  001124              MOV     #IR,$GDDAT      ;GET GOOD = 100
10609  031606  053737  005010  001124              BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10610
10611  031614  017737  152250  001126              MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10612  031622  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10613                                                                   ;DATA WITH DATA READ FROM
10614                                                                   ;RHCS2
10615  031630  001401                              BEQ     69$             ;BRANCH IF GOOD
10616  031632  104126                              ERROR   126
10617                                                                   ;AFTER SETTING "CLR" BIT #5
10618                                                                   ;IN RHCS2 TO INIT THE RH
10619                                                                   ;A ELEVEN WORD WRITE FROM AN
10620                                                                   ;ODD WORD BOUNDARY WAS DONE
10621                                                                   ;THEN
10622                                                                   ;RHCS2 SHOULD HAVE IR
10623                                                                   ;=100
10624                                                                   ;TOGETHER WITH UNIT NUMBER
10625                                                                   ;BUT CONTAINED WHAT IS
10626                                                                   ;GIVEN IN BAD RHCS2
10627  031634                                69$:
10628                                              ;CHECK THAT RHWC HAS 0
10629  031634  012737  000000  001124              MOV     #0,$GDDAT       ;GET GOOD = 0
10630
10631  031642  017737  152214  001126              MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10632  031650  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10633                                                                   ;DATA WITH DATA READ FROM
10634                                                                   ;RHWC
10635  031656  001401                              BEQ     71$             ;BRANCH IF GOOD
10636  031660  104131                              ERROR   131
10637                                                                   ;AFTER SETTING "CLR" BIT #5
10638                                                                   ;IN RHCS2 TO INIT THE RH
10639                                                                   ;A ELEVEN WORD WRITE FROM AN
10640                                                                   ;ODD WORD BOUNDARY WAS DONE
10641                                                                   ;THEN
10642                                                                   ;RHWC SHOULD HAVE 0
10643                                                                   ;BUT CONTAINED WHAT IS
10644                                                                   ;GIVEN IN BAD RHWC
10645  031662                                71$:
10646
10647
10648
10649
10650                                  ;;*****************************************************************
10651                                  ;*TEST 46       TEST DBL (RHCS3 BIT #10) TEST G
10652
10653                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10654                                  ;*      SET UP FOR A NINE WORD READ FROM AN EVEN WORD BOUNDARY
10655                                  ;*      DO A NINE WORD READ FROM AN EVEN WORD BOUNDARY
```

```
10656                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10657                                   ;*      CHECK RHCS3
10658                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10659
10660                                   ;;*****************************************************************
10661  031662  000004                   TST46:  SCOPE
10662  031664  012706  001000                   MOV     #STACK,SP       ;RESET STACK
10663  031670  012737  000046  004656           MOV     #46,@#TSTNM     ;SAVE TEST NUMBER
10664
10665  031676  004737  040312                   JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10666                                                                   ;SETUP UNIT NUBER
10667                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10668
10669                                   ;SET UP FOR A NINE WORD READ FROM AN EVEN WORD BOUNDARY
10670  031702  012701  000003                   MOV     #3,R1           ;COUNT OF THREE
10671  031706  012702  003000                   MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10672                                                                   ;FROM AN EVEN WORD BOUNDARY
10673  031712  012722  052525           1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10674  031716  005301                           DEC     R1              ;COUNT
10675  031720  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
10676  031722  012777  003000  152134           MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10677                                                                   ;FROM AN EVEN WORD BOUNDARY
10678  031730  012777  177767  152124           MOV #-9.,@RHWC ;WORD COUNT NINE
10679  031736  004077  152676                   JSR     R0,@COMND       ;GO TO DO COMMAND
10680  031742  004160                           READAT                  ;READ DATA
10681
10682
10683                                   ;CHECK THAT RHCS3 HAS 0
10684  031744  012737  000000  001124           MOV     #0,$GDDAT       ;GET GOOD = 0
10685
10686  031752  017737  152154  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10687  031760  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10688                                                                   ;DATA WITH DATA READ FROM
10689                                                                   ;RHCS3
10690  031766  001401                           BEQ     65$             ;BRANCH IF GOOD
10691  031770  104124                           ERROR   124
10692                                                                   ;AFTER SETTING "CLR" BIT #5
10693                                                                   ;IN RHCS2 TO INIT THE RH
10694                                                                   ;A NINE WORD READ FROM AN
10695                                                                   ;EVEN WORD BOUNDARY WAS DONE
10696                                                                   ;THEN
10697                                                                   ;RHCS3 SHOULD HAVE 0
10698                                                                   ;BUT CONTAINED WHAT IS
10699                                                                   ;GIVEN IN BAD RHCS3
10700  031772                           65$:
10701  031772  042777  000076  152060           BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10702                                   ;CHECK THAT RHCS1 HAS RDY!DVA
10703  032000  012737  004200  001124           MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10704
10705  032006  017737  152046  001126           MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10706  032014  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10707                                                                   ;DATA WITH DATA READ FROM
10708                                                                   ;RHCS1
10709  032022  001401                           BEQ     67$             ;BRANCH IF GOOD
10710  032024  104125                           ERROR   125
10711                                                                   ;AFTER SETTING "CLR" BIT #5
```

```
10712                                                           ;IN RHCS2 TO INIT THE RH
10713                                                           ;A NINE WORD READ FROM AN
10714                                                           ;EVEN WORD BOUNDARY WAS DONE
10715                                                           ;THEN
10716                                                           ;RHCS1 SHOULD HAVE RDY!DVA
10717                                                           ;=4200
10718                                                           ;BUT CONTAINED WHAT IS
10719                                                           ;GIVEN IN BAD RHCS1
10720  032026                           67$:
10721                                          ;CHECK THAT RHCS2 HAS IR
10722  032026  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10723  032034  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10724
10725  032042  017737  152022  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10726  032050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10727                                                                  ;DATA WITH DATA READ FROM
10728                                                                  ;RHCS2
10729  032056  001401                          BEQ     69$             ;BRANCH IF GOOD
10730  032060  104126                          ERROR   126
10731                                                           ;AFTER SETTING "CLR" BIT #5
10732                                                           ;IN RHCS2 TO INIT THE RH
10733                                                           ;A NINE WORD READ FROM AN
10734                                                           ;EVEN WORD BOUNDARY WAS DONE
10735                                                           ;THEN
10736                                                           ;RHCS2 SHOULD HAVE IR
10737                                                           ;=100
10738                                                           ;TOGETHER WITH UNIT NUMBER
10739                                                           ;BUT CONTAINED WHAT IS
10740                                                           ;GIVEN IN BAD RHCS2
10741  032062                           69$:
10742                                          ;CHECK THAT RHWC HAS 0
10743  032062  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10744
10745  032070  017737  151766  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10746  032076  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10747                                                                  ;DATA WITH DATA READ FROM
10748                                                                  ;RHWC
10749  032104  001401                          BEQ     71$             ;BRANCH IF GOOD
10750  032106  104131                          ERROR   131
10751                                                           ;AFTER SETTING "CLR" BIT #5
10752                                                           ;IN RHCS2 TO INIT THE RH
10753                                                           ;A NINE WORD READ FROM AN
10754                                                           ;EVEN WORD BOUNDARY WAS DONE
10755                                                           ;THEN
10756                                                           ;RHWC SHOULD HAVE 0
10757                                                           ;BUT CONTAINED WHAT IS
10758                                                           ;GIVEN IN BAD RHWC
10759  032110                           71$:
10760
10761
10762
10763                                   ;;*****************************************************************
10764                                   ;*TEST 47        TEST DBL (RHCS3 BIT #10) TEST H
10765
10766                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10767                                   ;*      SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
```

```
10768                                   ;*      DO A TEN WORD READ FROM AN EVEN WORD BOUNDARY
10769                                   ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
10770                                   ;*      CHECK RHCS3
10771                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10772
10773                                   ;;*************************************************************
10774  032110  000004                  TST47:  SCOPE
10775  032112  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10776  032116  012737  000047  004656          MOV     #47,@#TSTNM     ;SAVE TEST NUMBER
10777
10778  032124  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10779                                                                  ;SETUP UNIT NUBER
10780                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10781
10782                                   ;SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
10783  032130  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
10784  032134  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10785                                                                  ;FROM AN EVEN WORD BOUNDARY
10786  032140  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10787  032144  005301                          DEC     R1              ;COUNT
10788  032146  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
10789  032150  012777  003000  151706          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10790                                                                  ;FROM AN EVEN WORD BOUNDARY
10791  032156  012777  177766  151676          MOV #-10.,@RHWC ;WORD COUNT TEN
10792  032164  004077  152450                  JSR     R0,@COMND       ;GO TO DO COMMAND
10793  032170  004160                          READAT                  ;READ DATA
10794
10795
10796                                   ;CHECK THAT RHCS3 HAS DBL
10797  032172  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
10798
10799  032200  017737  151726  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10800  032206  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10801                                                                  ;DATA WITH DATA READ FROM
10802                                                                  ;RHCS3
10803  032214  001401                          BEQ     65$             ;BRANCH IF GOOD
10804  032216  104124                          ERROR   124
10805                                                                  ;AFTER SETTING "CLR" BIT #5
10806                                                                  ;IN RHCS2 TO INIT THE RH
10807                                                                  ;A TEN WORD READ FROM AN
10808                                                                  ;EVEN WORD BOUNDARY WAS DONE
10809                                                                  ;THEN
10810                                                                  ;RHCS3 SHOULD HAVE DBL
10811                                                                  ;=2000
10812                                                                  ;BUT CONTAINED WHAT IS
10813                                                                  ;GIVEN IN BAD RHCS3
10814  032220                   65$:
10815  032220  042777  000076  151632          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10816                                   ;CHECK THAT RHCS1 HAS RDY!DVA
10817  032226  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10818
10819  032234  017737  151620  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10820  032242  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10821                                                                  ;DATA WITH DATA READ FROM
10822                                                                  ;RHCS1
10823  032250  001401                          BEQ     67$             ;BRANCH IF GOOD
```

```
10824  032252  104125                          ERROR   125
10825                                                      ;AFTER SETTING "CLR" BIT #5
10826                                                      ;IN RHCS2 TO INIT THE RH
10827                                                      ;A TEN WORD READ FROM AN
10828                                                      ;EVEN WORD BOUNDARY WAS DONE
10829                                                      ;THEN
10830                                                      ;RHCS1 SHOULD HAVE RDY!DVA
10831                                                      ;=4200
10832                                                      ;BUT CONTAINED WHAT IS
10833                                                      ;GIVEN IN BAD RHCS1
10834  032254                          67$:
10835                                      ;CHECK THAT RHCS2 HAS IR
10836  032254  012737  000100  001124      MOV     #IR,$GDDAT      ;GET GOOD = 100
10837  032262  053737  005010  001124      BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10838
10839  032270  017737  151574  001126      MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10840  032276  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10841                                                      ;DATA WITH DATA READ FROM
10842                                                      ;RHCS2
10843  032304  001401                      BEQ     69$             ;BRANCH IF GOOD
10844  032306  104126                      ERROR   126
10845                                                      ;AFTER SETTING "CLR" BIT #5
10846                                                      ;IN RHCS2 TO INIT THE RH
10847                                                      ;A TEN WORD READ FROM AN
10848                                                      ;EVEN WORD BOUNDARY WAS DONE
10849                                                      ;THEN
10850                                                      ;RHCS2 SHOULD HAVE IR
10851                                                      ;=100
10852                                                      ;TOGETHER WITH UNIT NUMBER
10853                                                      ;BUT CONTAINED WHAT IS
10854                                                      ;GIVEN IN BAD RHCS2
10855  032310                          69$:
10856                                      ;CHECK THAT RHWC HAS 0
10857  032310  012737  000000  001124      MOV     #0,$GDDAT       ;GET GOOD = 0
10858
10859  032316  017737  151540  001126      MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10860  032324  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10861                                                      ;DATA WITH DATA READ FROM
10862                                                      ;RHWC
10863  032332  001401                      BEQ     71$             ;BRANCH IF GOOD
10864  032334  104131                      ERROR   131
10865                                                      ;AFTER SETTING "CLR" BIT #5
10866                                                      ;IN RHCS2 TO INIT THE RH
10867                                                      ;A TEN WORD READ FROM AN
10868                                                      ;EVEN WORD BOUNDARY WAS DONE
10869                                                      ;THEN
10870                                                      ;RHWC SHOULD HAVE 0
10871                                                      ;BUT CONTAINED WHAT IS
10872                                                      ;GIVEN IN BAD RHWC
10873  032336                          71$:
10874
10875
10876
10877                          ;;*****************************************************************
10878                          ;*TEST 50        TEST DBL (RHCS3 BIT #10) TEST I
10879
```

```
10880                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10881                                   ;*      SET UP FOR A TEN WORD READ FROM AN ODD WORD BOUNDARY
10882                                   ;*      DO A TEN WORD READ FROM AN ODD WORD BOUNDARY
10883                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10884                                   ;*      CHECK RHCS3
10885                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10886
10887                                   ;;*****************************************************************
10888  032336  000004                 TST50:  SCOPE
10889  032340  012706  001000                 MOV     #STACK,SP       ;RESET STACK
10890  032344  012737  000050  004656         MOV     #50,@#TSTNM     ;SAVE TEST NUMBER
10891
10892  032352  004737  040312                 JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10893                                                                  ;SETUP UNIT NUBER
10894                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10895
10896                                          ;SET UP FOR A TEN WORD READ FROM AN ODD WORD BOUNDARY
10897  032356  012701  000003                 MOV     #3,R1           ;COUNT OF THREE
10898  032362  012702  003002                 MOV     #BFODD,R2       ;START THREE WORD BUFFER
10899                                                                  ;FROM AN ODD WORD BOUNDARY
10900  032366  012722  052525         1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10901  032372  005301                         DEC     R1              ;COUNT
10902  032374  001374                         BNE     1$              ;BRANCH IF THREE NOT DONE
10903  032376  012777  003002  151460         MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
10904                                                                  ;FROM AN ODD WORD BOUNDARY
10905  032404  012777  177766  151450         MOV #-10.,@RHWC ;WORD COUNT TEN
10906  032412  004077  152222                 JSR     R0,@COMND       ;GO TO DO COMMAND
10907  032416  004160                         READAT                  ;READ DATA
10908
10909
10910                                          ;CHECK THAT RHCS3 HAS 0
10911  032420  012737  000000  001124         MOV     #0,$GDDAT       ;GET GOOD = 0
10912
10913  032426  017737  151500  001126         MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10914  032434  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10915                                                                  ;DATA WITH DATA READ FROM
10916                                                                  ;RHCS3
10917  032442  001401                         BEQ     65$             ;BRANCH IF GOOD
10918  032444  104124                         ERROR   124
10919                                                                  ;AFTER SETTING "CLR" BIT #5
10920                                                                  ;IN RHCS2 TO INIT THE RH
10921                                                                  ;A TEN WORD READ FROM AN
10922                                                                  ;ODD WORD BOUNDARY WAS DONE
10923                                                                  ;THEN
10924                                                                  ;RHCS3 SHOULD HAVE 0
10925                                                                  ;BUT CONTAINED WHAT IS
10926                                                                  ;GIVEN IN BAD RHCS3
10927  032446                          65$:
10928  032446  042777  000076  151404         BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10929                                          ;CHECK THAT RHCS1 HAS RDY!DVA
10930  032454  012737  004200  001124         MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10931
10932  032462  017737  151372  001126         MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10933  032470  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10934                                                                  ;DATA WITH DATA READ FROM
10935                                                                  ;RHCS1
```

```
10936  032476  001401                          BEQ     67$             ;BRANCH IF GOOD
10937  032500  104125                          ERROR   125
10938                                                                  ;AFTER SETTING "CLR" BIT #5
10939                                                                  ;IN RHCS2 TO INIT THE RH
10940                                                                  ;A TEN WORD READ FROM AN
10941                                                                  ;ODD WORD BOUNDARY WAS DONE
10942                                                                  ;THEN
10943                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
10944                                                                  ;=4200
10945                                                                  ;BUT CONTAINED WHAT IS
10946                                                                  ;GIVEN IN BAD RHCS1
10947  032502                          67$:
10948                                          ;CHECK THAT RHCS2 HAS IR
10949  032502  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10950  032510  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10951
10952  032516  017737  151346  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10953  032524  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10954                                                                  ;DATA WITH DATA READ FROM
10955                                                                  ;RHCS2
10956  032532  001401                          BEQ     69$             ;BRANCH IF GOOD
10957  032534  104126                          ERROR   126
10958                                                                  ;AFTER SETTING "CLR" BIT #5
10959                                                                  ;IN RHCS2 TO INIT THE RH
10960                                                                  ;A TEN WORD READ FROM AN
10961                                                                  ;ODD WORD BOUNDARY WAS DONE
10962                                                                  ;THEN
10963                                                                  ;RHCS2 SHOULD HAVE IR
10964                                                                  ;=100
10965                                                                  ;TOGETHER WITH UNIT NUMBER
10966                                                                  ;BUT CONTAINED WHAT IS
10967                                                                  ;GIVEN IN BAD RHCS2
10968  032536                          69$:
10969                                          ;CHECK THAT RHWC HAS 0
10970  032536  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10971
10972  032544  017737  151312  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10973  032552  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10974                                                                  ;DATA WITH DATA READ FROM
10975                                                                  ;RHWC
10976  032560  001401                          BEQ     71$             ;BRANCH IF GOOD
10977  032562  104131                          ERROR   131
10978                                                                  ;AFTER SETTING "CLR" BIT #5
10979                                                                  ;IN RHCS2 TO INIT THE RH
10980                                                                  ;A TEN WORD READ FROM AN
10981                                                                  ;ODD WORD BOUNDARY WAS DONE
10982                                                                  ;THEN
10983                                                                  ;RHWC SHOULD HAVE 0
10984                                                                  ;BUT CONTAINED WHAT IS
10985                                                                  ;GIVEN IN BAD RHWC
10986  032564                          71$:
10987
10988
10989
10990                                  ;;*****************************************************************
10991                                  ;*TEST 51        TEST DBL (RHCS3 BIT #10) TEST J
```

```
10992
10993                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10994                                   ;*      SET UP FOR A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
10995                                   ;*      DO A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
10996                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10997                                   ;*      CHECK RHCS3
10998                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10999
11000                                   ;;*****************************************************************
11001  032564  000004                  TST51:  SCOPE
11002  032566  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11003  032572  012737  000051  004656          MOV     #51,@#TSTNM     ;SAVE TEST NUMBER
11004
11005  032600  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11006                                                                  ;SETUP UNIT NUBER
11007                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11008
11009                                   ;SET UP FOR A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
11010  032604  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11011  032610  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11012                                                                  ;FROM AN EVEN WORD BOUNDARY
11013  032614  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11014  032620  005301                          DEC     R1              ;COUNT
11015  032622  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11016  032624  012777  003000  151232          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
11017                                                                  ;FROM AN EVEN WORD BOUNDARY
11018  032632  012777  177765  151222          MOV #-11.,@RHWC ;WORD COUNT ELEVEN
11019  032640  004077  151774                  JSR     R0,@COMND       ;GO TO DO COMMAND
11020  032644  004160                          READAT                  ;READ DATA
11021
11022
11023                                   ;CHECK THAT RHCS3 HAS 0
11024  032646  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11025
11026  032654  017737  151252  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11027  032662  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11028                                                                  ;DATA WITH DATA READ FROM
11029                                                                  ;RHCS3
11030  032670  001401                          BEQ     65$             ;BRANCH IF GOOD
11031  032672  104124                          ERROR   124
11032                                                                  ;AFTER SETTING "CLR" BIT #5
11033                                                                  ;IN RHCS2 TO INIT THE RH
11034                                                                  ;A ELEVEN WORD READ FROM AN
11035                                                                  ;EVEN WORD BOUNDARY WAS DONE
11036                                                                  ;THEN
11037                                                                  ;RHCS3 SHOULD HAVE 0
11038                                                                  ;BUT CONTAINED WHAT IS
11039                                                                  ;GIVEN IN BAD RHCS3
11040  032674                  65$:
11041  032674  042777  000076  151156          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11042                                   ;CHECK THAT RHCS1 HAS RDY!DVA
11043  032702  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
11044
11045  032710  017737  151144  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11046  032716  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11047                                                                  ;DATA WITH DATA READ FROM
```

```
11048
11049  032724  001401                          BEQ     67$             ;RHCS1
11050  032726  104125                          ERROR   125             ;BRANCH IF GOOD
11051                                                                  ;AFTER SETTING "CLR" BIT #5
11052                                                                  ;IN RHCS2 TO INIT THE RH
11053                                                                  ;A ELEVEN WORD READ FROM AN
11054                                                                  ;EVEN WORD BOUNDARY WAS DONE
11055                                                                  ;THEN
11056                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
11057                                                                  ;=4200
11058                                                                  ;BUT CONTAINED WHAT IS
11059                                                                  ;GIVEN IN BAD RHCS1
11060  032730                          67$:
11061                                          ;CHECK THAT RHCS2 HAS IR
11062  032730  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
11063  032736  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11064
11065  032744  017737  151120  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11066  032752  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11067                                                                  ;DATA WITH DATA READ FROM
11068                                                                  ;RHCS2
11069  032760  001401                          BEQ     69$             ;BRANCH IF GOOD
11070  032762  104126                          ERROR   126
11071                                                                  ;AFTER SETTING "CLR" BIT #5
11072                                                                  ;IN RHCS2 TO INIT THE RH
11073                                                                  ;A ELEVEN WORD READ FROM AN
11074                                                                  ;EVEN WORD BOUNDARY WAS DONE
11075                                                                  ;THEN
11076                                                                  ;RHCS2 SHOULD HAVE IR
11077                                                                  ;=100
11078                                                                  ;TOGETHER WITH UNIT NUMBER
11079                                                                  ;BUT CONTAINED WHAT IS
11080                                                                  ;GIVEN IN BAD RHCS2
11081  032764                          69$:
11082                                          ;CHECK THAT RHWC HAS 0
11083  032764  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11084
11085  032772  017737  151064  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11086  033000  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11087                                                                  ;DATA WITH DATA READ FROM
11088                                                                  ;RHWC
11089  033006  001401                          BEQ     71$             ;BRANCH IF GOOD
11090  033010  104131                          ERROR   131
11091                                                                  ;AFTER SETTING "CLR" BIT #5
11092                                                                  ;IN RHCS2 TO INIT THE RH
11093                                                                  ;A ELEVEN WORD READ FROM AN
11094                                                                  ;EVEN WORD BOUNDARY WAS DONE
11095                                                                  ;THEN
11096                                                                  ;RHWC SHOULD HAVE 0
11097                                                                  ;BUT CONTAINED WHAT IS
11098                                                                  ;GIVEN IN BAD RHWC
11099  033012                          71$:
11100
11101
11102
11103                                  ;;*****************************************************************
```

```
11104                                   ;*TEST 52       TEST DBL (RHCS3 BIT #10) TEST K
11105
11106                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11107                                   ;*      SET UP FOR A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
11108                                   ;*      DO A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
11109                                   ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
11110                                   ;*      CHECK RHCS3
11111                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11112
11113                                   ;;*****************************************************************
11114  033012  000004                  TST52:  SCOPE
11115  033014  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11116  033020  012737  000052  004656          MOV     #52,@#TSTNM     ;SAVE TEST NUMBER
11117
11118  033026  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11119                                                                   ;SETUP UNIT NUBER
11120                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11121
11122                                   ;SET UP FOR A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
11123  033032  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11124  033036  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
11125                                                                   ;FROM AN ODD WORD BOUNDARY
11126  033042  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11127  033046  005301                          DEC     R1              ;COUNT
11128  033050  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11129  033052  012777  003002  151004          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
11130                                                                   ;FROM AN ODD WORD BOUNDARY
11131  033060  012777  177765  150774          MOV #-11.,@RHWC ;WORD COUNT ELEVEN
11132  033066  004077  151546                  JSR     R0,@COMND       ;GO TO DO COMMAND
11133  033072  004160                          READAT                  ;READ DATA
11134
11135
11136                                   ;CHECK THAT RHCS3 HAS DBL
11137  033074  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
11138
11139  033102  017737  151024  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11140  033110  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11141                                                                   ;DATA WITH DATA READ FROM
11142                                                                   ;RHCS3
11143  033116  001401                          BEQ     65$             ;BRANCH IF GOOD
11144  033120  104124                          ERROR   124
11145                                                                   ;AFTER SETTING "CLR" BIT #5
11146                                                                   ;IN RHCS2 TO INIT THE RH
11147                                                                   ;A ELEVEN WORD READ FROM AN
11148                                                                   ;ODD WORD BOUNDARY WAS DONE
11149                                                                   ;THEN
11150                                                                   ;RHCS3 SHOULD HAVE DBL
11151                                                                   ;=2000
11152                                                                   ;BUT CONTAINED WHAT IS
11153                                                                   ;GIVEN IN BAD RHCS3
11154  033122                           65$:
11155  033122  042777  000076  150730          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11156                                   ;CHECK THAT RHCS1 HAS RDY!DVA
11157  033130  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
11158
11159  033136  017737  150716  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
```

```
 11160  033144  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 11161                                                                   ;DATA WITH DATA READ FROM
 11162                                                                   ;RHCS1
 11163  033152  001401                          BEQ     67$             ;BRANCH IF GOOD
 11164  033154  104125                          ERROR   125
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A ELEVEN WORD READ FROM AN
                                                                        ;ODD WORD BOUNDARY WAS DONE
                                                                        ;THEN
                                                                        ;RHCS1 SHOULD HAVE RDY!DVA
                                                                        ;=4200
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS1
 11174  033156                          67$:
                                                ;CHECK THAT RHCS2 HAS IR
 11176  033156  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
 11177  033164  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 11178
 11179  033172  017737  150672  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 11180  033200  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 11181                                                                   ;DATA WITH DATA READ FROM
 11182                                                                   ;RHCS2
 11183  033206  001401                          BEQ     69$             ;BRANCH IF GOOD
 11184  033210  104126                          ERROR   126
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A ELEVEN WORD READ FROM AN
                                                                        ;ODD WORD BOUNDARY WAS DONE
                                                                        ;THEN
                                                                        ;RHCS2 SHOULD HAVE IR
                                                                        ;=100
                                                                        ;TOGETHER WITH UNIT NUMBER
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS2
 11196  033212                          69$:
                                                ;CHECK THAT RHWC HAS 0
 11197  033212  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 11198
 11199  033220  017737  150636  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
 11200  033226  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 11201                                                                   ;DATA WITH DATA READ FROM
 11202                                                                   ;RHWC
 11203  033234  001401                          BEQ     71$             ;BRANCH IF GOOD
 11204  033236  104131                          ERROR   131
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A ELEVEN WORD READ FROM AN
                                                                        ;ODD WORD BOUNDARY WAS DONE
                                                                        ;THEN
                                                                        ;RHWC SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHWC
 11213  033240                          71$:
```

```
                                ;;*****************************************************************
                                ;*TEST 53        TEST DBL (RHCS3 BIT #10) TEST L
                                ;*
                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                ;*      SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
                                ;*      WITH BAI IN RHCS2 BIT #3 SET
                                ;*      DO A TEN WORD READ FROM AN EVEN WORD BOUNDARY
                                ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                ;*      CHECK RHCS3
                                ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                ;;*****************************************************************
033240  000004                  ↑ST53:  SCOPE
033242  012706  001000                  MOV     #STACK,SP       ;RESET STACK
033246  012737  000053  004656          MOV     #53,@#↑STNM     ;SAVE TEST NUMBER

033254  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
033260  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
03326   012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
                                                                ;FROM AN EVEN WORD BOUNDARY
033270  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
033274  005301                  DEC     R1              ;COUNT
033276  001374                  BNE     1$              ;BRANCH IF THREE NOT DONE
033300  012777  003000  150556          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
                                                                ;FROM AN EVEN WORD BOUNDARY
033306  012777  177766  150546          MOV #-10.,@RHWC :WORD COUNT TEN
033314  052777  000010  150546  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
033322  004077  151312                  JSR     R0,@COMND       ;GO TO DO COMMAND
033326  004160                  READAT                          ;READ DATA


                                        ;CHECK THAT RHCS3 HAS 0
033330  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

033336  017737  150570  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
033344  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
033352  001401                          BEQ     65$             ;BRANCH IF GOOD
033354  104124                          ERROR   124
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;A TEN WORD READ FROM AN
                                                                ;EVEN WORD BOUNDARY WAS DONE
                                                                ;WITH BAI IN RHCS2 SET
                                                                ;THEN
                                                                ;RHCS3 SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
033356                  65$:
033356  042777  000076  150474          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
```

```
11272                                                   ;CHECK THAT RHCS1 HAS RDY!DVA
11273  033364  012737  004200  001124                   MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
11274
11275  033372  017737  150462  001126                   MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11276  033400  023737  001124  001126                   CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11277                                                                           ;DATA WITH DATA READ FROM
11278                                                                           ;RHCS1
11279  033406  001401                                   BEQ     67$             ;BRANCH IF GOOD
11280  033410  104125                                   ERROR   125
11281                                                                           ;AFTER SETTING "CLR" BIT #5
11282                                                                           ;IN RHCS2 TO INIT THE RH
11283                                                                           ;A TEN WORD READ FROM AN
11284                                                                           ;EVEN WORD BOUNDARY WAS DONE
11285                                                                           ;WITH BAI IN RHCS2 SET
11286                                                                           ;THEN
11287                                                                           ;RHCS1 SHOULD HAVE RDY!DVA
11288                                                                           ;=4200
11289                                                                           ;BUT CONTAINED WHAT IS
11290                                                                           ;GIVEN IN BAD RHCS1
11291  033412                                  67$:
11292                                                   ;CHECK THAT RHCS2 HAS IR!BAI
11293  033412  012737  000110  001124                   MOV     #IR!BAI,$GDDAT  ;GET GOOD = 110
11294  033420  053737  005010  001124                   BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11295
11296  033426  017737  150436  001126                   MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11297  033434  023737  001124  001126                   CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11298                                                                           ;DATA WITH DATA READ FROM
11299                                                                           ;RHCS2
11300  033442  001401                                   BEQ     69$             ;BRANCH IF GOOD
11301  033444  104126                                   ERROR   126
11302                                                                           ;AFTER SETTING "CLR" BIT #5
11303                                                                           ;IN RHCS2 TO INIT THE RH
11304                                                                           ;A TEN WORD READ FROM AN
11305                                                                           ;EVEN WORD BOUNDARY WAS DONE
11306                                                                           ;WITH BAI IN RHCS2 SET
11307                                                                           ;THEN
11308                                                                           ;RHCS2 SHOULD HAVE IR!BAI
11309                                                                           ;=110
11310                                                                           ;TOGETHER WITH UNIT NUMBER
11311                                                                           ;BUT CONTAINED WHAT IS
11312                                                                           ;GIVEN IN BAD RHCS2
11313  033446                                  69$:
11314                                                   ;CHECK THAT RHWC HAS 0
11315  033446  012737  000000  001124                   MOV     #0,$GDDAT       ;GET GOOD = 0
11316
11317  033454  017737  150402  001126                   MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11318  033462  023737  001124  001126                   CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11319                                                                           ;DATA WITH DATA READ FROM
11320                                                                           ;RHWC
11321  033470  001401                                   BEQ     71$             ;BRANCH IF GOOD
11322  033472  104131                                   ERROR   131
11323                                                                           ;AFTER SETTING "CLR" BIT #5
11324                                                                           ;IN RHCS2 TO INIT THE RH
11325                                                                           ;A TEN WORD READ FROM AN
11326                                                                           ;EVEN WORD BOUNDARY WAS DONE
11327                                                                           ;WITH BAI IN RHCS2 SET
```

```
                                                        ;THEN
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
033474                          71$:
                        ;;**************************************************************
                        ;*TEST 54        TEST DBL (RHCS3 BIT #10) TEST M

                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                        ;*      SET UP FOR A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                        ;*      DO A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                        ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                        ;*      CHECK RHCS3
                        ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                        ;;**************************************************************
033474  000004          TST54:  SCOPE
033476  012706  001000          MOV     #STACK,SP       ;RESET STACK
033502  012737  000054  004656  MOV     #54,@#TSTNM     ;SAVE TEST NUMBER

033510  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                        ;SET UP FOR A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
033514  012701  000003          MOV     #3,R1           ;COUNT OF THREE
033520  012702  003000          MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
                                                        ;FROM AN EVEN WORD BOUNDARY
033524  012722  052525  1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
033530  005301                  DEC     R1              ;COUNT
033532  001374                  BNE     1$              ;BRANCH IF THREE NOT DONE
033534  012777  003000  150322  MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
                                                        ;FROM AN EVEN WORD BOUNDARY
033542  012777  177777  150312  MOV #-1,@RHWC ;WORD COUNT ONE
033550  004077  151064          JSR     R0,@COMND       ;GO TO DO COMMAND
033554  004204                  REVWRT                  ;REVERSEWRITE DATA

                        ;CHECK THAT RHCS3 HAS 0
033556  012737  000000  001124  MOV     #0,$GDDAT       ;GET GOOD = 0

033564  017737  150342  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
033572  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
033600  001401                  BEQ     65$             ;BRANCH IF GOOD
033602  104124                  ERROR   124
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A ONE WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;THEN
```

```
11384                                                          ;RHCS3 SHOULD HAVE 0
11385                                                          ;BUT CONTAINED WHAT IS
11386                                                          ;GIVEN IN BAD RHCS3
11387  033604                                  65$:
11388  033604  042777  000076  150246          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11389                                          ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11390  033612  012737  104200  001124          MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11391
11392  033620  017737  150234  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11393  033626  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11394                                                          ;DATA WITH DATA READ FROM
11395                                                          ;RHCS1
11396  033634  001401                          BEQ     67$             ;BRANCH IF GOOD
11397  033636  104125                          ERROR   125
11398                                                          ;AFTER SETTING "CLR" BIT #5
11399                                                          ;IN RHCS2 TO INIT THE RH
11400                                                          ;A ONE WORD WRITE REVERSE FROM AN
11401                                                          ;EVEN WORD BOUNDARY WAS DONE
11402                                                          ;THEN
11403                                                          ;RHCS1 SHOULD HAVE SC!RDY!DVA
11404                                                          ;=104200
11405                                                          ;BUT CONTAINED WHAT IS
11406                                                          ;GIVEN IN BAD RHCS1
11407  033640                                  67$:
11408                                          ;CHECK THAT RHCS2 HAS IR!OR
11409  033640  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11410  033646  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11411
11412  033654  017737  150210  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11413  033662  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11414                                                          ;DATA WITH DATA READ FROM
11415                                                          ;RHCS2
11416  033670  001401                          BEQ     69$             ;BRANCH IF GOOD
11417  033672  104126                          ERROR   126
11418                                                          ;AFTER SETTING "CLR" BIT #5
11419                                                          ;IN RHCS2 TO INIT THE RH
11420                                                          ;A ONE WORD WRITE REVERSE FROM AN
11421                                                          ;EVEN WORD BOUNDARY WAS DONE
11422                                                          ;THEN
11423                                                          ;RHCS2 SHOULD HAVE IR!OR
11424                                                          ;=300
11425                                                          ;TOGETHER WITH UNIT NUMBER
11426                                                          ;BUT CONTAINED WHAT IS
11427                                                          ;GIVEN IN BAD RHCS2
11428  033674                                  69$:
11429                                          ;CHECK THAT RHWC HAS 0
11430  033674  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11431
11432  033702  017737  150154  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11433  033710  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11434                                                          ;DATA WITH DATA READ FROM
11435                                                          ;RHWC
11436  033716  001401                          BEQ     71$             ;BRANCH IF GOOD
11437  033720  104131                          ERROR   131
11438                                                          ;AFTER SETTING "CLR" BIT #5
11439                                                          ;IN RHCS2 TO INIT THE RH
```

```
                                                        ;A ONE WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;THEN
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
033722                          71$:

                        ;;**************************************************************
                        ;*TEST 55       TEST DBL (RHCS3 BIT #10) TEST N

                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                        ;*      SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                        ;*      DO A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                        ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                        ;*      CHECK RHCS3
                        ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                        ;;**************************************************************
033722  000004          TST55:  SCOPE
033724  012706  001000          MOV     #STACK,SP       ;RESET STACK
033730  012737  000055  004656  MOV     #55,@#TSTNM     ;SAVE TEST NUMBER

033736  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                ;SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
033742  012701  000003          MOV     #3,R1           ;COUNT OF THREE
033746  012702  003000          MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
                                                        ;FROM AN EVEN WORD BOUNDARY
033752  012722  052525  1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
033756  005301                  DEC     R1              ;COUNT
033760  001374                  BNE     1$              ;BRANCH IF THREE NOT DONE
033762  012777  003000  150074  MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
                                                        ;FROM AN EVEN WORD BOUNDARY
033770  012777  177776  150064  MOV #-2,@RHWC ;WORD COUNT TWO
033776  004077  150636          JSR     R0,@COMND       ;GO TO DO COMMAND
034002  004204                  REVWRT                  ;REVERSEWRITE DATA

                                ;CHECK THAT RHCS3 HAS 0
034004  012737  000000  001124  MOV     #0,$GDDAT       ;GET GOOD = 0

034012  017737  150114  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
034020  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
034026  001401                  BEQ     65$             ;BRANCH IF GOOD
034030  104124                  ERROR   124

                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TWO WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
```

```
11496                                                           ;THEN
11497                                                           ;RHCS3 SHOULD HAVE 0
11498                                                           ;BUT CONTAINED WHAT IS
11499                                                           ;GIVEN IN BAD RHCS3
11500  034032                                65$:
11501  034032  042777  000076  150020              BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11502                                              ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11503  034040  012737  104200  001124              MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11504
11505  034046  017737  150006  001126              MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11506  034054  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11507                                                                       ;DATA WITH DATA READ FROM
11508                                                                       ;RHCS1
11509  034062  001401                              BEQ     67$             ;BRANCH IF GOOD
11510  034064  104125                              ERROR   125
11511                                                           ;AFTER SETTING "CLR" BIT #5
11512                                                           ;IN RHCS2 TO INIT THE RH
11513                                                           ;A TWO WORD WRITE REVERSE FROM AN
11514                                                           ;EVEN WORD BOUNDARY WAS DONE
11515                                                           ;THEN
11516                                                           ;RHCS1 SHOULD HAVE SC!RDY!DVA
11517                                                           ;=104200
11518                                                           ;BUT CONTAINED WHAT IS
11519                                                           ;GIVEN IN BAD RHCS1
11520  034066                                67$:
11521                                              ;CHECK THAT RHCS2 HAS IR!OR
11522  034066  012737  000300  001124              MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11523  034074  053737  005010  001124              BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11524
11525  034102  017737  147762  001126              MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11526  034110  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11527                                                                       ;DATA WITH DATA READ FROM
11528                                                                       ;RHCS2
11529  034116  001401                              BEQ     69$             ;BRANCH IF GOOD
11530  034120  104126                              ERROR   126
11531                                                           ;AFTER SETTING "CLR" BIT #5
11532                                                           ;IN RHCS2 TO INIT THE RH
11533                                                           ;A TWO WORD WRITE REVERSE FROM AN
11534                                                           ;EVEN WORD BOUNDARY WAS DONE
11535                                                           ;THEN
11536                                                           ;RHCS2 SHOULD HAVE IR!OR
11537                                                           ;=300
11538                                                           ;TOGETHER WITH UNIT NUMBER
11539                                                           ;BUT CONTAINED WHAT IS
11540                                                           ;GIVEN IN BAD RHCS2
11541  034122                                69$:
11542                                              ;CHECK THAT RHWC HAS 0
11543  034122  012737  000000  001124              MOV     #0,$GDDAT       ;GET GOOD = 0
11544
11545  034130  017737  147726  001126              MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11546  034136  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11547                                                                       ;DATA WITH DATA READ FROM
11548                                                                       ;RHWC
11549  034144  001401                              BEQ     71$             ;BRANCH IF GOOD
11550  034146  104131                              ERROR   131
11551                                                           ;AFTER SETTING "CLR" BIT #5
```

```
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TWO WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;THEN
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
034150                          71$:

                        ;;**************************************************************
                        ;*TEST 56       TEST DBL (RHCS3 BIT #10) TEST 0

                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                        ;*      SET UP FOR A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
                        ;*      DO A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
                        ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
                        ;*      CHECK RHCS3
                        ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                        ;;**************************************************************
034150  000004          TST56:  SCOPE
034152  012706  001000          MOV     #STACK,SP       ;RESET STACK
034156  012737  000056  004656  MOV     #56,@#TSTNM     ;SAVE TEST NUMBER

034164  004737  040312          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                        ;SET UP FOR A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
034170  012701  000003          MOV     #3,R1           ;COUNT OF THREE
034174  012702  003002          MOV     #BFODD,R2       ;START THREE WORD BUFFER
                                                        ;FROM AN ODD WORD BOUNDARY
034200  012722  052525  1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
034204  005301                  DEC     R1              ;COUNT
034206  001374                  BNE     1$              ;BRANCH IF THREE NOT DONE
034210  012777  003002  147646  MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
                                                        ;FROM AN ODD WORD BOUNDARY
034216  012777  177776  147636  MOV #-2,@RHWC ;WORD COUNT TWO
034224  004077  150410          JSR     R0,@COMND       ;GO TO DO COMMAND
034230  004204                  REVWRT                  ;REVERSEWRITE DATA

                        ;CHECK THAT RHCS3 HAS DBL
034232  012737  002000  001124  MOV     #DBL,$GDDAT     ;GET GOOD = 2000

034240  017737  147666  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
034246  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
034254  001401                  BEQ     65$             ;BRANCH IF GOOD
034256  104124                  ERROR   124

                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TWO WORD WRITE REVERSE FROM AN
```

```
11608                                                                   ;ODD WORD BOUNDARY WAS DONE
11609                                                                   ;THEN
11610                                                                   ;RHCS3 SHOULD HAVE DBL
11611                                                                   ;=2000
11612                                                                   ;BUT CONTAINED WHAT IS
11613                                                                   ;GIVEN IN BAD RHCS3
11614  034260                                  65$:
11615  034260  042777  000076  147572                BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11616                                                ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11617  034266  012737  104200  001124                MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11618
11619  034274  017737  147560  001126                MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11620  034302  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11621                                                                   ;DATA WITH DATA READ FROM
11622                                                                   ;RHCS1
11623  034310  001401                                BEQ     67$             ;BRANCH IF GOOD
11624  034312  104125                                ERROR   125
11625                                                                   ;AFTER SETTING "CLR" BIT #5
11626                                                                   ;IN RHCS2 TO INIT THE RH
11627                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11628                                                                   ;ODD WORD BOUNDARY WAS DONE
11629                                                                   ;THEN
11630                                                                   ;RHCS1 SHOULD HAVE SC!RDY!DVA
11631                                                                   ;=104200
11632                                                                   ;BUT CONTAINED WHAT IS
11633                                                                   ;GIVEN IN BAD RHCS1
11634  034314                                  67$:
11635                                                ;CHECK THAT RHCS2 HAS IR!OR
11636  034314  012737  000300  001124                MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11637  034322  053737  005010  001124                BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11638
11639  034330  017737  147534  001126                MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11640  034336  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11641                                                                   ;DATA WITH DATA READ FROM
11642                                                                   ;RHCS2
11643  034344  001401                                BEQ     69$             ;BRANCH IF GOOD
11644  034346  104126                                ERROR   126
11645                                                                   ;AFTER SETTING "CLR" BIT #5
11646                                                                   ;IN RHCS2 TO INIT THE RH
11647                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11648                                                                   ;ODD WORD BOUNDARY WAS DONE
11649                                                                   ;THEN
11650                                                                   ;RHCS2 SHOULD HAVE IR!OR
11651                                                                   ;=300
11652                                                                   ;TOGETHER WITH UNIT NUMBER
11653                                                                   ;BUT CONTAINED WHAT IS
11654                                                                   ;GIVEN IN BAD RHCS2
11655  034350                                  69$:
11656                                                ;CHECK THAT RHWC HAS 0
11657  034350  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0
11658
11659  034356  017737  147500  001126                MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11660  034364  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11661                                                                   ;DATA WITH DATA READ FROM
11662                                                                   ;RHWC
11663  034372  001401                                BEQ     71$             ;BRANCH IF GOOD
```

```
11664  034374  104131                          ERROR   131
11665                                                   ;AFTER SETTING "CLR" BIT #5
11666                                                   ;IN RHCS2 TO INIT THE RH
11667                                                   ;A TWO WORD WRITE REVERSE FROM AN
11668                                                   ;ODD WORD BOUNDARY WAS DONE
11669                                                   ;THEN
11670                                                   ;RHWC SHOULD HAVE 0
11671                                                   ;BUT CONTAINED WHAT IS
11672                                                   ;GIVEN IN BAD RHWC
11673  034376                          71$:
11674
11675
11676
11677                                   ;;*****************************************************************
11678                                   ;*TEST 57       TEST DBL (RHCS3 BIT #10) TEST P
11679
11680                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11681                                   ;*      SET UP FOR A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11682                                   ;*      DO A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11683                                   ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
11684                                   ;*      CHECK RHCS3
11685                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11686
11687                                   ;;*****************************************************************
11688  034376  000004                   TST57:  SCOPE
11689  034400  012706  001000                   MOV     #STACK,SP       ;RESET STACK
11690  034404  012737  000057  004656           MOV     #57,@#TSTNM     ;SAVE TEST NUMBER
11691
11692  034412  004737  040312                   JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11693                                                                   ;SETUP UNIT NUBER
11694                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11695
11696                                           ;SET UP FOR A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11697  034416  012701  000003                   MOV     #3,R1           ;COUNT OF THREE
11698  034422  012702  003000                   MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11699                                                                   ;FROM AN EVEN WORD BOUNDARY
11700  034426  012722  052525           1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11701  034432  005301                           DEC     R1              ;COUNT
11702  034434  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
11703  034436  012777  003000  147420           MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
11704                                                                   ;FROM AN EVEN WORD BOUNDARY
11705  034444  012777  177775  147410           MOV #-3,@RHWC ;WORD COUNT THREE
11706  034452  004077  150162                   JSR     R0,@COMND       ;GO TO DO COMMAND
11707  034456  004204                           REVWRT                  ;REVERSEWRITE DATA
11708
11709
11710                                           ;CHECK THAT RHCS3 HAS DBL
11711  034460  012737  002000  001124           MOV     #DBL,$GDDAT     ;GET GOOD = 2000
11712
11713  034466  017737  147440  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11714  034474  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11715                                                                   ;DATA WITH DATA READ FROM
11716                                                                   ;RHCS3
11717  034502  001401                           BEQ     65$             ;BRANCH IF GOOD
11718  034504  104124                           ERROR   124
11719                                                                   ;AFTER SETTING "CLR" BIT #5
```

```
11720                                                           ;IN RHCS2 TO INIT THE RH
11721                                                           ;A THREE WORD WRITE REVERSE FROM AN
11722                                                           ;EVEN WORD BOUNDARY WAS DONE
11723                                                           ;THEN
11724                                                           ;RHCS3 SHOULD HAVE DBL
11725                                                           ;=2000
11726                                                           ;BUT CONTAINED WHAT IS
11727                                                           ;GIVEN IN BAD RHCS3
11728  034506                                65$:
11729  034506  042777  000076  147344          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11730                                          ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11731  034514  012737  104200  001124          MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11732
11733  034522  017737  147332  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11734  034530  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11735                                                                  ;DATA WITH DATA READ FROM
11736                                                                  ;RHCS1
11737  034536  001401                          BEQ     67$             ;BRANCH IF GOOD
11738  034540  104125                          ERROR   125
11739                                                           ;AFTER SETTING "CLR" BIT #5
11740                                                           ;IN RHCS2 TO INIT THE RH
11741                                                           ;A THREE WORD WRITE REVERSE FROM AN
11742                                                           ;EVEN WORD BOUNDARY WAS DONE
11743                                                           ;THEN
11744                                                           ;RHCS1 SHOULD HAVE SC!RDY!DVA
11745                                                           ;=104200
11746                                                           ;BUT CONTAINED WHAT IS
11747                                                           ;GIVEN IN BAD RHCS1
11748  034542                                67$:
11749                                          ;CHECK THAT RHCS2 HAS IR!OR
11750  034542  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11751  034550  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11752
11753  034556  017737  147306  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11754  034564  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11755                                                                  ;DATA WITH DATA READ FROM
11756                                                                  ;RHCS2
11757  034572  001401                          BEQ     69$             ;BRANCH IF GOOD
11758  034574  104126                          ERROR   126
11759                                                           ;AFTER SETTING "CLR" BIT #5
11760                                                           ;IN RHCS2 TO INIT THE RH
11761                                                           ;A THREE WORD WRITE REVERSE FROM AN
11762                                                           ;EVEN WORD BOUNDARY WAS DONE
11763                                                           ;THEN
11764                                                           ;RHCS2 SHOULD HAVE IR!OR
11765                                                           ;=300
11766                                                           ;TOGETHER WITH UNIT NUMBER
11767                                                           ;BUT CONTAINED WHAT IS
11768                                                           ;GIVEN IN BAD RHCS2
11769  034576                                69$:
11770                                          ;CHECK THAT RHWC HAS 0
11771  034576  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11772
11773  034604  017737  147252  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11774  034612  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11775                                                                  ;DATA WITH DATA READ FROM
```

```
11776
11777  034620  001401                          BEQ     71$             ;RHWC
11778  034622  104131                          ERROR   131             ;BRANCH IF GOOD
11779
11780                                                                   ;AFTER SETTING "CLR" BIT #5
11781                                                                   ;IN RHCS2 TO INIT THE RH
11782                                                                   ;A THREE WORD WRITE REVERSE FROM AN
11783                                                                   ;EVEN WORD BOUNDARY WAS DONE
11784                                                                   ;THEN
11785                                                                   ;RHWC SHOULD HAVE 0
11786                                                                   ;BUT CONTAINED WHAT IS
11787  034624                          71$:                            ;GIVEN IN BAD RHWC
11788
11789
11790
11791                                   ;;*****************************************************************
11792                                   ;*TEST 60       TEST DBL (RHCS3 BIT #10) TEST Q
11793
11794                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11795                                   ;*      SET UP FOR A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11796                                   ;*      DO A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11797                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
11798                                   ;*      CHECK RHCS3
11799                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11800
11801                                   ;;*****************************************************************
11802  034624  000004                   TST60:  SCOPE
11803  034626  012706  001000                   MOV     #STACK,SP       ;RESET STACK
11804  034632  012737  000060  004656           MOV     #60,@#TSTNM     ;SAVE TEST NUMBER
11805
11806  034640  004737  040312                   JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11807                                                                   ;SETUP UNIT NUBER
11808                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11809
11810                                           ;SET UP FOR A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11811  034644  012701  000003                   MOV     #3,R1           ;COUNT OF THREE
11812  034650  012702  003002                   MOV     #BFODD,R2       ;START THREE WORD BUFFER
11813                                                                   ;FROM AN ODD WORD BOUNDARY
11814  034654  012722  052525           1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11815  034660  005301                           DEC     R1              ;COUNT
11816  034662  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
11817  034664  012777  003002  147172           MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
11818                                                                   ;FROM AN ODD WORD BOUNDARY
11819  034672  012777  177775  147162           MOV #-3,@RHWC ;WORD COUNT THREE
11820  034700  004077  147734                   JSR     R0,@COMND       ;GO TO DO COMMAND
11821  034704  004204                           REVWRT                  ;REVERSEWRITE DATA
11822
11823
11824                                           ;CHECK THAT RHCS3 HAS 0
11825  034706  012737  000000  001124           MOV     #0,$GDDAT       ;GET GOOD = 0
11826
11827  034714  017737  147212  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11828  034722  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11829                                                                   ;DATA WITH DATA READ FROM
11830                                                                   ;RHCS3
11831  034730  001401                           BEQ     65$             ;BRANCH IF GOOD
```

```
11832  034732  104124                          ERROR   124
11833                                                  ;AFTER SETTING "CLR" BIT #5
11834                                                  ;IN RHCS2 TO INIT THE RH
11835                                                  ;A THREE WORD WRITE REVERSE FROM AN
11836                                                  ;ODD WORD BOUNDARY WAS DONE
11837                                                  ;THEN
11838                                                  ;RHCS3 SHOULD HAVE 0
11839                                                  ;BUT CONTAINED WHAT IS
11840                                                  ;GIVEN IN BAD RHCS3
11841  034734                          65$:
11842  034734  042777  000076  147116          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11843                                         ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11844  034742  012737  104200  001124          MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11845
11846  034750  017737  147104  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11847  034756  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11848                                                  ;DATA WITH DATA READ FROM
11849                                                  ;RHCS1
11850  034764  001401                          BEQ     67$             ;BRANCH IF GOOD
11851  034766  104125                          ERROR   125
11852                                                  ;AFTER SETTING "CLR" BIT #5
11853                                                  ;IN RHCS2 TO INIT THE RH
11854                                                  ;A THREE WORD WRITE REVERSE FROM AN
11855                                                  ;ODD WORD BOUNDARY WAS DONE
11856                                                  ;THEN
11857                                                  ;RHCS1 SHOULD HAVE SC!RDY!DVA
11858                                                  ;=104200
11859                                                  ;BUT CONTAINED WHAT IS
11860                                                  ;GIVEN IN BAD RHCS1
11861  034770                          67$:
11862                                         ;CHECK THAT RHCS2 HAS IR!OR
11863  034770  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11864  034776  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11865
11866  035004  017737  147060  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11867  035012  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11868                                                  ;DATA WITH DATA READ FROM
11869                                                  ;RHCS2
11870  035020  001401                          BEQ     69$             ;BRANCH IF GOOD
11871  035022  104126                          ERROR   126
11872                                                  ;AFTER SETTING "CLR" BIT #5
11873                                                  ;IN RHCS2 TO INIT THE RH
11874                                                  ;A THREE WORD WRITE REVERSE FROM AN
11875                                                  ;ODD WORD BOUNDARY WAS DONE
11876                                                  ;THEN
11877                                                  ;RHCS2 SHOULD HAVE IR!OR
11878                                                  ;=300
11879                                                  ;TOGETHER WITH UNIT NUMBER
11880                                                  ;BUT CONTAINED WHAT IS
11881                                                  ;GIVEN IN BAD RHCS2
11882  035024                          69$:
11883                                         ;CHECK THAT RHWC HAS 0
11884  035024  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11885
11886  035032  017737  147024  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11887  035040  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
035046  001401                      BEQ     71$           ;BRANCH IF GOOD
035050  104131                      ERROR   131

                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A THREE WORD WRITE REVERSE FROM AN
                                                        ;ODD WORD BOUNDARY WAS DONE
                                                        ;THEN
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
035052                      71$:


                            ;;*****************************************************************
                            ;*TEST 61       TEST DBL (RHCS3 BIT #10) TEST R

                            ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                            ;*      SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                            ;*      WITH BAI IN RHCS2 BIT #3 SET
                            ;*      DO A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                            ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                            ;*      CHECK RHCS3
                            ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                            ;;*****************************************************************
035052  000004            ↑ST61:  SCOPE
035054  012706  001000            MOV     #STACK,SP       ;RESET STACK
035060  012737  000061  004656    MOV     #61,@#↑STNM     ;SAVE TEST NUMBER

035066  004737  040312            JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                    ;SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
035072  012701  000003            MOV     #3,R1           ;COUNT OF THREE
035076  012702  003000            MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
                                                        ;FROM AN EVEN WORD BOUNDARY
035102  012722  052525    1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
035106  005301                    DEC     R1              ;COUNT
035110  001374                    BNE     1$              ;BRANCH IF THREE NOT DONE
035112  012777  003000  146744    MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
                                                        ;FROM AN EVEN WORD BOUNDARY
035120  012777  177776  146734    MOV #-2,@RHWC ;WORD COUNT TWO
035126  052777  000010  146734  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
035134  004077  147500            JSR     R0,@COMND       ;GO TO DO COMMAND
035140  004204                    REVWRT                  ;REVERSEWRITE DATA


                                    ;CHECK THAT RHCS3 HAS 0
035142  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD = 0

035150  017737  146756  001126    MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
035156  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
035164  001401                    BEQ     65$         ;BRANCH IF GOOD.
035166  104124                    ERROR   124
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TWO WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;WITH BAI IN RHCS2 SET
                                                        ;THEN
                                                        ;RHCS3 SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS3
035170                      65$:
035170  042777  000076  146662    BIC     #76,@RHCS1  ;CLEAR FUNCTION BITS
                        ;CHECK THAT RHCS1 HAS SC!RDY!DVA
035176  012737  104200  001124    MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200

035204  017737  146650  001126    MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
035212  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
035220  001401                    BEQ     67$             ;BRANCH IF GOOD
035222  104125                    ERROR   125
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TWO WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;WITH BAI IN RHCS2 SET
                                                        ;THEN
                                                        ;RHCS1 SHOULD HAVE SC!RDY!DVA
                                                        ;=104200
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS1
035224                      67$:
                        ;CHECK THAT RHCS2 HAS IR!OR!BAI
035224  012737  000310  001124    MOV     #IR!OR!BAI,$GDDAT       ;GET GOOD = 310
035232  053737  005010  001124    BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

035240  017737  146624  001126    MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
035246  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
035254  001401                    BEQ     69$             ;BRANCH IF GOOD
035256  104126                    ERROR   126
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TWO WORD WRITE REVERSE FROM AN
                                                        ;EVEN WORD BOUNDARY WAS DONE
                                                        ;WITH BAI IN RHCS2 SET
                                                        ;THEN
                                                        ;RHCS2 SHOULD HAVE IR!OR!BAI
                                                        ;=310
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
```

```
12000  035260                          69$:
12001                                          ;CHECK THAT RHWC HAS 0
12002  035260  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12003
12004  035266  017737  146570  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12005  035274  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12006                                                                  ;DATA WITH DATA READ FROM
12007                                                                  ;RHWC
12008  035302  001401                          BEQ     71$             ;BRANCH IF GOOD
12009  035304  104131                          ERROR   131
12010                                                                  ;AFTER SETTING "CLR" BIT #5
12011          . .                                                     ;IN RHCS2 TO INIT THE RH
12012                                                                  ;A TWO WORD WRITE REVERSE FROM AN
12013                                                                  ;EVEN WORD BOUNDARY WAS DONE
12014                                                                  ;WITH BAI IN RHCS2 SET
12015                                                                  ;THEN
12016                                                                  ;RHWC SHOULD HAVE 0
12017                                                                  ;BUT CONTAINED WHAT IS
12018                                                                  ;GIVEN IN BAD RHWC
12019  035306                          71$:
12020
12021
12022
12023
12024                                  ;;*****************************************************************
12025                                  ;*TEST 62       TEST DBL (RHCS3 BIT #10) TEST S
12026
12027                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12028                                  ;*      SET UP FOR A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12029                                  ;*      DO A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12030                                  ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
12031                                  ;*      CHECK RHCS3
12032                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12033                                  ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12034
12035                                  ;;*****************************************************************
12036  035306  000004                  TST62:  SCOPE
12037  035310  012706  001000                  MOV     #STACK,SP       ;RESET STACK
12038  035314  012737  000062  004656          MOV     #62,@#TSTNM     ;SAVE TEST NUMBER
12039
12040  035322  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12041                                                                  ;SETUP UNIT NUBER
12042                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12043  035326  022737  041710  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
12044
12045  035334  001103                          BNE     TST63   ;BRANCH IF TU NOT THERE
12046
12047
12048                                          ;SET UP FOR A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12049  035336  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12050  035342  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
12051                                                                  ;FROM AN EVEN WORD BOUNDARY
12052  035346  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12053  035352  005301                          DEC     R1              ;COUNT
12054  035354  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12055  035356  012777  003000  146500          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
```

```
12056                                                           ;FROM AN EVEN WORD BOUNDARY
12057  035364  012777  177767  146470          MOV #-9.,@RHWC ;WORD COUNT NINE
12058
12059  035372  004077  147242                  JSR     R0,@COMND       ;GO TO DO COMMAND
12060  035376  004200                          REVRED                  ;REVERSE READ
12061
12062                                          ;CHECK THAT RHCS3 HAS DBL
12063  035400  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
12064
12065  035406  017737  146520  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12066  035414  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12067                                                                  ;DATA WITH DATA READ FROM
12068                                                                  ;RHCS3
12069  035422  001401                          BEQ     65$             ;BRANCH IF GOOD
12070  035424  104124                          ERROR   124
12071                                                                  ;AFTER SETTING "CLR" BIT #5
12072                                                                  ;IN RHCS2 TO INIT THE RH
12073                                                                  ;A NINE WORD READ REVERSE FROM AN
12074                                                                  ;EVEN WORD BOUNDARY WAS DONE
12075                                                                  ;THEN
12076                                                                  ;RHCS3 SHOULD HAVE DBL
12077                                                                  ;=2000
12078                                                                  ;BUT CONTAINED WHAT IS
12079                                                                  ;GIVEN IN BAD RHCS3
12080  035426                          65$:
12081  035426  042777  000076  146424          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12082                                          ;CHECK THAT RHCS1 HAS RDY!DVA
12083  035434  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12084
12085  035442  017737  146412  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12086  035450  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12087                                                                  ;DATA WITH DATA READ FROM
12088                                                                  ;RHCS1
12089  035456  001401                          BEQ     67$             ;BRANCH IF GOOD
12090  035460  104125                          ERROR   125
12091                                                                  ;AFTER SETTING "CLR" BIT #5
12092                                                                  ;IN RHCS2 TO INIT THE RH
12093                                                                  ;A NINE WORD READ REVERSE FROM AN
12094                                                                  ;EVEN WORD BOUNDARY WAS DONE
12095                                                                  ;THEN
12096                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
12097                                                                  ;=4200
12098                                                                  ;BUT CONTAINED WHAT IS
12099                                                                  ;GIVEN IN BAD RHCS1
12100  035462                          67$:
12101                                          ;CHECK THAT RHCS2 HAS IR
12102  035462  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12103  035470  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12104
12105  035476  017737  146366  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12106  035504  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12107                                                                  ;DATA WITH DATA READ FROM
12108                                                                  ;RHCS2
12109  035512  001401                          BEQ     69$             ;BRANCH IF GOOD
12110  035514  104126                          ERROR   126
12111                                                                  ;AFTER SETTING "CLR" BIT #5
```

```
12112                                                                   ;IN RHCS2 TO INIT THE RH
12113                                                                   ;A NINE WORD READ REVERSE FROM AN
12114                                                                   ;EVEN WORD BOUNDARY WAS DONE
12115                                                                   ;THEN
12116                                                                   ;RHCS2 SHOULD HAVE IR
12117                                                                   ;=100
12118                                                                   ;TOGETHER WITH UNIT NUMBER
12119                                                                   ;BUT CONTAINED WHAT IS
12120                                                                   ;GIVEN IN BAD RHCS2
12121  035516                                  69$:
12122                                          ;CHECK THAT RHWC HAS 0
12123  035516  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12124
12125  035524  017737  146332  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12126  035532  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12127                                                                  ;DATA WITH DATA READ FROM
12128                                                                  ;RHWC
12129  035540  001401                          BEQ     71$             ;BRANCH IF GOOD
12130  035542  104131                          ERROR   131
12131                                                                  ;AFTER SETTING "CLR" BIT #5
12132                                                                  ;IN RHCS2 TO INIT THE RH
12133                                                                  ;A NINE WORD READ REVERSE FROM AN
12134                                                                  ;EVEN WORD BOUNDARY WAS DONE
12135                                                                  ;THEN
12136                                                                  ;RHWC SHOULD HAVE 0
12137                                                                  ;BUT CONTAINED WHAT IS
12138                                                                  ;GIVEN IN BAD RHWC
12139  035544                                  71$:
12140
12141
12142
12143                                          ;;*****************************************************************
12144                                          ;*TEST 63       TEST DBL (RHCS3 BIT #10) TEST T
12145
12146                                          ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12147                                          ;*      SET UP FOR A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12148                                          ;*      DO A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12149                                          ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
12150                                          ;*      CHECK RHCS3
12151                                          ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12152                                          ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12153
12154                                          ;;*****************************************************************
12155  035544  000004                          TST63:  SCOPE
12156  035546  012706  001000                          MOV     #STACK,SP       ;RESET STACK
12157  035552  012737  000063  004656                  MOV     #63,@#TSTNM     ;SAVE TEST NUMBER
12158
12159  035560  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12160                                                                  ;SETUP UNIT NUBER
12161                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12162  035564  022737  041710  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
12163
12164  035572  001103                          BNE     TST64   ;BRANCH IF TU NOT THERE
12165
12166
12167                                          ;SET UP FOR A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
```

```
12168  035574  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12169  035600  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
12170                                                                  ;FROM AN EVEN WORD BOUNDARY
12171  035604  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12172  035610  005301                          DEC     R1              ;COUNT
12173  035612  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12174  035614  012777  003000  146242          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
12175                                                                  ;FROM AN EVEN WORD BOUNDARY
12176  035622  012777  177766  146232          MOV #-10.,@RHWC ;WORD COUNT TEN
12177
12178  035630  004077  147004                  JSR     R0,@COMND       ;GO TO DO COMMAND
12179  035634  004200                          REVRED                  ;REVERSE READ
12180
12181                                          ;CHECK THAT RHCS3 HAS 0
12182  035636  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12183
12184  035644  017737  146262  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12185  035652  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12186                                                                  ;DATA WITH DATA READ FROM
12187                                                                  ;RHCS3
12188  035660  001401                          BEQ     65$             ;BRANCH IF GOOD
12189  035662  104124                          ERROR   124
12190                                                                  ;AFTER SETTING "CLR" BIT #5
12191                                                                  ;IN RHCS2 TO INIT THE RH
12192                                                                  ;A TEN WORD READ REVERSE FROM AN
12193                                                                  ;EVEN WORD BOUNDARY WAS DONE
12194                                                                  ;THEN
12195                                                                  ;RHCS3 SHOULD HAVE 0
12196                                                                  ;BUT CONTAINED WHAT IS
12197                                                                  ;GIVEN IN BAD RHCS3
12198  035664                          65$:
12199  035664  042777  000076  146166          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12200                                          ;CHECK THAT RHCS1 HAS RDY!DVA
12201  035672  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12202
12203  035700  017737  146154  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12204  035706  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12205                                                                  ;DATA WITH DATA READ FROM
12206                                                                  ;RHCS1
12207  035714  001401                          BEQ     67$             ;BRANCH IF GOOD
12208  035716  104125                          ERROR   125
12209                                                                  ;AFTER SETTING "CLR" BIT #5
12210                                                                  ;IN RHCS2 TO INIT THE RH
12211                                                                  ;A TEN WORD READ REVERSE FROM AN
12212                                                                  ;EVEN WORD BOUNDARY WAS DONE
12213                                                                  ;THEN
12214                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
12215                                                                  ;=4200
12216                                                                  ;BUT CONTAINED WHAT IS
12217                                                                  ;GIVEN IN BAD RHCS1
12218  035720                          67$:
12219                                          ;CHECK THAT RHCS2 HAS IR
12220  035720  012[illegible]  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12221  035726  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12222
12223  035734  017737  146130  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
```

```
12224  035742  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12225                                                                   ;DATA WITH DATA READ FROM
12226                                                                   ;RHCS2
12227  035750  001401                          BEQ     69$             ;BRANCH IF GOOD
12228  035752  104126                          ERROR   126
12229                                                                   ;AFTER SETTING "CLR" BIT #5
12230                                                                   ;IN RHCS2 TO INIT THE RH
12231                                                                   ;A TEN WORD READ REVERSE FROM AN
12232                                                                   ;EVEN WORD BOUNDARY WAS DONE
12233                                                                   ;THEN
12234                                                                   ;RHCS2 SHOULD HAVE IR
12235                                                                   ;=100
12236                                                                   ;TOGETHER WITH UNIT NUMBER
12237                                                                   ;BUT CONTAINED WHAT IS
12238                                                                   ;GIVEN IN BAD RHCS2
12239  035754                          69$:
12240                                          ;CHECK THAT RHWC HAS 0
12241  035754  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12242
12243  035762  017737  146074  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12244  035770  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12245                                                                   ;DATA WITH DATA READ FROM
12246                                                                   ;RHWC
12247  035776  001401                          BEQ     71$             ;BRANCH IF GOOD
12248  036000  104131                          ERROR   131
12249                                                                   ;AFTER SETTING "CLR" BIT #5
12250                                                                   ;IN RHCS2 TO INIT THE RH
12251                                                                   ;A TEN WORD READ REVERSE FROM AN
12252                                                                   ;EVEN WORD BOUNDARY WAS DONE
12253                                                                   ;THEN
12254                                                                   ;RHWC SHOULD HAVE 0
12255                                                                   ;BUT CONTAINED WHAT IS
12256                                                                   ;GIVEN IN BAD RHWC
12257  036002                          71$:
12258
12259
12260
12261                                  ;;*****************************************************************
12262                                  ;*TEST 64        TEST DBL (RHCS3 BIT #10) TEST U
12263
12264                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12265                                  ;*      SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12266                                  ;*      DO A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12267                                  ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
12268                                  ;*      CHECK RHCS3
12269                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12270                                  ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12271
12272                                  ;;*****************************************************************
12273  036002  000004                  TST64:  SCOPE
12274  036004  012706  001000                  MOV     #STACK,SP       ;RESET STACK
12275  036010  012737  000064  004656          MOV     #64,@#TSTNM     ;SAVE TEST NUMBER
12276
12277  036016  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12278                                                                   ;SETUP UNIT NUBER
12279                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
```

```
12280  036022  022737  041710  004640          CMP     #CMNDTM,COMND  ;IS TU THERE
12281
12282  036030  001103                          BNE     TST65   ;BRANCH IF TU NOT THERE
12283
12284
12285                                          ;SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12286  036032  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12287  036036  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
12288                                                                  ;FROM AN ODD WORD BOUNDARY
12289  036042  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12290  036046  005301                          DEC     R1              ;COUNT
12291  036050  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12292  036052  012777  003002  146004          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
12293                                                                  ;FROM AN ODD WORD BOUNDARY
12294  036060  012777  177766  145774          MOV #-10.,@RHWC ;WORD COUNT TEN
12295
12296  036066  004077  146546                  JSR     R0,@COMND       ;GO TO DO COMMAND
12297  036072  004200                          REVRED                  ;REVERSE READ
12298
12299                                          ;CHECK THAT RHCS3 HAS DBL
12300  036074  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
12301
12302  036102  017737  146024  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12303  036110  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12304                                                                  ;DATA WITH DATA READ FROM
12305                                                                  ;RHCS3
12306  036116  001401                          BEQ     65$             ;BRANCH IF GOOD
12307  036120  104124                          ERROR   124
12308                                                                  ;AFTER SETTING "CLR" BIT #5
12309                                                                  ;IN RHCS2 TO INIT THE RH
12310                                                                  ;A TEN WORD READ REVERSE FROM AN
12311                                                                  ;ODD WORD BOUNDARY WAS DONE
12312                                                                  ;THEN
12313                                                                  ;RHCS3 SHOULD HAVE DBL
12314                                                                  ;=2000
12315                                                                  ;BUT CONTAINED WHAT IS
12316                                                                  ;GIVEN IN BAD RHCS3
12317  036122                          65$:
12318  036122  042777  000076  145730          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12319                                          ;CHECK THAT RHCS1 HAS RDY!DVA
12320  036130  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12321
12322  036136  017737  145716  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12323  036144  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12324                                                                  ;DATA WITH DATA READ FROM
12325                                                                  ;RHCS1
12326  036152  001401                          BEQ     67$             ;BRANCH IF GOOD
12327  036154  104125                          ERROR   125
12328                                                                  ;AFTER SETTING "CLR" BIT #5
12329                                                                  ;IN RHCS2 TO INIT THE RH
12330                                                                  ;A TEN WORD READ REVERSE FROM AN
12331                                                                  ;ODD WORD BOUNDARY WAS DONE
12332                                                                  ;THEN
12333                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
12334                                                                  ;=4200
12335                                                                  ;BUT CONTAINED WHAT IS
```

```
12336                                                                   ;GIVEN IN BAD RHCS1
12337  036156                          67$:
12338                                          ;CHECK THAT RHCS2 HAS IR
12339  036156  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12340  036164  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12341
12342  036172  017737  145672  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12343  036200  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12344                                                                  ;DATA WITH DATA READ FROM
12345                                                                  ;RHCS2
12346  036206  001401                          BEQ     69$             ;BRANCH IF GOOD
12347  036210  104126                          ERROR   126
12348                                                                  ;AFTER SETTING "CLR" BIT #5
12349                                                                  ;IN RHCS2 TO INIT THE RH
12350                                                                  ;A TEN WORD READ REVERSE FROM AN
12351                                                                  ;ODD WORD BOUNDARY WAS DONE
12352                                                                  ;THEN
12353                                                                  ;RHCS2 SHOULD HAVE IR
12354                                                                  ;=100
12355                                                                  ;TOGETHER WITH UNIT NUMBER
12356                                                                  ;BUT CONTAINED WHAT IS
12357                                                                  ;GIVEN IN BAD RHCS2
12358  036212                          69$:
12359                                          ;CHECK THAT RHWC HAS 0
12360  036212  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12361
12362  036220  017737  145636  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12363  036226  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12364                                                                  ;DATA WITH DATA READ FROM
12365                                                                  ;RHWC
12366  036234  001401                          BEQ     71$             ;BRANCH IF GOOD
12367  036236  104131                          ERROR   131
12368                                                                  ;AFTER SETTING "CLR" BIT #5
12369                                                                  ;IN RHCS2 TO INIT THE RH
12370                                                                  ;A TEN WORD READ REVERSE FROM AN
12371                                                                  ;ODD WORD BOUNDARY WAS DONE
12372                                                                  ;THEN
12373                                                                  ;RHWC SHOULD HAVE 0
12374                                                                  ;BUT CONTAINED WHAT IS
12375                                                                  ;GIVEN IN BAD RHWC
12376  036240                          71$:
12377
12378
12379
12380                                  ;;*****************************************************************
12381                                  ;*TEST 65       TEST DBL (RHCS3 BIT #10) TEST V
12382
12383                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12384                                  ;*      SET UP FOR A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12385                                  ;*      DO A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12386                                  ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
12387                                  ;*      CHECK RHCS3
12388                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12389                                  ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12390
12391                                  ;;*****************************************************************
```

```
12392  036240  000004                    TST65:  SCOPE
12393  036242  012706  001000                    MOV     #STACK,SP       ;RESET STACK
12394  036246  012737  000065  004656            MOV     #65,@#TSTNM     ;SAVE TEST NUMBER
12395
12396  036254  004737  040312                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12397                                                                    ;SETUP UNIT NUBER
12398                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12399  036260  022737  041710  004640            CMP     #CMNDTM,COMND   ;IS TU THERE
12400
12401  036266  001103                            BNE     TST66   ;BRANCH IF TU NOT THERE
12402
12403
12404                                            ;SET UP FOR A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12405  036270  012701  000003                    MOV     #3,R1           ;COUNT OF THREE
12406  036274  012702  003000                    MOV     #3FEVEN,R2      ;START THREE WORD BUFFER
12407                                                                    ;FROM AN EVEN WORD BOUNDARY
12408  036300  012722  052525            1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12409  036304  005301                            DEC     R1              ;COUNT
12410  036306  001374                            BNE     1$              ;BRANCH IF THREE NOT DONE
12411  036310  012777  003000  145546            MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
12412                                                                    ;FROM AN EVEN WORD BOUNDARY
12413  036316  012777  177765  145536            MOV #-11.,@RHWC ;WORD COUNT ELEVEN
12414
12415  036324  004077  146310                    JSR     R0,@COMND       ;GO TO DO COMMAND
12416  036330  004200                            REVRED                  ;REVERSE READ
12417
12418                                            ;CHECK THAT RHCS3 HAS DBL
12419  036332  012737  002000  001124            MOV     #DBL,$GDDAT     ;GET GOOD = 2000
12420
12421  036340  017737  145566  001126            MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12422  036346  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12423                                                                    ;DATA WITH DATA READ FROM
12424                                                                    ;RHCS3
12425  036354  001401                            BEQ     65$             ;BRANCH IF GOOD
12426  036356  104124                            ERROR   124
12427                                                                    ;AFTER SETTING "CLR" BIT #5
12428                                                                    ;IN RHCS2 TO INIT THE RH
12429                                                                    ;A ELEVEN WORD READ REVERSE FROM AN
12430                                                                    ;EVEN WORD BOUNDARY WAS DONE
12431                                                                    ;THEN
12432                                                                    ;RHCS3 SHOULD HAVE DBL
12433                                                                    ;=2000
12434                                                                    ;BUT CONTAINED WHAT IS
12435                                                                    ;GIVEN IN BAD RHCS3
12436  036360                            65$:
12437  036360  042777  000076  145472            BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12438                                            ;CHECK THAT RHCS1 HAS RDY!DVA
12439  036366  012737  004200  001124            MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12440
12441  036374  017737  145460  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12442  036402  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12443                                                                    ;DATA WITH DATA READ FROM
12444                                                                    ;RHCS1
12445  036410  001401                            BEQ     67$             ;BRANCH IF GOOD
12446  036412  104125                            ERROR   125
12447                                                                    ;AFTER SETTING "CLR" BIT #5
```

```
12448                                                                   ;IN RHCS2 TO INIT THE RH
12449                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12450                                                                   ;EVEN WORD BOUNDARY WAS DONE
12451                                                                   ;THEN
12452                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
12453                                                                   ;=4200
12454                                                                   ;BUT CONTAINED WHAT IS
12455                                                                   ;GIVEN IN BAD RHCS1
12456  036414                                67$:
12457                                                ;CHECK THAT RHCS2 HAS IR
12458  036414  012737  000100  001124                MOV     #IR,$GDDAT      ;GET GOOD = 100
12459  036422  053737  005010  001124                BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12460
12461  036430  017737  145434  001126                MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12462  036436  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12463                                                                   ;DATA WITH DATA READ FROM
12464                                                                   ;RHCS2
12465  036444  001401                                BEQ     69$             ;BRANCH IF GOOD
12466  036446  104126                                ERROR   126
12467                                                                   ;AFTER SETTING "CLR" BIT #5
12468                                                                   ;IN RHCS2 TO INIT THE RH
12469                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12470                                                                   ;EVEN WORD BOUNDARY WAS DONE
12471                                                                   ;THEN
12472                                                                   ;RHCS2 SHOULD HAVE IR
12473                                                                   ;=100
12474                                                                   ;TOGETHER WITH UNIT NUMBER
12475                                                                   ;BUT CONTAINED WHAT IS
12476                                                                   ;GIVEN IN BAD RHCS2
12477  036450                                69$:
12478                                                ;CHECK THAT RHWC HAS 0
12479  036450  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0
12480
12481  036456  017737  145400  001126                MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12482  036464  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12483                                                                   ;DATA WITH DATA READ FROM
12484                                                                   ;RHWC
12485  036472  001401                                BEQ     71$             ;BRANCH IF GOOD
12486  036474  104131                                ERROR   131
12487                                                                   ;AFTER SETTING "CLR" BIT #5
12488                                                                   ;IN RHCS2 TO INIT THE RH
12489                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12490                                                                   ;EVEN WORD BOUNDARY WAS DONE
12491                                                                   ;THEN
12492                                                                   ;RHWC SHOULD HAVE 0
12493                                                                   ;BUT CONTAINED WHAT IS
12494                                                                   ;GIVEN IN BAD RHWC
12495  036476                                71$:
12496
12497
12498
12499                                ;;*****************************************************************
12500                                ;*TEST 66       TEST DBL (RHCS3 BIT #10) TEST W
12501
12502                                ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12503                                ;*      SET UP FOR A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
```

```
12504                                   ;*      DO A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12505                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
12506                                   ;*      CHECK RHCS3
12507                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12508                                   ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12509
12510                                   ;;*****************************************************************
12511  036476  000004                   TST66:  SCOPE
12512  036500  012706  001000                   MOV     #STACK,SP       ;RESET STACK
12513  036504  012737  000066  004656           MOV     #66,@#TSTNM     ;SAVE TEST NUMBER
12514
12515  036512  004737  040312                   JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12516                                                                   ;SETUP UNIT NUBER
12517                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12518  036516  022737  041710  004640           CMP     #CMNDTM,COMND   ;IS TU THERE
12519
12520  036524  001103                           BNE     TST67   ;BRANCH IF TU NOT THERE
12521
12522
12523                                           ;SET UP FOR A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12524  036526  012701  000003                   MOV     #3,R1           ;COUNT OF THREE
12525  036532  012702  003002                   MOV     #BFODD,R2       ;START THREE WORD BUFFER
12526                                                                   ;FROM AN ODD WORD BOUNDARY
12527  036536  012722  052525           1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12528  036542  005301                           DEC     R1              ;COUNT
12529  036544  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
12530  036546  012777  003002  145310           MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
12531                                                                   ;FROM AN ODD WORD BOUNDARY
12532  036554  012777  177765  145300           MOV #-11.,@RHWC ;WORD COUNT ELEVEN
12533
12534  036562  004077  146052                   JSR     R0,@COMND       ;GO TO DO COMMAND
12535  036566  004200                           REVRED                  ;REVERSE READ
12536
12537                                           ;CHECK THAT RHCS3 HAS 0
12538  036570  012737  000000  001124           MOV     #0,$GDDAT       ;GET GOOD = 0
12539
12540  036576  017737  145330  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12541  036604  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12542                                                                   ;DATA WITH DATA READ FROM
12543                                                                   ;RHCS3
12544  036612  001401                           BEQ     65$             ;BRANCH IF GOOD
12545  036614  104124                           ERROR   124
12546                                                                   ;AFTER SETTING "CLR" BIT #5
12547                                                                   ;IN RHCS2 TO INIT THE RH
12548                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12549                                                                   ;ODD WORD BOUNDARY WAS DONE
12550                                                                   ;THEN
12551                                                                   ;RHCS3 SHOULD HAVE 0
12552                                                                   ;BUT CONTAINED WHAT IS
12553                                                                   ;GIVEN IN BAD RHCS3
12554  036616                           65$:
12555  036616  042777  000076  145234           BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12556                                           ;CHECK THAT RHCS1 HAS RDY!DVA
12557  036624  012737  004200  001124           MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12558
12559  036632  017737  145222  001126           MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
```

```
12560  036640  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12561                                                                   ;DATA WITH DATA READ FROM
12562                                                                   ;RHCS1
12563  036646  001401                          BEQ     67$             ;BRANCH IF GOOD
12564  036650  104125                          ERROR   125
12565                                                                   ;AFTER SETTING "CLR" BIT #5
12566                                                                   ;IN RHCS2 TO INIT THE RH
12567                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12568                                                                   ;ODD WORD BOUNDARY WAS DONE
12569                                                                   ;THEN
12570                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
12571                                                                   ;=4200
12572                                                                   ;BUT CONTAINED WHAT IS
12573                                                                   ;GIVEN IN BAD RHCS1
12574  036652                          67$:
12575                                          ;CHECK THAT RHCS2 HAS IR
12576  036652  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12577  036660  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12578
12579  036666  017737  145176  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12580  036674  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMFARE EXPECTED
12581                                                                   ;DATA WITH DATA READ FROM
12582                                                                   ;RHCS2
12583  036702  001401                          BEQ     69$             ;BRANCH IF GOOD
12584  036704  104126                          ERROR   126
12585                                                                   ;AFTER SETTING "CLR" BIT #5
12586                                                                   ;IN RHCS2 TO INIT THE RH
12587                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12588                                                                   ;ODD WORD BOUNDARY WAS DONE
12589                                                                   ;THEN
12590                                                                   ;RHCS2 SHOULD HAVE IR
12591                                                                   ;=100
12592                                                                   ;TOGETHER WITH UNIT NUMBER
12593                                                                   ;BUT CONTAINED WHAT IS
12594                                                                   ;GIVEN IN BAD RHCS2
12595  036706                          69$:
12596                                          ;CHECK THAT RHWC HAS 0
12597  036706  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12598
12599  036714  017737  145142  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12600  036722  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12601                                                                   ;DATA WITH DATA READ FROM
12602                                                                   ;RHWC
12603  036730  001401                          BEQ     71$             ;BRANCH IF GOOD
12604  036732  104131                          ERROR   131
12605                                                                   ;AFTER SETTING "CLR" BIT #5
12606                                                                   ;IN RHCS2 TO INIT THE RH
12607                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12608                                                                   ;ODD WORD BOUNDARY WAS DONE
12609                                                                   ;THEN
12610                                                                   ;RHWC SHOULD HAVE 0
12611                                                                   ;BUT CONTAINED WHAT IS
12612                                                                   ;GIVEN IN BAD RHWC
12613  036734                          71$:
12614
12615
```

```
12616
12617                                   ;;*****************************************************
12618                                   ;*TEST 67       TEST DBL (RHCS3 BIT #10) TEST X
12619
12620                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12621                                   ;*      SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12622                                   ;*      WITH BAI IN RHCS2 BIT #3 SET
12623                                   ;*      DO A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12624                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
12625                                   ;*      CHECK RHCS3
12626                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12627                                   ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12628
12629                                   ;;*****************************************************
12630  036734  000004                  TST67:  SCOPE
12631  036736  012706  001000                  MOV     #STACK,SP       ;RESET STACK
12632  036742  012737  000067  004656          MOV     #67,@#TSTNM     ;SAVE TEST NUMBER
12633
12634  036750  004737  040312                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12635                                                                  ;SETUP UNIT NUBER
12636                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12637  036754  022737  041710  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
12638
12639  036762  001106                          BNE     TST70   ;BRANCH IF TU NOT THERE
12640
12641
12642                                           ;SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12643  036764  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12644  036770  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
12645                                                                  ;FROM AN ODD WORD BOUNDARY
12646  036774  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12647  037000  005301                          DEC     R1              ;COUNT
12648  037002  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12649  037004  012777  003002  145052          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
12650                                                                  ;FROM AN ODD WORD BOUNDARY
12651  037012  012777  177766  145042          MOV #-10.,@RHWC ;WORD COUNT TEN
12652  037020  052777  000010  145042  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
12653
12654  037026  004077  145606                  JSR     R0,@COMND       ;GO TO DO COMMAND
12655  037032  004200                          REVRED                  ;REVERSE READ
12656
12657                                           ;CHECK THAT RHCS3 HAS 0
12658  037034  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12659
12660  037042  017737  145064  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12661  037050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12662                                                                  ;DATA WITH DATA READ FROM
12663                                                                  ;RHCS3
12664  037056  001401                          BEQ     65$             ;BRANCH IF GOOD
12665  037060  104124                          ERROR   124
12666                                                                  ;AFTER SETTING "CLR" BIT #5
12667                                                                  ;IN RHCS2 TO INIT THE RH
12668                                                                  ;A TEN WORD READ REVERSE FROM AN
12669                                                                  ;ODD WORD BOUNDARY WAS DONE
12670                                                                  ;WITH BAI IN RHCS2 SET
12671                                                                  ;THEN
```

```
12672                                                                   ;RHCS3 SHOULD HAVE 0
12673                                                                   ;BUT CONTAINED WHAT IS
12674                                                                   ;GIVEN IN BAD RHCS3
12675  037062                                65$:
12676  037062  042777  000076  144770        BIC     #76,@RHCS1        ;CLEAR FUNCTION BITS
12677                                        ;CHECK THAT RHCS1 HAS RDY!DVA
12678  037070  012737  004200  001124        MOV     #RDY!DVA,$GDDAT   ;GET GOOD = 4200
12679
12680  037076  017737  144756  001126        MOV     @RHCS1,$BDDAT     ;READ RHCS1 FOR COMPARISON
12681  037104  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;COMPARE EXPECTED
12682                                                                   ;DATA WITH DATA READ FROM
12683                                                                   ;RHCS1
12684  037112  001401                        BEQ     67$               ;BRANCH IF GOOD
12685  037114  104125                        ERROR   125
12686                                                                   ;AFTER SETTING "CLR" BIT #5
12687                                                                   ;IN RHCS2 TO INIT THE RH
12688                                                                   ;A TEN WORD READ REVERSE FROM AN
12689                                                                   ;ODD WORD BOUNDARY WAS DONE
12690                                                                   ;WITH BAI IN RHCS2 SET
12691                                                                   ;THEN
12692                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
12693                                                                   ;=4200
12694                                                                   ;BUT CONTAINED WHAT IS
12695                                                                   ;GIVEN IN BAD RHCS1
12696  037116                                67$:
12697                                        ;CHECK THAT RHCS2 HAS IR!BAI
12698  037116  012737  000110  001124        MOV     #IR!BAI,$GDDAT    ;GET GOOD = 110
12699  037124  053737  005010  001124        BIS     UNIT,$GDDAT       ;INCLUDE UNIT NUMBER
12700
12701  037132  017737  144732  001126        MOV     @RHCS2,$BDDAT     ;READ RHCS2 FOR COMPARISON
12702  037140  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;COMPARE EXPECTED
12703                                                                   ;DATA WITH DATA READ FROM
12704                                                                   ;RHCS2
12705  037146  001401                        BEQ     69$               ;BRANCH IF GOOD
12706  037150  104126                        ERROR   126
12707                                                                   ;AFTER SETTING "CLR" BIT #5
12708                                                                   ;IN RHCS2 TO INIT THE RH
12709                                                                   ;A TEN WORD READ REVERSE FROM AN
12710                                                                   ;ODD WORD BOUNDARY WAS DONE
12711                                                                   ;WITH BAI IN RHCS2 SET
12712                                                                   ;THEN
12713                                                                   ;RHCS2 SHOULD HAVE IR!BAI
12714                                                                   ;=110
12715                                                                   ;TOGETHER WITH UNIT NUMBER
12716                                                                   ;BUT CONTAINED WHAT IS
12717                                                                   ;GIVEN IN BAD RHCS2
12718  037152                                69$:
12719                                        ;CHECK THAT RHWC HAS 0
12720  037152  012737  000000  001124        MOV     #0,$GDDAT         ;GET GOOD = 0
12721
12722  037160  017737  144676  001126        MOV     @RHWC,$BDDAT      ;READ RHWC FOR COMPARISON
12723  037166  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;COMPARE EXPECTED
12724                                                                   ;DATA WITH DATA READ FROM
12725                                                                   ;RHWC
12726  037174  001401                        BEQ     71$               ;BRANCH IF GOOD
12727  037176  104131                        ERROR   131
```

```
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;A TEN WORD READ REVERSE FROM AN
                                                        ;ODD WORD BOUNDARY WAS DONE
                                                        ;WITH BAI IN RHCS2 SET
                                                        ;THEN
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
12737  037200                       71$:
                                ;;*****************************************************************
                                ;*TEST 70        END OF ONE RH
                                ;*      THIS IS THE END OF TEST FOR ONE RH
                                ;*      IF THERE ARE MORE RH THEN THE PROGRAM
                                ;*      JUMPS TO TEST 2 FOR NEXT RH TEST
                                ;*      END PASS IS REACHED ONLY AFTER ALL RH ARE COMPLETE
                                ;;*****************************************************************
12761  037200  000004           TST70:  SCOPE
12762  037202  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
12763  037210  005037  177776                  CLR     PS              ;REINSTATE PS TO 0
12764  037214  104401  037222                  TYPE    ,65$            ;;TYPE ASCIZ STRING
12765  037220  000414                          BR      64$             ;;GET OVER THE ASCIZ
12766                           ;;65$:  .ASCIZ  <15><12>/END OF TEST FOR RH NO/
12767  037252                   64$:
12770  037252  013746  005510                  MOV     WORUNT,-(SP)    ;GET WORKING RH NUMBER LEFT
12771  037256  013702  005506                  MOV     TSTUNT,R2       ;GET TOTAL NO OF RH FOUND
12772  037262  162602                          SUB     (SP)+,R2        ;NO OF RH TESTED
12773  037264  006302                          ASL     R2              ;INDEX FOR RH TESTED
12774  037266  016246  005436                  MOV     FRHNM(R2),-(SP) ;TYPE OF RH NO.
12775  037272  104405                          TYPDS
12776  037274  104401  037302                  TYPE    ,67$            ;;TYPE ASCIZ STRING
12777  037300  000410                          BR      66$             ;;GET OVER THE ASCIZ
13000                           ;;67$:  .ASCIZ  /           BASE /
13001  037322                   66$:
13002  037322  013746  004060                  MOV     RHCS1,-(SP)     ;TYPE BASE ADDRESS
13003  037326  104402                          TYPOC
13004  037330  104401  037336                  TYPE    ,69$            ;;TYPE ASCIZ STRING
13005  037334  000421                          BR      68$             ;;GET OVER THE ASCIZ
13006                           ;;69$:  .ASCIZ  <15><12>/TOTAL NO OF ERRORS ON THIS PASS/
13007  037400                   68$:
13010  037400  013746  001112                  MOV     $ERTTL,-(SP)    ;TYPE TOTAL NO OF ERRORS
13011  037404  104405                          TYPDS
13012  037406  104401  037414                  TYPE    ,71$            ;;TYPE ASCIZ STRING
13013  037412  000402                          BR      70$             ;;GET OVER THE ASCIZ
13014                           ;;71$:  .ASCIZ  <15><12>/ /
13015  037420                   70$:
13016  037420  104401  037426                  TYPE    ,73$            ;;TYPE ASCIZ STRING
13017  037424  000402                          BR      72$             ;;GET OVER THE ASCIZ
13020                           ;;73$:  .ASCIZ  <15><12>/ /
13021  037432                   72$:
```

E03


```
12784  037432  005037  001112                    CLR     $ERTTL          ;CLEAR TOTAL NO OF ERRORS
12785  037436  005337  005510                    DEC     WORUNT          ;DECREASE NO. OF RH TESTED
12786  037442  001402                            BEQ     1$
12787  037444  000137  010622                    JMP     TST2
12788  037450  013737  005506  005510  1$:       MOV     TSTUNT,WORUNT   ;MAKE WORKING RH SAME
12789                                                                    ;AS TOTAL RH
```

```
                                        .SBTTL  END OF PASS ROUTINE

                                        ;;**************************************************************
                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO TST2

037456                                  $EOP:
037456  000004                                SCOPE
037460  005037  001102                        CLR     $TSTNM          ;;ZERO THE TEST NUMBER
037464  005037  001212                        CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
037470  005237  001100                        INC     $PASS           ;;INCREMENT THE PASS NUMBER
037474  042737  100000  001100              BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
037502  005327                              DEC     (PC)+           ;;LOOP?
037504  000001                          $EOPCT: .WORD   1
037506  003022                              BGT     $DOAGN          ;;YES
037510  012737                              MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
037512  000001                          $ENDCT: .WORD   1
037514  037504                                  $EOPCT
037516  104401  037563                        TYPE    ,$ENDMG         ;;TYPE "END PASS #"
037522  013746  001100                        MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
037526  104405                              TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037530  104401  037560                        TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
037534  013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
037540  001405                              BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
037542  000005                              RESET                   ;;CLEAR THE WORLD
037544  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
037546  000240                              NOP                     ;;SAVE ROOM
037550  000240                              NOP                     ;;FOR
037552  000240                              NOP                     ;;ACT11
037554                                  $DOAGN:
037554  000137                              JMP     @(PC)+          ;;RETURN
037556  010622                          $RTNAD: .WORD   TST2
037560     377        377     000      $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
037563     015  042412  042116          $ENDMG: .ASCIZ  <15><12>/END PASS #/
037570  050040  051501  020123
037576  000043

                                        .SBTTL  SUBROUTINES

                                        ;;**************************************************************
                                        ;*      THIS ROUTINE HANDLES TIME OUT TRAPS THROUGH LOC. 4
                                        ;;**************************************************************

037600                                  TIEOUT:
037600  104401  037606                        TYPE    ,65$            ;;TYPE ASCIZ STRING
037604  000433                              BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/TIMEOUT OCCURED THROUGH LOCATION 4 FROM PROGRAM PC/
```

```
12846  037674                          64$:
12847  037674  162716  000002                  SUB     #2,(SP)         ;TYPE PC
12848  037700  104402                          TYPOC
12849  037702  104401  037710                  TYPE    ,67$            ;;TYPE ASCIZ STRING
12850  037706  000410                          BR      66$             ;;GET OVER THE ASCIZ
12851                                  ;;67$:  .ASCIZ  <15><12>/PSW WAS     /
12852  037730                          66$:
12853  037730  104402                          TYPOC
12854  037732  000137  043120                  JMP     $SCOPE  ;RETURN TO TEST
12855
12856
12857
12858
12859
12860
12861
12862
12863                                   ;;*****************************************************************
12864                                   ;*      THIS ROUTINE HANDLES PARITY ERRORS
12865                                   ;;*****************************************************************
12866  037736                          PARITY:
12867  037736  104401  037744                  TYPE    ,65$            ;;TYPE ASCIZ STRING
12868  037742  000417                          BR      64$             ;;GET OVER THE ASCIZ
12869                                  ;;65$:  .ASCIZ  <15><12>/PARITY TRAP THRU VECTOR 114/
12870  040002                          64$:
12871  040002  104401  040010                  TYPE    ,67$            ;;TYPE ASCIZ STRING
12872  040006  000413                          BR      66$             ;;GET OVER THE ASCIZ
12873                                  ;;67$:  .ASCIZ  <15><12>/FROM PROGRAM PC   /
12874  040036                          66$:
12875  040036  162716  000002                  SUB     #2,(SP)                 ;TYPE PC
12876  040042  104402                          TYPOC
12877  040044  104401  040052                  TYPE    ,69$            ;;TYPE ASCIZ STRING
12878  040050  000407                          BR      68$             ;;GET OVER THE ASCIZ
12879                                  ;;69$:  .ASCIZ  <15><12>/PSW WAS   /
12880  040070                          68$:
12881  040070  104402                          TYPOC
12882  040072  104401  040100                  TYPE    ,71$            ;;TYPE ASCIZ STRING
12883  040076  000421                          BR      70$             ;;GET OVER THE ASCIZ
12884                                  ;;71$:  .ASCIZ  <15><1>/HIGH ERROR ADDRESS REGISTER    /
12885  040142                          70$:
12886  040142  017746  145350                MOV     @HERADD,-(SP)           ;TYPE HIGH ERROR
12887  040146  104402                          TYPOC
12888  040150  104401  040156                  TYPE    ,73$            ;;TYPE ASCIZ STRING
12889  040154  000421                          BR      72$             ;;GET OVER THE ASCIZ
12890                                  ;;73$:  .ASCIZ  <15><12>/LOW ERROR ADDRESS REGISTER    /
12891  040220                          72$:
12892  040220  017746  145270                MOV     @LERADD,-(SP)           ;TYPE LOW ERROR
12893  040224  104402                          TYPOC
12894  040226  104401  040234                  TYPE    ,75$            ;;TYPE ASCIZ STRING
12895  040232  000422                          BR      74$             ;;GET OVER THE ASCIZ
12896                                  ;;75$:  .ASCIZ  <15><12>/MEMORY SYSTEM ERROR REGISTER    /
12897  040300                          74$:
12898  040300  017746  145214                MOV     @MEMERR,-(SP)
12899  040304  104402                          TYPOC
12900  040306  000137  043120                  JMP     $SCOPE
12901
```

```
12902
12903
12904
12905
12906
12907
12908  040312  012777  000040  143550  CLDISK: MOV     #CLR,@RHCS2             ;CLEAR ALL REG.
12909  040320  013777  005010  143542          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
12910  040326  042777  000077  143524          BIC     #77,@RHCS1                      ;CLEAR FUNCTION BITS
12911  040334  005077  143522                  CLR     @RHWC                   ;CLEAR RHWC
12912  040340  000207                          RTS     PC
```

```
12913
12914
12915
12916
12917                                   ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
12918                                   ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
12919  040342  000000                  PCJSR:  0                       ;PC OF JSR
12920
12921
12922
12923
12924  040344  011637  040342          CHECK:  MOV     (SP),@#PCJSR            ;SAVE PC OF JSR+4
12925  040350  162737  000004  040342          SUB     #4,@#PCJSR      ;GET PC OF JSR
12926  040356  011346                  MOV     @R3,-(SP)       ;GET RHDS1
12927  040360  052716  000100          BIS     #VV,(SP)        ;DONT CHECK VV BIT
12928  040364  000406                  BR      CHECKC          ;GOTO COMMON CHECK ROUTINE
12929  040366  011637  040342  CHECKT: MOV     (SP),@#PCJSR            ;SAVE PC OF JSR+4
12930  040372  162737  000004  040342          SUB     #4,@#PCJSR      ;GET PC OF JSR
12931  040400  011346                  MOV     @R3,-(SP)       ;GET RHDS1 & DO VV CHECK AT 3$
12932  040402  011146          CHECKC: MOV     @R1,-(SP)       ;GET CS1
12933  040404  042716  173577          BIC     #173577,(SP)            ;CLEAR UNWANTED BITS
12934  040410  022726  004200          CMP     #DVA!RDY,(SP)+          ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
12935                                                          ;AND BE READY
12936  040414  001403                  BEQ     3$              ;BRANCH IF GOOD
12937  040416  011137  001122          MOV     @R1,@#$BDADR    ;BAD DATA REGISTER (RHCS1)
12938  040422  104062                  ERROR   62              ;RHCS1 DID NOT HAVE DEVICE
12939                                                          ;AVAILABLE RIGHT AT THE START
12940                                                          ;ALL OTHER BITS SHOULD BE 0
12941  040424  042716  102000  3$:     BIC     #ATA!LBT,(SP)           ;CLEAR UNWANTED BITS
12942  040430  022726  010700          CMP     #MOL!DPR!DRY!VV,(SP)+;RHDS1 SHOULD HAVE THESE SET
12943  040434  001403                  BEQ     7$              ;BRANCH IF GOOD
12944  040436  011337  001122          MOV     @R3,@#$BDADR    ;BAD DATA IN  REGISTER (RHDS1)
12945  040442  104061                  ERROR   61              ;RHDS1 HAS SOME BITS OTHER
12946                                                          ;THAN MOL, DRY, DPR,VV SET
12947                                                          ;ALL OTHER BITS SHOULD BE 0
12948  040444  000207          7$:     RTS     PC                      ;RETURN TO TEST NO.
12949
```

```
12950
12951
12952
12953
12954
12955
12956
12957
12958
12959
12960                                   ;THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
12961                                   ;NO TIMING IS DONE
12962                                   ;CALL IS
12963                                   ;       WAT
12964                                   ;       A       ;POINTER TO REGISTER ADDRESS
12965                                   ;       B       ;BIT WAITED FOR
12966                                   ;R3-IS A TEMPORARY COUNTER
12967  040446  177777                  TIMCNT: 177777                  ;COUNT FOR WAIT LOOP
12968
12969  040450                           WAIT.T:
12970  040450  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
12971  040452  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
12972  040454  016600  000004                  MOV     4(SP),R0        ;R0 HAS ADDRESS OF NEXT LOCATION
12973  040460  010037  004666                  MOV     R0,@#WAITPC     ;WAT PC +2 IS IN WAITPC
12974  040464  162737  000002  004666          SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
12975  040472  013037  004670                  MOV     @(R0)+,@#WAITRE ;WAIT ON REGISTER ADDRESS
12976  040476  012037  004672                  MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
12977  040502  010066  000004                  MOV     R0,4(SP)        ;RESTORE RETURN ON STACK
12978
12979                                   ;THIS HAS THE TWO COUNT DOWNS FROM 177777
12980                                   ;FOR WAITING FOR A BIT TO SET
12981  040506  005737  004646                  TST     WAT0            ;WAITING FOR 1?
12982  040512  001025                          BNE     4$              ;BRANCH IF WAITING FOR 0
12983  040514  013703  040446                  MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
12984  040520  033777  004672  144142  1$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
12985  040526  001046                          BNE     3$              ;BRANCH IF YES
12986  040530  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
12987  040532  001372                          BNE     1$
12988  040534  013703  040446                  MOV     @#TIMCNT,R3     ;SECOND COUNT DOWN FROM 177777
12989  040540  033777  004672  144122  2$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
12990  040546  001036                          BNE     3$              ;BRANCH IF YES
12991  040550  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOOT THERE
12992  040552  001372                          BNE     2$
12993  040554  017737  144110  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
12994  040562  104135                          ERROR   135             ;WAS WAITING FOR A BIT TO SET
12995                                                                  ;BIT IN QUESTION IS IN "BIT WAITED"
12996                                                                  ;REGISTER IN QUESTION IS IN "REG ADDR"
12997  040564  000427                          BR      3$              ;BRANCH OUT
12998
12999
13000
13001                                   ;THIS HAS THE TWO COUNT DOWNS FROM 177777
13002                                   ;FOR WAITING FOR A BIT TO RESET
13003  040566  013703  040446          4$:     MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
13004  040572  033777  004672  144070  5$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
13005  040600  001421                          BEQ     3$              ;BRANCH IF NO
```

```
13006  040602  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
13007  040604  001372                          BNE     5$
13008  040606  013703  040446                  MOV     @#TIMCNT,R3     ;SECOND COUNT DOWN FROM 177777
13009  040612  033777  004672  144050  6$:     BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
13010  040620  001411                          BEQ     3$              ;BRANCH IF NO
13011  040622  005303                          DEC     R3              ;COUNT IF REQUIRED BIT NOOT THERE
13012  040624  001372                          BNE     6$
13013  040626  017737  144036  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
13014  040634  104136                          ERROR   136             ;WAS WAITING FOR A BIT TO RESET
13015                                                                  ;BIT IN QUESTION IS IN "BIT WAITED"
13016                                                                  ;REGISTER IN QUESTION IS IN "REG ADDR"
13017  040636  017737  144026  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
13018  040644                          3$:
13019  040644  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
13020  040646  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
13021  040650  000002                          RTI                     ;RETURN TO MAIN TEST
```

```
 13022
 13023
 13024
 13025                                  ;HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
 13026                                  ;ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
 13027                                  ;PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
 13028
 13029                                  ;WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
 13030                                  ;THE PROGRAM GOES BACK TO CAN BE CHANGED.
 13031                                  ;THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
 13032                                  ;1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
 13033                                  ;2.  LOOP ON ERROR SWITCH MUST BE SET
 13034                                  ;3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
 13035                                  ;IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
 13036                                  ;THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
 13037                                  ;TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
 13038                                  ;THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
 13039                                  ;COMES TO THE END OF THE TEST UNDER CONSIDERATION.
 13040                                  ;
 13041                                  ;AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
 13042                                  ;NORMAL OPERATION WILL CONTINUE.
 13043
 13044  040652  000000                  TESTAD: 0                       ;FIRST ADDRESS OF TEST
 13045  040654                          OPERSEL:
 13046  040654  005037  177776                  CLR     PS              ;MAKE PROCESSOR STATUS ZERO
 13047  040660  012737  177777  045100          MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
 13048  040666  104401  040674                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 13049  040672  000421                          BR      64$             ;;GET OVER THE ASCIZ
 13050                                  ;;65$:  .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
 13051  040736                          64$:
 13052  040736  013746  004656                  MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
 13053  040742  104402                          TYPOC                   ;NUMBER
 13054  040744  104401  040752                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 13055  040750  000414                          BR      66$             ;;GET OVER THE ASCIZ
 13056                                  ;;67$:  .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
 13057  041002                          66$:
 13058  041002  013746  001110                  MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
 13059  041006  104402                          TYPOC
 13060  041010  104401  001223                  TYPE    ,$CRLF
 13061  041014  104401  041022                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 13062  041020  000430                          BR      68$             ;;GET OVER THE ASCIZ
 13063                                  ;;69$:  .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
 13064  041102                          68$:
 13065  041102  104401  041110                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 13066  041106  000430                          BR      70$             ;;GET OVER THE ASCIZ
 13067                                  ;;71$:  .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
 13068  041170                          70$:
 13069  041170  104401  041176                  TYPE    ,73$            ;;TYPE ASCIZ STRING
 13070  041174  000422                          BR      72$             ;;GET OVER THE ASCIZ
 13071                                  ;;73$:  .ASCIZ  <15><12>/ FOLLOWED BY A CARRIAGE RETURN //<15><12>
 13072  041242                          72$:
 13073  041242  104410                          RDOCT
 13074  041244  062716  000002                  ADD     #2,(SP)         ;GET LPADR
 13075  041250  012637  001106                  MOV     (SP)+,@#$LPADR
 13076  041254  104401  041262                  TYPE    ,75$            ;;TYPE ASCIZ STRING
 13077  041260  000417                          BR      74$             ;;GET OVER THE ASCIZ
```

```
13078                                          ;;75$:  .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
13079  041320                                  74$:
13080  041320  104401  041326                         TYPE    ,77$            ;;TYPE ASCIZ STRING
13091  041324  000440                                 BR      76$             ;;GET OVER THE ASCIZ
13082                                          ;;77$:  .ASCIZ  <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15
13083  041426                                  76$:
13084  041426  104410                                 RDOCT
13085  041430  012637  001110                         MOV     (SP)+,@#$LPERR  ;GET LPERR
13086  041434  013746  001106                         MOV     @#$LPADR,-(SP)
13087  041440  000002                                 RTI
13088
13089
13090
13091
13092                                          ;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
13093                                          ;IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
13094                                          ;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
13095                                          ;AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
13096                                          ;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
13097
13098
13099
13100
13101  041442                                  PUTREG:
13102  041442  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
13103  041444  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
13104  041446  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
13105  041450  012700  004062                         MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
13106  041454  012701  000002                         MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
13107  041460  012702  000010                         MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
13108  041464  013021                          10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
13109  041466  005302                                 DEC     R2
13110  041470  001375                                 BNE     10$
13111  041472  012602                                 MOV     (SP)+,R2        ;;POP STACK INTO R2
13112  041474  012601                                 MOV     (SP)+,R1        ;;POP STACK INTO R1
13113  041476  012600                                 MOV     (SP)+,R0        ;;POP STACK INTO R0
13114  041500  000207                                 RTS     PC
```

```
13115
13116                                   .SBTTL  RS DATA TRANSFER SUBROUTINE
13117
13118                                   ;*      THIS SUBROUTINE WRITES OR READS OR DOES A
13119                                   ;*      WRITE CHECK ON CYLINDER 0 TRACK 0 SECTOR 0
13120                                   ;*      IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY
13121                                   ;*      FILLED
13122                                   ;*      WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
13123                                   ;*      MAIN PROGRAM IT MEANS THE TRANSFER IS COMPLETE
13124                                   ;*
13125                                   ;*      THE CALL IS BY A JSR R0,@COMND COMAND
13126
13127  041502  012037  004642          CMNDRS: MOV     (R0)+,COMAND    ;GET COMMAND
13128
13129                                          ;CHECK THAT RHDS1 HAS DRY!DPR!MOL
13130  041506  012737  010600  001124          MOV     #DRY!DPR!MOL,$GDDAT     ;GET GOOD = 10300
13131
13132  041514  017737  142352  001126          MOV     @RHDS1,$BDDAT   ;READ RHDS1 FOR COMPARISON
13133  041522  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
13134                                                                  ;DATA WITH DATA READ FROM
13135                                                                  ;RHDS1
13136  041530  001401                          BEQ     65$             ;BRANCH IF GOOD
13137  041532  104133                          ERROR   133
13138                                                                  ;BEFORE ANY COMMAND IS GIVEN
13139                                                                  ;RHDS1 SHOULD HAVE DRY!DPR!MOL
13140                                                                  ;=10300
13141                                                                  ;BUT CONTAINED WHAT IS
13142                                                                  ;GIVEN IN BAD RHDS1
13143  041534                          65$:
13144
13145  041534  005077  142326                  CLR     @RHDA           ;TRANSFER ON CYLINDER 0
13146                                                                  ;TRANSFER ON SECTOR 0
13147                                                                  ;TRACK 0
13148  041540  017777  143076  142312          MOV     @COMAND,@RHCS1  ;SET UP COMMAND
13149  041546  005737  004644                  TST     NOGO            ;IS GO TO BE GIVEN
13150  041552  001006                          BNE     1$              ;BRANCH IF NO
13151  041554  052777  000001  142276          BIS     #GO,@RHCS1      ;SET GO
13152
13153                                          ;WAIT FOR RDY BIT IN RHCS1 REGISTER TO SET
13154  041562  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13155  041564  004060                          RHCS1                   ;AND WAIT FOR RDY BIT IN
13156  041566  000200                          RDY                     ;RHCS1 REGISTER
13157                                                                  ;IF ERROR OCCURS HERE
13158                                                                  ;IT MEANS "RDY" DID NOT
13159                                                                  ;SET FOR THE FULL COUNT
13160                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13161                                                                  ;THIS TIME IS APPROXIMATELY
13162                                                                  ;LARGER THAN 500 MILLISECONDS
13163  041570  000200                  1$:     RTS     R0
13164
13165                                   .SBTTL  RP04 DATA TRANSFER SUBROUTINE
13166
13167                                   ;*      THIS SUBROUTINE WRITE/READ/WRITE CHECK ON THE RP04 CYLINDER 0,
13168                                   ;*      TRACK 0, SECTOR 0.
13169                                   ;*      IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
13170                                   ;*      WHEN IT RETURNS FROM THIS SUBROUTINE TO THE MAIN
```

```
                                        ;*      PROGRAM IT MEANS THE COMMAND IS COMPLETE
                                        ;*
                                        ;*      THE CALL IS BY A JSR R0,@COMND COMMAND.
                                        
041572  012037  004642          CMNDRP: MOV     (R0)+,COMAND    ;GET COMMAND
                                        
041576  013746  004172                  MOV     PKACK,-(SP)     ;GET READY TO SET VV
041602  062716  000001                  BIS     #GO,(SP)        ;GET PACK ACKNOWLEDGE COMMAND
041606  012677  142246                  MOV     (SP)+,@RHCS1    ;SET VV
                                        
                                        ;CHECK THAT RHDS1 HAS VV!DRY!DPR!MOL
041612  012737  010700  001124          MOV     #VV!DRY!DPR!MOL,$GDDAT  ;GET GOOD = 10700
                                        
041620  017737  142246  001126          MOV     @RHDS1,$BDDAT   ;READ RHDS1 FOR COMPARISON
041626  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDS1
041634  001401                          BEQ     65$             ;BRANCH IF GOOD
041636  104132                          ERROR   132
                                                                ;BEFORE ANY COMMAND IS GIVEN
                                                                ;RHDS1 SHOULD HAVE VV!DRY!DPR!MOL
                                                                ;=10700
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDS1
041640                          65$:
                                        
041640  005077  142250                  CLR     @RHCA           ;TRANSFER ON CYLINDER 0
041644  005077  142216                  CLR     @RHDST          ;TRANSFER ON SECTOR 0
                                                                ;TRACK 0
041650  012777  014000  142234          MOV     #ECI!FMT22,@RHOF;INHIBIT ECC
                                                                ;FORMAT 16 BIT PER WORD
041656  017777  142760  142174          MOV     @COMAND,@RHCS1  ;SET UP COMMAND
041664  005737  004644                  TST     NOGO            ;IS GO TO BE GIVEN
041670  001006                          BNE     1$              ;BRANCH IF NO
041672  052777  000001  142160          BIS     #GO,@RHCS1      ;SET GO
                                        
                                        ;WAIT FOR RDY BIT IN RHCS1 REGISTER TO SET
041700  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
041702  004060                          RHCS1                   ;AND WAIT FOR RDY BIT IN
041704  000200                          RDY                     ;RHCS1 REGISTER
                                                                ;IF ERROR OCCURS HERE
                                                                ;IT MEANS "RDY" DID NOT
                                                                ;SET FOR THE FULL COUNT
                                                                ;DOWN OF THE WAIT.T SUBROUTINE
                                                                ;THIS TIME IS APPROXIMATELY
                                                                ;LARGER THAN 500 MILLISECONDS
041706  000200                  1$:     RTS     R0
                                        
                                        .SBTTL  TM02 DATA TRANSFER SUBROUTINE
                                        
                                        ;*      THIS SUBROUTINE WRITES/READS/WRITE CHECKS ON THE TM02-TU16
                                        ;*      ON THE FIRST BLOCK FROM BEGINNING OF TAPE (BOT)
                                        ;*      IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
                                        ;*      WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
                                        ;*      MAIN PROGRAM IT MEANS THE COMMAND IS COMPLETE.
                                        ;*
```

```
                                        ;*      THE CALL IS
                                        ;*      JSR     R0,@COMND
                                        
001710  012037  004642          CMNDTM: MOV     (R0)+,COMAND
001714  013777  004660  142170          MOV     SLAVE,@RHTC     ;GET SLAVE NUMBER
001722  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT ;GET GOOD DATA
001730  017737  142136  001126          MOV     @RHDS1,$BDDAT   ;GET RHDS1
001736  042737  000062  001126          BIC     #BOT!BIT05!BIT04,$BDDAT ;DISREGARD BOT
001744  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED DATA WITH
                                                                ;DATA READ FROM RHDS1
001752  001401                          BEQ     1$              ;BRANCH IF GOOD
001754  104134                          ERROR   134             ;BEFORE ANY COMMAND
                                                                ;IS GIVEN DRIVE STATUS REGISTER
                                                                ;DID NOT CONTAIN WHAT IS IN
                                                                ;GOOD
001756  032777  000002  142106  1$:     BIT     #BOT,@RHDS1     ;IS IT ALREADY AT BOT
001764  001035                          BNE     2$              ;BRANCH IF YES
                                        ;IF NOT AT BOT GIVE A REWIND COMMAND
001766  013777  004176  142064          MOV     REWIND,@RHCS1   ;LOAD RHCS1 WITH REWIND COMMAND
001774  052777  000001  142056          BIS     #GO,@RHCS1
                                        ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
042002  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
042004  004072                          RHDS1                   ;AND WAIT FOR PIP BIT IN
042006  020000                          PIP                     ;RHDS1 REGISTER
                                                                ;IF ERROR OCCURS HERE
                                                                ;IT MEANS "PIP" DID NOT
                                                                ;SET FOR THE FULL COUNT
                                                                ;DOWN OF THE WAIT.T SUBROUTINE
                                                                ;THIS TIME IS APPROXIMATELY
                                                                ;LARGER THAN 500 MILLISECONDS
                                        ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
042010  012737  177777  004646                  MOV     #-1,WATO        ;WAIT FOR PIP TO GO FROM 1 TO 0
042016  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
042020  004072                          RHDS1                   ;AND WAIT FOR PIP BIT IN
042022  020000                          PIP                     ;RHDS1 REGISTER
                                                                ;IF ERROR OCCURS HERE
                                                                ;IT MEANS "PIP" DID NOT
                                                                ;SET FOR THE FULL COUNT
                                                                ;DOWN OF THE WAIT.T SUBROUTINE
                                                                ;THIS TIME IS APPROXIMATELY
                                                                ;LARGER THAN 500 MILLISECONDS
042024  005037  004646                  CLR     WATO
                                        ;WAIT FOR BOT BIT IN RHDS1 REGISTER TO SET
042030  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
042032  004072                          RHDS1                   ;AND WAIT FOR BOT BIT IN
042034  000002                          BOT                     ;RHDS1 REGISTER
                                                                ;IF ERROR OCCURS HERE
                                                                ;IT MEANS "BOT" DID NOT
                                                                ;SET FOR THE FULL COUNT
                                                                ;DOWN OF THE WAIT.T SUBROUTINE
                                                                ;THIS TIME IS APPROXIMATELY
                                                                ;LARGER THAN 500 MILLISECONDS
                                        
                                        ;CLEAR ATTENTIONS
042036  013746  004142                  MOV     DCLEAR,-(SP)
042042  052716  000001                  BIS     #GO,(SP)
```

```
13283  042046  012677  142006                  MOV     (SP)+,@RHCS1
13284
13285                                          ;WAIT FOR DRY BIT IN RHDS1 REGISTER TO SET
13286  042052  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
13287  042054  004072                         RHDS1                   ;AND WAIT FOR DRY BIT IN
13288  042056  000200                         DRY                     ;RHDS1 REGISTER
13289                                                                  ;IF ERROR OCCURS HERE
13290                                                                  ;IT MEANS "DRY" DID NOT
13291                                                                  ;SET FOR THE FULL COUNT
13292                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13293                                                                  ;THIS TIME IS APPROXIMATELY
13294                                                                  ;LARGER THAN 500 MILLISECONDS
13295  042060                          2$:     ;CHECK IF COMMAND IS REVERSE READ
13296                                         ;IF IT IS REVERSE READ, SPACE FORWARD ONE BLOCK
13297  042060  023777  004200  142554          CMP     REVRED,@COMAND  ;IS IT REVERSE READ
13298  042066  001035                         BNE     3$              ;BRANCH IF NOT
13299  042070  052777  001700  142014          BIS     #BPI8!NML,@RHTC ;800 BPI, NORMAL
13300  042076  042777  000010  142006          BIC     #EPAR, @RHTC    ;ODD PARITY
13301
13302  042104  012777  177777  141754          MOV     #-1,@RHFC       ;FRAME COUNT 1
13303                                                                  ;FOR SPACE FORWARD ONE BLOCK
13304  042112  013777  004202  141740          MOV     SPACFD,@RHCS1   ;SPACE FORWARD
13305  042120  052777  000001  141732          BIS     #GO,@RHCS1      ;GO
13306                                         ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
13307  042126  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
13308  042130  004072                         RHDS1                   ;AND WAIT FOR PIP BIT IN
13309  042132  020000                         PIP                     ;RHDS1 REGISTER
13310                                                                  ;IF ERROR OCCURS HERE
13311                                                                  ;IT MEANS "PIP" DID NOT
13312                                                                  ;SET FOR THE FULL COUNT
13313                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13314                                                                  ;THIS TIME IS APPROXIMATELY
13315                                                                  ;LARGER THAN 500 MILLISECONDS
13316                                         ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
13317  042134  012737  177777  004646                  MOV     #-1,WATO        ;WAIT FOR PIP TO GO FROM 1 TO 0
13318  042142  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
13319  042144  004072                         RHDS1                   ;AND WAIT FOR PIP BIT IN
13320  042146  020000                         PIP                     ;RHDS1 REGISTER
13321                                                                  ;IF ERROR OCCURS HERE
13322                                                                  ;IT MEANS "PIP" DID NOT
13323                                                                  ;SET FOR THE FULL COUNT
13324                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13325                                                                  ;THIS TIME IS APPROXIMATELY
13326                                                                  ;LARGER THAN 500 MILLISECONDS
13327  042150  005037  004646                  CLR     WATO
13328                                         ;WAIT FOR DRY BIT IN RHDS1 REGISTER TO SET
13329  042154  104411                         WAT                     ;TRAP TO WAIT.T SUBROUTINE
13330  042156  004072                         RHDS1                   ;AND WAIT FOR DRY BIT IN
13331  042160  000200                         DRY                     ;RHDS1 REGISTER
13332                                                                  ;IF ERROR OCCURS HERE
13333                                                                  ;IT MEANS "DRY" DID NOT
13334                                                                  ;SET FOR THE FULL COUNT
13335                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13336                                                                  ;THIS TIME IS APPROXIMATELY
13337                                                                  ;LARGER THAN 500 MILLISECONDS
13338  042162                          3$:     ;NOW GIVE COMMAND
```

```
13339  042162  052777  001700  141722          BIS     #BPI8!NML,@RHTC ;800 BPI, NORMAL
13340  042170  042777  000010  141714          BIC     #EPAR,@RHTC     ;ODD PARITY
13341  042176  017746  141660                  MOV     @RHWC,-(SP)     ;GET WORD COUNT
13342  042202  005416                          NEG     (SP)            ;GET POSITIVE NUMBER
13343  042204  061616                          ADD     (SP),(SP)       ;DOUBLE WORD COUNT
13344  042206  005416                          NEG     (SP)            ;GET NEGATIVE NUMBER FOR
13345                                                                  ;FRAME COUNT REGISTER
13346  042210  012677  141652                  MOV     (SP)+,@RHFC     ;FILL FRAME COUNT
13347  042214  017777  142422  141636          MOV     @COMAND,@RHCS1  ;GET COMMAND
13348
13349  042222  005737  004644                  TST     NOGO            ;IS GO TO BE GIVEN
13350  042226  001006                          BNE     4$              ;BRANCH IF NO
13351  042230  052777  000001  141622          BIS     #GO,@RHCS1      ;GO
13352
13353                                          ;WAIT FOR DRY BIT IN RHDS1 REGISTER TO SET
13354  042236  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13355  042240  004072                          RHDS1                   ;AND WAIT FOR DRY BIT IN
13356  042242  000200                          DRY                     ;RHDS1 REGISTER
13357                                                                  ;IF ERROR OCCURS HERE
13358                                                                  ;IT MEANS "DRY" DID NOT
13359                                                                  ;SET FOR THE FULL COUNT
13360                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13361                                                                  ;THIS TIME IS APPROXIMATELY
13362                                                                  ;LARGER THAN 500 MILLISECONDS
13363  042244                          4$:
13364                                          ;WAIT FOR RDY BIT IN RHCS1 REGISTER TO SET
13365  042244  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13366  042246  004060                          RHCS1                   ;AND WAIT FOR RDY BIT IN
13367  042250  000200                          RDY                     ;RHCS1 REGISTER
13368                                                                  ;IF ERROR OCCURS HERE
13369                                                                  ;IT MEANS "RDY" DID NOT
13370                                                                  ;SET FOR THE FULL COUNT
13371                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13372                                                                  ;THIS TIME IS APPROXIMATELY
13373                                                                  ;LARGER THAN 500 MILLISECONDS
13374  042252  000200                          RTS     R0
```

```
13375
13376                                   ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
13377                                   ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
13378
13379  042254                           BASECH:
13380  042254  104401  042262                   TYPE    ,65$            ;;TYPE ASCIZ STRING
13381  042260  000424                           BR      64$             ;;GET OVER THE ASCIZ
13382                                   ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS /
13383  042332                           64$:
13384  042332  013746  004060                   MOV     @#RHCS1,-(SP)   ;GET READY TO TYPE OLD BASE
13385  042336  104402                           TYPOC
13386  042340  104401  042346                   TYPE    ,67$            ;;TYPE ASCIZ STRING
13387  042344  000425                           BR      66$             ;;GET OVER THE ASCIZ
13388                                   ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'/
13389  042420                           66$:
13390  042420  104410                           RDOCT
13391  042422  012700  004102                   MOV     #RHDB,R0        ;GET STARTING ADDRESS OF RGISTERS
13392  042426  012701  000024                   MOV     #20.,R1         ;NUMBER OF REGISTERS
13393  042432  042710  177700           1$:     BIC     #^C77,(R0)      ;CLEAR OLD BASE
13394  042436  051620                           BIS     (SP),(R0)+      ;SET NEW BASE
13395  042440  005301                           DEC     R1              ;COUNT
13396  042442  001373                           BNE     1$              ;BRANCH IF 20 NOT DONE
13397  042444  104401  042452                   TYPE    ,69$            ;;TYPE ASCIZ STRING
13398  042450  000417                           BR      68$             ;;GET OVER THE ASCIZ
13399                                   ;;69$:  .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
13400  042510                           68$:
13401  042510  013746  002716                   MOV     @#RPVEC,-(SP)   ;GET READY TO TYPE OLD VECTOR ADDRESS
13402  042514  104402                           TYPOC
13403  042516  104401  042524                   TYPE    ,71$            ;;TYPE ASCIZ STRING
13404  042522  000437                           BR      70$             ;;GET OVER THE ASCIZ
13405                                   ;;71$:  .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR"/
13406  042622                           70$:
13407  042622  104410                           RDOCT
13408  042624  012637  002716                   MOV     (SP)+,@#RPVEC   ;SETUP VECTOR ADDRESS
13409  042630  104401  042636                   TYPE    ,73$            ;;TYPE ASCIZ STRING
13410  042634  000421                           BR      72$             ;;GET OVER THE ASCIZ
13411                                   ;;73$:  .ASCIZ  <15><12>/RESTART PROGRAM FROM 200 OR 210/
13412  042700                           72$:
13413  042700  104401  042706                   TYPE    ,75$            ;;TYPE ASCIZ STRING
13414  042704  000414                           BR      74$             ;;GET OVER THE ASCIZ
13415                                   ;;75$:  .ASCIZ  <15><12>/NEW BASE WILL REMAIN/
13416  042736                           74$:
13417  042736  013746  004060                   MOV     @#RHCS1,-(SP)
13418  042742  104402                           TYPOC
13419  042744  104401  042752                   TYPE    ,77$            ;;TYPE ASCIZ STRING
13420  042750  000415                           BR      76$             ;;GET OVER THE ASCIZ
13421                                   ;;77$:  .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN /
13422  043004                           76$:
13423  043004  013746  002716                   MOV     @#RPVEC,-(SP)
13424  043010  104402                           TYPOC
13425  043012  104401  043020                   TYPE    ,79$            ;;TYPE ASCIZ STRING
13426  043016  000416                           BR      78$             ;;GET OVER THE ASCIZ
13427                                   ;;79$:  .ASCIZ  <15><12>/UNTIL PROGRAM IS RELOADED/
13428  043054                           78$:
13429  043054  000000                           HALT
13430
```

```
13431  043056                          RPVECT:
13432  043056  104401  043064                  TYPE    ,65$            ;;TYPE ASCIZ STRING
13433  043062  000411                  BR      64$             ;;GET OVER THE ASCIZ
13434                                  ;;65$:  .ASCIZ  /TRAPED FROM PC = /
13435  043106                          64$:
13436  043106  104402                  TYPOC                           ;TYPE FROM PC
13437  043110  012777  043056  137600  MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
13438  043116  000000                  HALT                            ;CHANGE TO CONTINUE
13439
13440
13441
13442
```

```
13443                                                                                   ;****************************
13444                                   .SBTTL  SCOPE HANDLER ROUTINE
13445
13446                                   ;;*****************************************************************
13447                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
13448                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
13449                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
13450                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
13451                                   ;*SW14=1         LOOP ON TEST
13452                                   ;*SW11=1         INHIBIT ITERATIONS
13453                                   ;*SW09=1         LOOP ON ERROR
13454                                   ;*SW08=1         LOOP ON TEST IN SWR<7:0>
13455                                   ;*CALL
13456                                   ;*      SCOPE           ;;SCOPE=IOT
13457
13458   043120                          $SCOPE:
13459   043120  005037  004646                 CLR     WAT0            ;WAIT FOR BIT TO SET
13460                                                                   ;USED IN "WAT" WAIT ROUTINE
13461   043124  005037  004644                 CLR NOGO                ;DO GIVE GO IN COMMAND ROUTINE
13462   043130  032777  040000  136002 1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
13463   043136  001111                 BNE     $OVER           ;;YES IF SW14=1
13464                                   ;#####START OF CODE FOR THE XOR TESTER#####
13465   043140  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
13466                                                                   ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
13467   043142  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
13468   043146  012737  043166  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
13469   043154  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
13470   043160  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
13471   043164  000463                 BR      $SVLAD          ;;GO TO THE NEXT TEST
13472   043166  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
13473   043170  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
13474   043174  000423                 BR      7$              ;;LOOP ON THE PRESENT TEST
13475   043176                          6$:;#####END OF CODE FOR THE XOR TESTER#####
13476   043176  032777  000400  135734         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
13477   043204  001404                 BEQ     2$              ;;BR IF NO
13478   043206  127737  135726  001102         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
13479   043214  001462                 BEQ     $OVER           ;;BR IF YES
13480   043216  105737  001103         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
13481   043222  001421                 BEQ     3$              ;;BR IF NO
13482   043224  123737  001115  001103         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
13483   043232  101015                 BHI     3$              ;;BR IF NO
13484   043234  032777  001000  135676         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
13485   043242  001404                 BEQ     4$              ;;BR IF NO
13486   043244  013737  001110  001106 7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
13487   043252  000443                 BR      $OVER
13488   043254  105037  001103         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
13489   043260  005037  001212                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
13490   043264  000415                 BR      1$              ;;ESCAPE TO THE NEXT TEST
13491   043266  032777  004000  135644 3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
13492   043274  001011                 BNE     1$              ;;BR IF YES
13493   043276  005737  001100                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
13494   043302  001406                 BEQ     1$              ;;      INHIBIT ITERATIONS
13495   043304  005237  001104                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
13496   043310  023737  001212  001104         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
13497   043316  002021                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
13498   043320  012737  000001  001104 1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
```

```
13499  043326  013737  043376  001212          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
13500  043334  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
13501  043340  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
13502  043344  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
13503  043350  005037  001214                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
13504  043354  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
13505  043362  013777  001102  135552  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
13506  043370  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
13507  043374  000002                          RTI                     ;;FIXES PS
13508  043376  000004                  $MXCNT: 4                       ;;MAX. NUMBER OF ITERATIONS
```

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

043400                          $TYPDS:
043400  010C046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043402  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043404  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043406  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
043410  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
043412  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
043416  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
043422  100004                        BPL     1$              ;;BR IF INPUT IS POS.
043424  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
043426  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
043434  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
043436  012703  043614                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
043442  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
043446  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
043450  016001  043604                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
043454  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
043456  002402                        BLT     4$              ;;BR IF DONE
043460  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
043462  000774                        BR      3$
043464  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
043466  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
043470  001002                        BNE     5$              ;;FALL THROUGH IF 0
043472  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
043474  100407                        BMI     7$              ;;BR IF YES
043476  106316                5$:     ASLB    (SP)            ;;MSD?
043500  103003                        BCC     6$              ;;BR IF NO
043502  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
043510  052702  000060        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
043514  052702  000040        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
043520  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
043522  005720                        TST     (R0)+           ;;JUST INCREMENTING
043524  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
043530  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
043532  003002                        BGT     8$              ;;GO TO EXIT
043534  010502                        MOV     R5,R2           ;;GET THE LSD
043536  000764                        BR      6$              ;;GO CHANGE TO ASCII
043540  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
043542  100003                        BPL     9$              ;;BR IF NO
043544  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
043552  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
043554  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
043556  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
043560  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
13565  043562  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
13566  043564  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
13567  043566  104401  043614                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
13568  043572  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
13569  043600  012616                          MOV     (SP)+,(SP)
13570  043602  000002                          RTI                     ;;RETURN TO USER
13571  043604  023420                $DTBL:    10000.
13572  043606  001750                          1000.
13573  043610  000144                          100.
13574  043612  000012                          10.
13575  043614  000004                $DBLK:    .BLKW   4
13576                                .SBTTL  TYPE ROUTINE
13577
13578                                ;;******************************************************************
13579                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
13580                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
13581                                ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
13582                                ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
13583                                ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
13584                                ;*
13585                                ;*CALL:
13586                                ;*1) USING A TRAP INSTRUCTION
13587                                ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
13588                                ;*OR
13589                                ;*      TYPE
13590                                ;*      MESADR
13591                                ;*
13592
13593  043624  105737  001157        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
13594  043630  100002                        BPL     1$              ;;BR IF YES
13595  043632  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
13596  043634  000407                        BR      3$              ;;LEAVE
13597  043636  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
13598  043640  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
13599  043644  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
13600  043646  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
13601  043650  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
13602  043652  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
13603  043654  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
13604  043660  000002                        RTI                     ;;RETURN
13605  043662  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
13606  043666  001430                        BEQ     8$
13607  043670  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
13608  043674  001006                        BNE     5$
13609  043676  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
13610  043700  104401                        TYPE                    ;;TYPE A CR AND LF
13611  043702  001223                        $CRLF
13612  043704  105037  044040                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
13613  043710  000755                        BR      2$              ;;GET NEXT CHARACTER
13614  043712  004737  043774        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
13615  043716  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
13616  043722  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
13617  043724  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
13618                                                                ;;AND THE NULL CHAR.
13619  043730  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
13620  043734  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
13621  043736  004737  043774                JSR     PC,$TYPEC       ;;GO TYPE A NULL
13622  043742  105337  044040                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
13623  043746  000770                        BR      7$              ;;LOOP
13624
13625                                  ;HORIZONTAL TAB PROCESSOR
13626
13627  043750  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
13628  043754  004737  043774          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
13629  043760  132737  000007  044040        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
13630  043766  001372                        BNE     9$              ;;TAB STOP
13631  043770  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
13632  043772  000724                        BR      2$              ;;GET NEXT CHARACTER
13633  043774  105777  135150          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
13634  044000  100375                        BPL     $TYPEC
13635  044002  116677  000002  135142        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
13636  044010  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
13637  044016  001003                        BNE     1$              ;;BRANCH IF NO
13638  044020  105037  044040                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
13639  044024  000406                        BR      $TYPEX          ;;EXIT
13640  044026  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
13641  044034  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
13642  044036  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
13643  044040  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
13644  044042  000207          $TYPEX: RTS     PC
13645
13646                                  .SBTTL  TTY INPUT ROUTINE
13647
13648                                  ;;**************************************************************
13649                                  .ENABL  LSB
13650  044044  000000          $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
13651  044046  000000          $TKQIN: .WORD   0               ;;INPUT POINTER
13652  044050  000000          $TKQOUT: .WORD  0               ;;OUTPUT POINTER
13653  044052  000011          $TKQSRT: .BLKB  9.              ;;TTY KEYBOARD QUEUE
13654          044063          $TKQEND=.
13655          044064                  .EVEN
13656
13657                                  ;*TK INITIALIZE ROUTINE
13658                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
13659                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
13660                                  ;
13661                                  ;*CALL:
13662                                  ;*      JSR     PC,$TKINT
13663                                  ;*      RETURN
13664                                  ;
13665  044064  005037  044044          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
13666  044070  012737  044052  044046        MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
13667  044076  013737  044046  044050        MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
13668  044104  012737  044134  000060        MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
13669  044112  012737  000200  000062        MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
13670  044120  005777  135022                TST     @$TKB           ;;CLEAR DONE FLAG
13671  044124  012777  000100  135012        MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
13672  044132  000207                        RTS     PC              ;;RETURN TO CALLER
13673
13674                                  ;*TK SERVICE ROUTINE
13675                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
13676                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
```

```
13677                                   ;*IT IN THE QUEUE.
13678                                   ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
13679                                   ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
13680                                   ;
13681  044134  117746  135006          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
13682  044140  042716  177600                  BIC     #↑C177,(SP)     ;;STRIP THE JUNK
13683  044144  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
13684  044150  001007                          BNE     1$              ;;BRANCH IF NO
13685  044152  104401  044543                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
13686  044156  004737  044064                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
13687  044162  005726                          TST     (SP)+           ;;CLEAN UP STACK
13688  044164  000137  040654                  JMP     OPERSEL         ;;CONTROL C RESTART
13689
13690  044170                          1$:
13691  044170  022737  000011  044044          CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
13692  044176  001004                          BNE     3$              ;;BRANCH IF NO
13693  044200  104401  001216                  TYPE    ,$BELL          ;;RING THE TTY BELL
13694  044204  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
13695  044206  000451                          BR      5$              ;;EXIT
13696  044210  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
13697  044214  001021                          BNE     32$             ;;BRANCH IF NO
13698  044216  005077  134722                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
13699  044222  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
13700  044224  105777  134714          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
13701  044230  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
13702  044232  117746  134710                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
13703  044236  042716  177600                  BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
13704  044242  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
13705  044246  001366                          BNE     31$             ;;BRANCH IF NO
13706  044250  012777  000100  134666          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
13707  044256  000002                          RTI                     ;;RETURN
13708  044260  005237  044044          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
13709  044264  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
13710  044270  002405                          BLT     4$              ;;BRANCH IF YES
13711  044272  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
13712  044276  003002                          BGT     4$              ;;BRANCH IF YES
13713  044300  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
13714  044304  112677  177536          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
13715  044310  005237  044046                  INC     $TKQIN          ;;UPDATE THE POINTER
13716  044314  023727  044046  044063          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
13717  044322  001003                          BNE     5$              ;;BRANCH IF NO
13718  044324  012737  044052  044046          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
13719  044332  000002                  5$:     RTI                     ;;RETURN
13720
13721                                   .DSABL  LSB
13722
13723
13724                                   ;;*****************************************************************
13725                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
13726                                   ;*CALL:
13727                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
13728                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
13729                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
13730                                   ;
13731
13732  044334  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
```

```
13733  044336  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
13734  044344  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
13735  044350  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
13736  044352  012746  044360                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
13737  044356  000002                          RTI                     ;;POP NEW PC AND PS
13738  044360                          64$:
13739  044360  005737  044044          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
13740  044364  001775                          BEQ     1$
13741  044366  005337  044044                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
13742  044372  117766  177452  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
13743  044400  005237  044050                  INC     $TKQOUT         ;;UPDATE THE POINTER
13744  044404  023727  044050  044063          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
13745  044412  001003                          BNE     2$              ;;BRANCH IF NO
13746  044414  012737  044052  044050          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
13747  044422  000002                  2$:     RTI                     ;;RETURN
13748                                  ;;*****************************************************************
13749                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
13750                                  ;*CALL:
13751                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
13752                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
13753                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
13754
13755  044424  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
13756  044426  012703  044532          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
13757  044432  022703  044543          2$:     CMP     #$TTYIN+9.,R3   ;;BUFFER FULL?
13758  044436  101405                          BLOS    4$              ;;BR IF YES
13759  044440  104406                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
13760  044442  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
13761  044444  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
13762  044450  001003                          BNE     3$              ;;SKIP IF NOT
13763  044452  104401  001222          4$:     TYPE    ,$QUES          ;;TYPE A '?'
13764  044456  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
13765  044460  111337  044530          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
13766  044464  104401  044530                  TYPE    ,9$
13767  044470  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
13768  044474  001356                          BNE     2$              ;;LOOP IF NOT RETURN
13769  044476  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
13770  044502  104401  001224                  TYPE    ,$LF            ;;TYPE A LINE FEED
13771  044506  012603                          MOV     (SP)+,R3        ;;RESTORE R3
13772  044510  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
13773  044512  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
13774  044520  012766  044532  000004          MOV     #$TTYIN,4(SP)
13775  044526  000002                          RTI                     ;;RETURN
13776  044530     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
13777  044531     000                          .BYTE   0               ;;TERMINATOR
13778  044532  000011                  $TTYIN: .BLKB   9.              ;;RESERVE 9. BYTES FOR TTY INPUT
13779  044543     136  006503  000012  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
13780  044550  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
13781  044555     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
13782  044562  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
13783  044570  020075     000
13784  044573     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
13785  044600  036440  000040
13786                                                          ;FROM THE TTY
13787
13788
```

```
                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                ;*CHANGE IT TO BINARY.
                                ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
                                ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
                                ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
                                ;*CALL:
                                ;*      RDOCT                   ;;READ AN OCTAL NUMBER
                                ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
                                ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT

044604  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
044606  016666  000004  000002  MOV     4(SP),2(SP)     ;;INPUT NUMBER
044614  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
044616  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
044620  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
044622  104407          1$:     RDLIN                   ;;READ AN ASCIZ LINE
044624  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
044626  010037  044732          MOV     R0,5$           ;;AND SAVE IT
044632  005001                  CLR     R1              ;;CLEAR DATA WORD
044634  005002                  CLR     R2
044636  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
044640  001420                  BEQ     3$              ;;IF ZERO GET OUT
044642  122716  000060          CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
044646  003026                  BGT     4$              ;;IS AN OCTAL DIGIT
044650  122716  000067          CMPB    #'7,(SP)
044654  002423                  BLT     4$
044656  006301                  ASL     R1              ;;*2
044660  006102                  ROL     R2
044662  006301                  ASL     R1              ;;*4
044664  006102                  ROL     R2
044666  006301                  ASL     R1              ;;*8
044670  006102                  ROL     R2
044672  042716  177770          BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
044676  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
044700  000756                  BR      2$              ;;LOOP
044702  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
044704  010166  000012          MOV     R1,12(SP)       ;;SAVE THE RESULT
044710  010237  044742          MOV     R2,$HIOCT
044714  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
044716  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
044720  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
044722  000002                  RTI                     ;;RETURN
044724  005726          4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
044726  105010                  CLRB    (R0)            ;;SET A TERMINATOR
044730  104401                  TYPE                    ;;TYPE UP THRU THE BAD CHAR.
044732  000000          5$:     .WORD   0
044734  104401  001222          TYPE    ,$QUES          ;;"?" "CR" & "LF"
044740  000730                  BR      1$              ;;TRY AGAIN
044742  000000          $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
                                        .SBTTL  ERROR HANDLER ROUTINE
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO $ERRTYP ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
                                        $ERROR:
044744                                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
044744  105237  001103                        BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
044750  001775                                MOV     $TSTNM,@DISPLAY-1;;DISPLAY TEST NUMBER AND ERROR FLAG
044752  013777  001102  134162                BIT     #BIT10,@SWR     ;;BELL ON ERROR?
044760  032777  002000  134152                BEQ     1$              ;;NO - SKIP
044766  001402                                TYPE    ,$BELL          ;;RING BELL
044770  104401  001216                  1$:   INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
044774  005237  001112                        MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
045000  011637  001116                        SUB     #2,$ERRPC
045004  162737  000002  001116                MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
045012  117737  134100  001114                BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
045020  032777  020000  134112                BNE     20$             ;;SKIP TYPEOUTS
045026  001004                                JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
045030  004737  045102                        TYPE    ,$CRLF
045034  104401  001223                  20$:
045040                                  2$:     TST     @SWR            ;;HALT ON ERROR
045040  005777  134074                        BPL     3$              ;;SKIP IF CONTINUE
045044  100001                                HALT                    ;;HALT ON ERROR!
045046  000000                          3$:   BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
045050  032777  001000  134062                BEQ     4$              ;;BR IF NO
045056  001402                                MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
045060  013716  001110                  4$:   TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
045064  005737  001214                        BEQ     5$              ;;BR IF NONE
045070  001402                                MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
045072  013716  001214                  5$:
045076                                          RTI                     ;;RETURN
045076  000002                          ;;*****************************************************
                                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
                                        ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                        ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                        ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
                                        ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
                                        ;*WITH ONLY MINOR CHANGES
                                        ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
                                        ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
                                        ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
                                        ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
                                        ;*AND NOT THE ERROR MESSAGE AND HEADER.
045100  000000                          PRITEM: 0                       ;PREVIOUS ITEM NO. LOCATION
045102  013746  001140                  $ERRTYP: MOV    @#SWR,-(SP)     ;GET SWITCH SETTING
045106  042716  177277                        BIC     #^C500,(SP)     ;KEEP ONLY SWITCH 8 AND 6
```

```
13899  045112  022726  000100                  CMP     #SW06,(SP)+     ;IS 6 SET AND 8 RESET
13900  045116  001001                          BNE     1$              ;IF NOT BRANCH
13901  045120  000402                          BR      2$              ;BRANCH IF SW 6 IS SET AND 8 RESET
13902  045122  000137  046022          1$:     JMP     @#TYPERR        ;JUMP IF SW 8 IS SET
13903                                                                  ;OR IF SW 8 IS RESET AND SW 6 IS RESET
13904  045126  104401  000200          2$:     TYPE    ,CRLF
13905
13906  045132  104401  045140                  TYPE    ,65$            ;;TYPE ASCIZ STRING
13907  045136  000405                          BR      64$             ;;GET OVER THE ASCIZ
13908                                  ;;65$:  .ASCIZ  /   RHWC= /
13909  045152                          64$:
13910  045152  017746  136704                  MOV     @RHWC,-(SP)     ;GET READY TO TYPE RHWC CONTENTS
13911  045156  104402                          TYPOC
13912
13913
13914  045160  104401  045166                  TYPE    ,67$            ;;TYPE ASCIZ STRING
13915  045164  000405                          BR      66$             ;;GET OVER THE ASCIZ
13916                                  ;;67$:  .ASCIZ  /   RHBA= /
13917  045200                          66$:
13918  045200  017746  136660                  MOV     @RHBA,-(SP)     ;GET READY TO TYPE RHBA CONTENTS
13919  045204  104402                          TYPOC
13920
13921
13922  045206  104401  045214                  TYPE    ,69$            ;;TYPE ASCIZ STRING
13923  045212  000406                          BR      68$             ;;GET OVER THE ASCIZ
13924                                  ;;69$:  .ASCIZ  /   RHCS2= /
13925  045230                          68$:
13926  045230  017746  136634                  MOV     @RHCS2,-(SP)    ;GET READY TO TYPE RHCS2 CONTENTS
13927  045234  104402                          TYPOC
13928
13929
13930  045236  104401  045244                  TYPE    ,71$            ;;TYPE ASCIZ STRING
13931  045242  000406                          BR      70$             ;;GET OVER THE ASCIZ
13932                                  ;;71$:  .ASCIZ  /   RHCS1= /
13933  045260                          70$:
13934  045260  017746  136574                  MOV     @RHCS1,-(SP)    ;GET READY TO TYPE RHCS1 CONTENTS
13935  045264  104402                          TYPOC
13936
13937
13938  045266  104401  045274                  TYPE    ,73$            ;;TYPE ASCIZ STRING
13939  045272  000406                          BR      72$             ;;GET OVER THE ASCIZ
13940                                  ;;73$:  .ASCIZ  /   RHDS1= /
13941  045310                          72$:
13942  045310  017746  136556                  MOV     @RHDS1,-(SP)    ;GET READY TO TYPE RHDS1 CONTENTS
13943  045314  104402                          TYPOC
13944
13945
13946  045316  104401  045324                  TYPE    ,75$            ;;TYPE ASCIZ STRING
13947  045322  000406                          BR      74$             ;;GET OVER THE ASCIZ
13948                                  ;;75$:  .ASCIZ  /   RHER1= /
13949  045340                          74$:
13950  045340  017746  136530                  MOV     @RHER1,-(SP)    ;GET READY TO TYPE RHER1 CONTENTS
13951  045344  104402                          TYPOC
13952
13953
13954  045346  104401  045354                  TYPE    ,77$            ;;TYPE ASCIZ STRING
```

```
13955  045352  000406                      BR      76$             ;;GET OVER THE ASCIZ
13956                              ;;77$:   .ASCIZ  /   RHER2= /
13957  045370                      76$:
13958  045370  017746  136524              MOV     @RHER2,-(SP)    ;GET READY TO TYPE RHER2 CONTENTS
13959  045374  104402                      TYPOC
13960
13961
13962  045376  104401  045404              TYPE    ,79$            ;;TYPE ASCIZ STRING
13963  045402  000406                      BR      78$             ;;GET OVER THE ASCIZ
13964                              ;;79$:   .ASCIZ  /   RHER3= /
13965  045420                      78$:
13966  045420  017746  136476              MOV     @RHER3,-(SP)    ;GET READY TO TYPE RHER3 CONTENTS
13967  045424  104402                      TYPOC
13968
13969
13970  045426  104401  045434              TYPE    ,81$            ;;TYPE ASCIZ STRING
13971  045432  000406                      BR      80$             ;;GET OVER THE ASCIZ
13972                              ;;81$:   .ASCIZ  /   RHDST= /
13973  045450                      80$:
13974  045450  017746  136412              MOV     @RHDST,-(SP)    ;GET READY TO TYPE RHDST CONTENTS
13975  045454  104402                      TYPOC
13976
13977
13978  045456  104401  045464              TYPE    ,83$            ;;TYPE ASCIZ STRING
13979  045462  000405                      BR      82$             ;;GET OVER THE ASCIZ
13980                              ;;83$:   .ASCIZ  /   RHCA= /
13981  045476                      82$:
13982  045476  017746  136412              MOV     @RHCA,-(SP)     ;GET READY TO TYPE RHCA CONTENTS
13983  045502  104402                      TYPOC
13984
13985
13986  045504  104401  045512              TYPE    ,85$            ;;TYPE ASCIZ STRING
13987  045510  000405                      BR      84$             ;;GET OVER THE ASCIZ
13988                              ;;85$:   .ASCIZ  /   RHAS= /
13989  045524                      84$:
13990  045524  017746  136346              MOV     @RHAS,-(SP)     ;GET READY TO TYPE RHAS CONTENTS
13991  045530  104402                      TYPOC
13992
13993
13994  045532  104401  045540              TYPE    ,87$            ;;TYPE ASCIZ STRING
13995  045536  000405                      BR      86$             ;;GET OVER THE ASCIZ
13996                              ;;87$:   .ASCIZ  /   RHOF= /
13997  045552                      86$:
13998  045552  017746  136334              MOV     @RHOF,-(SP)     ;GET READY TO TYPE RHOF CONTENTS
13999  045556  104402                      TYPOC
14000
14001
14002  045560  104401  045566              TYPE    ,89$            ;;TYPE ASCIZ STRING
14003  045564  000405                      BR      88$             ;;GET OVER THE ASCIZ
14004                              ;;89$:   .ASCIZ  /   RHMR= /
14005  045600                      88$:
14006  045600  017746  136300              MOV     @RHMR,-(SP)     ;GET READY TO TYPE RHMR CONTENTS
14007  045604  104402                      TYPOC
14008
14009
14010  045606  104401  045614              TYPE    ,91$            ;;TYPE ASCIZ STRING
```

```
14011  045612  000405                          BR      90$             ;;GET OVER THE ASCIZ
14012                                  ;;91$:   .ASCIZ  /   RHLA= /
14013  045626                          90$:
14014  045626  017746  136246                  MOV     @RHLA,-(SP)     ;GET READY TO TYPE RHLA CONTENTS
14015  045632  104402                          TYPOC
14016
14017
14018  045634  104401  045642                  TYPE    ,93$            ;;TYPE ASCIZ STRING
14019  045640  000405                          BR      92$             ;;GET OVER THE ASCIZ
14020                                  ;;93$:   .ASCIZ  /   RHCC= /
14021  045654                          92$:
14022  045654  017746  136220                  MOV     @RHCC,-(SP)     ;GET READY TO TYPE RHCC CONTENTS
14023  045660  104402                          TYPOC
14024
14025
14026  045662  104401  045670                  TYPE    ,95$            ;;TYPE ASCIZ STRING
14027  045666  000406                          BR      94$             ;;GET OVER THE ASCIZ
14028                                  ;;95$:   .ASCIZ  /   RHEC1= /
14029  045704                          94$:
14030  045704  017746  136214                  MOV     @RHEC1,-(SP)    ;GET READY TO TYPE RHEC1 CONTENTS
14031  045710  104402                          TYPOC
14032
14033
14034  045712  104401  045720                  TYPE    ,97$            ;;TYPE ASCIZ STRING
14035  045716  000406                          BR      96$             ;;GET OVER THE ASCIZ
14036                                  ;;97$:   .ASCIZ  /   RHEC2= /
14037  045734                          96$:
14038  045734  017746  136166                  MOV     @RHEC2,-(SP)    ;GET READY TO TYPE RHEC2 CONTENTS
14039  045740  104402                          TYPOC
14040
14041
14042  045742  104401  045750                  TYPE    ,99$            ;;TYPE ASCIZ STRING
14043  045746  000405                          BR      98$             ;;GET OVER THE ASCIZ
14044                                  ;;99$:   .ASCIZ  /   RHDT= /
14045  045762                          98$:
14046  045762  017746  136120                  MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT CONTENTS
14047  045766  104402                          TYPOC
14048
14049
14050  045770  104401  045776                  TYPE    ,101$           ;;TYPE ASCIZ STRING
14051  045774  000405                          BR      100$            ;;GET OVER THE ASCIZ
14052                                  ;;101$:  .ASCIZ  /   RHSN= /
14053  046010                          100$:
14054  046010  017746  136074                  MOV     @RHSN,-(SP)     ;GET READY TO TYPE RHSN CONTENTS
14055  046014  104402                          TYPOC
14056
14057  046016  005037  045100                  CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR ITEM
14058  046022                          TYPERR:
14059  046022  104401  001223                  TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
14060  046026  010046                          MOV     R0,-(SP)        ;SAVE R0
14061  046030  005000                          CLR     R0              ;PICKUP THE ITEM INDEX
14062  046032  153700  001114                  BISB    @#$ITEMB,R0
14063  046036  001004                          BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
14064                                                                  ;TYPE THE PC OF THE ERROR
14065  046040  013746  001116                  MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
14066                                                                  ;ERROR ADDRESS
```

```
14067  046044  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
14068  046046  000454                        BR      10$             ;GET OUT
14069  046050  005300                1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
14070  046052  006300                        ASL     R0              ;       WORK FOR THE ERROR TABLE
14071  046054  006300                        ASL     R0
14072  046056  006300                        ASL     R0
14073  046060  062700  001226              ADD     #$ERRTB,R0      ;FORM TABLE POINTER
14074  046064  020037  045100              CMP     R0,@#PRITEM     ;WAS PREVIOUS ERROR SAME
14075  046070  001002                        BNE     13$             ;BRANCH IF NOT
14076  046072  022020                        CMP     (R0)+,(R0)+     ;POP R0 OVER EM AND DH
14077  046074  000420                        BR      5$
14078  046076  010037  045100        13$:    MOV     R0,@#PRITEM     ;SAVE NEW ERROR ITEM
14079  046102  012037  046112              MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
14080  046106  001404                        BEQ     3$              ;SKIP TYPEOUT IF NO POINTER
14081  046110  104401                        TYPE                    ;TYPE THE "ERROR MESSAGE"
14082  046112  000000                2$:     .WORD   0               ;"ERROR MESSAGE" POINTER GOES HERE
14083  046114  104401  001223              TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
14084  046120  012037  046130        3$:     MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
14085  046124  001404                        BEQ     5$              ;SKIP TYPEOUT IF 0
14086  046126  104401                        TYPE                    ;TYPE THE "DATA HEADER"
14087  046130  000000                4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
14088  046132  104401  001223              TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
14089  046136  010146                5$:     MOV     R1,-(SP)        ;SAVE R1
14090  046140  012001                        MOV     (R0)+,R1        ;PICKUP "DATA TABLE" POINTER
14091  046142  001415                        BEQ     9$              ;BR IF NO DATA TO BE TYPED
14092  046144  012000                        MOV     (R0)+,R0        ;PICKUP "DATA FORMAT" POINTER
14093  046146  105720                6$:     TSTB    (R0)+           ;"OCTAL" OR "DECIMAL"
14094  046150  001003                        BNE     7$              ;BR IF DECIMAL
14095  046152  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
14096  046154  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
14097  046156  000402                        BR      8$
14098  046160                         7$:
14099  046160  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
14100  046162  104405                        TYPDS                   ;GO TYPE--DECIMAL ASCII WITH SIGN
14101  046164  005711                8$:     TST     (R1)            ;IS THERE ANOTHER NUMBER?
14102  046166  001403                        BEQ     9$              ;BR IF NO
14103  046170  104401  046204              TYPE    ,11$            ;TYPE TWO(2) SPACES
14104  046174  000764                        BR      6$              ;LOOP
14105
14106  046176  012601                9$:     MOV     (SP)+,R1        ;RESTORE R1
14107  046200  012600                10$:    MOV     (SP)+,R0        ;"CARRIAGE RETURN" & "LINE FEED"
14108  046202  000207                        RTS     PC              ;RETURN
14109  046204  020040      000       11$:    .ASCIZ  /  /            ;TWO(2) SPACES
14110          046210                        .EVEN
14111                                 ;****************************************************************
14112                                 ;****************************************************************
14113                                 .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
14114
14115                                 ;;****************************************************************
14116                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
14117                                 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
14118                                 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
14119                                 ;*CALL:
14120                                 ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
14121                                 ;*      TYPOS                   ;;CALL FOR TYPEOUT
14122                                 ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
```

```
14123                                   ;*      .BYTE   M               ;;M=1 OR 0
14124                                   ;*                              ;;1=TYPE LEADING ZEROS
14125                                   ;*                              ;;0=SUPPRESS LEADING ZEROS
14126                                   ;*
14127                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
14128                                   ;*$TYPOS OR $TYPOC
14129                                   ;*CALL:
14130                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
14131                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
14132                                   ;*
14133                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
14134                                   ;*CALL:
14135                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
14136                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
14137
14138  046210  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
14139  046214  116637  000001  046433          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
14140  046222  112637  046435          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
14141  046226  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
14142  046232  000406                  BR      $TYPON
14143  046234  112737  000001  046433  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
14144  046242  112737  000006  046435          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
14145  046250  112737  000005  046432  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
14146  046256  010346                  MOV     R3,-(SP)        ;;SAVE R3
14147  046260  010446                  MOV     R4,-(SP)        ;;SAVE R4
14148  046262  010546                  MOV     R5,-(SP)        ;;SAVE R5
14149  046264  113704  046435          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
14150  046270  005404                  NEG     R4
14151  046272  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
14152  046276  110437  046434          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
14153  046302  113704  046433          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
14154  046306  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
14155  046312  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
14156  046314  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
14157  046316  000404                  BR      3$              ;;GO DO MSB
14158  046320  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
14159  046322  006105                  ROL     R5
14160  046324  006105                  ROL     R5
14161  046326  010503                  MOV     R5,R3
14162  046330  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
14163  046332  105337  046434          DECB    $OMODE          ;;TYPE THIS DIGIT?
14164  046336  100016                  BPL     7$              ;;BR IF NO
14165  046340  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
14166  046344  001002                  BNE     4$              ;;TEST FOR 0
14167  046346  005704                  TST     R4              ;;SUPPRESS THIS 0?
14168  046350  001403                  BEQ     5$              ;;BR IF YES
14169  046352  005204          4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
14170  046354  052703  000060          BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
14171  046360  052703  000040  5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
14172  046364  110337  046430          MOVB    R3,8$           ;;SAVE FOR TYPING
14173  046370  104401  046430          TYPE    ,8$             ;;GO TYPE THIS DIGIT
14174  046374  105337  046432  7$:     DECB    $OCNT           ;;COUNT BY 1
14175  046400  003347                  BGT     2$              ;;BR IF MORE TO DO
14176  046402  002402                  BLT     6$              ;;BR IF DONE
14177  046404  005204                  INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
14178  046406  000744                  BR      2$              ;;GO DO THE LAST DIGIT
```

```
14179  046410  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
14180  046412  012604                          MOV     (SP)+,R4        ;;RESTORE R4
14181  046414  012603                          MOV     (SP)+,R3        ;;RESTORE R3
14182  046416  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
14183  046424  012616                          MOV     (SP)+,(SP)
14184  046426  000002                          RTI                     ;;RETURN
14185  046430     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
14186  046431     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
14187  046432     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
14188  046433     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
14189  046434  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
14190                                   .SBTTL  TRAP DECODER
14191
14192                                   ;;**********************************************************************
14193                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
14194                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
14195                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
14196                                   ;*GO TO THAT ROUTINE.
14197
14198  046436  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
14199  046440  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
14200  046444  005740                           TST     -(R0)           ;;BACKUP BY 2
14201  046446  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
14202  046450  006300                           ASL     R0              ;;POSITION FOR INDEXING
14203  046452  016000  046472                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
14204  046456  000200                           RTS     R0              ;;GO TO ROUTINE
14205
14206
14207                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
14208
14209  046460  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
14210  046462  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
14211  046470  000002                           RTI                     ;;RESTORE THE PSW
14212
14213                                   .SBTTL  TRAP TABLE
14214
14215                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
14216                                   ;*BY THE "TRAP" INSTRUCTION.
14217
14218                                   ;       ROUTINE
14219                                   ;       -------
14220  046472  046460                   $TRPAD: .WORD   $TRAP2
14221  046474  043624                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
14222  046476  046234                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
14223  046500  046210                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
14224  046502  046250                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
14225  046504  043400                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
14226
14227
14228  046506  044334                           $RDCHR  ;;CALL=RDCHR    TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
14229  046510  044424                           $RDLIN  ;;CALL=RDLIN    TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
14230  046512  044604                           $RDOCT  ;;CALL=RDOCT    TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
14231  046514  040450                           WAIT.T  ;;CALL=WAT      TRAP+11(104411) DONT ADD ABOVE THIS TRAP
14232
14233
14234
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;**************************************************************
                                        ;POWER DOWN ROUTINE
046516  012737  046662  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
046524  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
046532  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
046534  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
046536  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
046540  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
046542  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
046544  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
046546  017746  132366                  MOV     @$WR,-(SP)      ;;PUSH @SWR ON STACK
046552  010637  046666                  MOV     SP,$SAVR6       ;;SAVE SP
046556  012737  046570  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
046564  000000                          HALT
046566  000776                          BR      .-2             ;;HANG UP

                                        ;;**************************************************************
                                        ;POWER UP ROUTINE
046570  012737  046662  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
046576  013706  046666                  MOV     $SAVR6,SP       ;;GET SP
046602  005037  046666                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
046606  005237  046666          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
046612  001375                          BNE     1$              ;;OF  WORD
046614  012677  132320                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
046620  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
046622  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
046624  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
046626  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
046630  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
046632  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
046634  012737  046516  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
046642  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
046650  104401                          TYPE                    ;REPORT THE POWER FAILURE
046652  046670          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
046654  012716                  MOV     (PC)+,(SP)              ;;RESTART AT BEGIN
046656  005522          $PWRAD: .WORD   BEGIN                   ;;RESTART ADDRESS
046660  000002                  RTI
046662  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
046664  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
046666  000000          $SAVR6: 0                               ;;PUT THE SP HERE
046670  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
046676  000122
                                        .EVEN
```

```
14280                                   ;*****************************************************************
14281                                   ;
14282                                   ;ERROR AND MESSAGE TABLE CONDIMENTS
14283                                   ;
14284                                   ;*****************************************************************
14285
14286  046700  044124  020105  044122  EM1:    .ASCII  /THE RH BASE ADDRESS WAS 172040 INDICATING AN RS BUT THE DRIVE TYPE/<15>
14287  046706  041040  051501  020105
14288  046714  042101  051104  051505
14289  046722  020123  040527  020123
14290  046730  033461  030062  030064
14291  046736  044440  042116  041511
14292  046744  052101  047111  020107
14293  046752  047101  051040  020123
14294  046760  052502  020124  044124
14295  046766  020105  051104  053111
14296  046774  020105  054524  042520
14297  047002  005015
14298  047004  044504  020104  047516          .ASCIZ  /DID NOT CONTAIN 0 OR 1 OR 2 OR 3 OR 4/
14299  047012  020124  047503  052116
14300  047020  044501  020116  020060
14301  047026  051117  030440  047440
14302  047034  020122  020062  051117
14303  047042  031440  047440  020122
14304  047050  000064
14305
14306  047052  044124  020105  044122  EM2:    .ASCII  /THE RH BASE ADDRESS WAS 176700 INDICATING AN RP04,5,6 BUT THE DRIVE TYP
14307  047060  041040  051501  020105
14308  047066  042101  051104  051505
14309  047074  020123  040527  020123
14310  047102  033461  033466  030060
14311  047110  044440  042116  041511
14312  047116  052101  047111  020107
14313  047124  047101  051040  030120
14314  047132  026064  026065  020066
14315  047140  052502  020124  044124
14316  047146  020105  051104  053111
14317  047154  020105  054524  042520
14318  047162  005015
14319  047164  044504  020104  047516          .ASCIZ  /DID NOT CONTAIN 20020,24020,20021,24021,20022, OR 24022/
14320  047172  020124  047503  052116
14321  047200  044501  020116  030062
14322  047206  031060  026060  032062
14323  047214  031060  026060  030062
14324  047222  031060  026061  032062
14325  047230  031060  026061  030062
14326  047236  031060  026062  047440
14327  047244  020122  032062  031060
14328  047252  000062
14329
14330  047254  044124  020105  044122  EM3:    .ASCII  /THE RH BASE ADDRESS WAS 172440 INDICATING A TM02 BUT THE DRIVE TYPE/<15
14331  047262  041040  051501  020105
14332  047270  042101  051104  051505
14333  047276  020123  040527  020123
14334  047304  033461  032062  030064
14335  047312  044440  042116  041511
```

```
14336  047320  052101  047111  020107
14337  047326  020101  046524  031060
14338  047334  041040  052125  052040
14339  047342  042510  042040  044522
14340  047350  042526  052040  050131
14341  047356  006505     012
14342  047361     104  042111  047040          .ASCIZ  /DID NOT CONTAIN 142010/
14343  047366  052117  041440  047117
14344  047374  040524  047111  030440
14345  047402  031064  030460  000060
14346
14347  047410  044124  020105  044122  EM4:    .ASCII  /THE RH BASE ADDRESS WAS 176300 INDICATING MORE THAN ONE RH DEVICE BUT/<
14348  047416  041040  051501  020105
14349  047424  042101  051104  051505
14350  047432  020123  040527  020123
14351  047440  033461  031466  030060
14352  047446  044440  042116  041511
14353  047454  052101  047111  020107
14354  047462  047515  042522  052040
14355  047470  040510  020116  047117
14356  047476  020105  044122  042040
14357  047504  053105  041511  020105
14358  047512  052502  006524     012
14359  047517     124  042510  042040          .ASCII  /THE DRIVE TYPE DID NOT CONTAIN 0,1,2,3,4,20020,24020,/<15><12>
14360  047524  044522  042526  052040
14361  047532  050131  020105  044504
14362  047540  020104  047516  020124
14363  047546  047503  052116  044501
14364  047554  020116  026060  026061
14365  047562  026062  026063  026064
14366  047570  030062  031060  026060
14367  047576  032062  031060  026060
14368  047604  005015
14369  047606  030062  031060  026061          .ASCIZ  /20021,24021,20022,24022,142010/<15><12>
14370  047614  032062  031060  026061
14371  047622  030062  031060  026062
14372  047630  032062  031060  026062
14373  047636  032061  030062  030061
14374  047644  005015     000
14375  047647     111  042116  041511          .ASCIZ  /INDICATING NO RH DEVICE PRESENT/
14376  047654  052101  047111  020107
14377  047662  047516  051040  020110
14378  047670  042504  044526  042503
14379  047676  050040  042522  042523
14380  047704  052116     000
14381
14382  047707     124  042510  052440  EM5:    .ASCII  /THE UNIBUS TIMED OUT FOR EACH OF THE FOLLOWING ADDRESSES, INDICATING/<1
14383  047714  044516  052502  020123
14384  047722  044524  042515  020104
14385  047730  052517  020124  047506
14386  047736  020122  040505  044103
14387  047744  047440  020106  044124
14388  047752  020105  047506  046114
14389  047760  053517  047111  020107
14390  047766  042101  051104  051505
14391  047774  042523  026123  044440
```

```
14392  050002  042116  041511  052101
14393  050010  047111  006507     012
14394  050015     116  020117  044122          .ASCIZ  /NO RH ON THE SYSTEM SO ABORT PROGRAM/
14395  050022  047440  020116  044124
14396  050030  020105  054523  052123
14397  050036  046505  051440  020117
14398  050044  041101  051117  020124
14399  050052  051120  043517  040522
14400  050060  000115
14401
14402  050062  043101  042524  020122  EM6:    .ASCII  /AFTER AN RH CLEAR (RHCS2-BIT #5) RHCS2 DOES NOT HAVE ONLY IR/<15><12>
14403  050070  047101  051040  020110
14404  050076  046103  040505  020122
14405  050104  051050  041510  031123
14406  050112  041055  052111  021440
14407  050120  024465  051040  041510
14408  050126  031123  042040  042517
14409  050134  020123  047516  020124
14410  050142  040510  042526  047440
14411  050150  046116  020131  051111
14412  050156  005015
14413  050160  047101  020104  047125          .ASCIZ  /AND UNIT NUMBER/
14414  050166  052111  047040  046525
14415  050174  042502  000122
14416
14417  050200  043101  042524  020122  EM7:    .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14418  050206  046103  040505  044522
14419  050214  043516  052040  042510
14420  050222  051040  020110  047101
14421  050230  020104  051127  052111
14422  050236  047111  020107  047117
14423  050244  020105  047527  042122
14424  050252  044440  052116  020117
14425  050260  044122  041104  040440
14426  050266  042116  005015
14427  050272  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHDB TO HAVE WRONG VALUE GIVEN IN "BAD"/
14428  050300  020107  052111  041040
14429  050306  041501  026113  041440
14430  050314  052501  042523  020104
14431  050322  044122  041104  052040
14432  050330  020117  040510  042526
14433  050336  053440  047522  043516
14434  050344  053040  046101  042525
14435  050352  043440  053111  047105
14436  050360  044440  020116  041042
14437  050366  042101  000042
14438
14439  050372  043101  042524  020122  EM10:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14440  050400  046103  040505  044522
14441  050406  043516  052040  042510
14442  050414  051040  020110  047101
14443  050422  020104  051127  052111
14444  050430  047111  020107  047117
14445  050436  020105  047527  042122
14446  050444  044440  052116  020117
14447  050452  044122  041104  040440
```

```
14448  050460  042116  005013
14449  050464  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14450  050472  020107  052111  041040
14451  050500  041501  026113  041440
14452  050506  052501  042523  020104
14453  050514  044122  051503  020062
14454  050522  047524  044040  053101
14455  050530  020105  051127  047117
14456  050536  020107  040526  052514
14457  050544  020105  044507  042526
14458  050552  020116  047111  021040
14459  050560  040502  021104      000
14460
14461  050565      101  052106  051105  EM11:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14462  050572  041440  042514  051101
14463  050600  047111  020107  044124
14464  050606  020105  044122  040440
14465  050614  042116  053440  044522
14466  050622  044524  043516  047440
14467  050630  042516  053440  051117
14468  050636  020104  047111  047524
14469  050644  051040  042110  020102
14470  050652  047101  006504      012
14471  050657      122  040505  044504          .ASCIZ  /READING IT BACK, CAUSED RHCS1 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14472  050664  043516  044440  020124
14473  050672  040502  045503  020054
14474  050700  040503  051525  042105
14475  050706  051040  041510  030523
14476  050714  052040  020117  040510
14477  050722  042526  053440  047522
14478  050730  043516  053040  046101
14479  050736  042525  043440  053111
14480  050744  047105  044440  020116
14481  050752  041042  042101  000042
14482
14483  050760  043101  042524  020122  EM12:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14484  050766  046103  040505  044522
14485  050774  043516  052040  042510
14486  051002  051040  020110  047101
14487  051010  020104  051127  052111
14488  051016  047111  020107  047117
14489  051024  020105  047527  042122
14490  051032  044440  052116  020117
14491  051040  044122  041104  040440
14492  051046  042116  005015
14493  051052  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS3 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14494  051060  020107  052111  041040
14495  051066  041501  026113  041440
14496  051074  052501  042523  020104
14497  051102  044122  051503  020063
14498  051110  047524  044040  053101
14499  051116  020105  051127  047117
14500  051124  020107  040526  052514
14501  051132  020105  044507  042526
14502  051140  020116  047111  021040
14503  051146  040502  021104      000
```

```
14504
14505  051153  101  052106  051105  EM13:  .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14506  051160  041440  042514  051101
14507  051166  047111  020107  044124
14508  051174  020105  044122  040440
14509  051202  042116  053440  044522
14510  051210  044524  043516  047440
14511  051216  042516  053440  051117
14512  051224  020104  047111  047524
14513  051232  051040  042110  020102
14514  051240  047101  006504  012
14515  051245  122  040505  044504  .ASCIZ  /READING IT BACK, CAUSED RHBA TO HAVE WRONG VALUE GIVEN IN "BAD"/
14516  051252  043516  044440  020124
14517  051260  040502  045503  020054
14518  051266  040503  051525  042105
14519  051274  051040  041110  020101
14520  051302  047524  044040  053101
14521  051310  020105  051127  047117
14522  051316  020107  040526  052514
14523  051324  020105  044507  042526
14524  051332  020116  047111  021040
14525  051340  040502  021104  000
14526
14527  051345  101  052106  051105  EM14:  .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14528  051352  041440  042514  051101
14529  051360  047111  020107  044124
14530  051366  020105  044122  040440
14531  051374  042116  053440  044522
14532  051402  044524  043516  047440
14533  051410  042516  053440  051117
14534  051416  020104  047111  047524
14535  051424  051040  042110  020102
14536  051432  047101  006504  012
14537  051437  122  040505  044504  .ASCIZ  /READING IT BACK, CAUSED RHBAE TO HAVE WRONG VALUE GIVEN IN "BAD"/
14538  051444  043516  044440  020124
14539  051452  040502  045503  020054
14540  051460  040503  051525  042105
14541  051466  051040  041110  042501
14542  051474  052040  020117  040510
14543  051502  042526  053440  047522
14544  051510  043516  053040  046101
14545  051516  042525  043440  053111
14546  051524  047105  044440  020116
14547  051532  041042  042101  000042
14548
14549  051540  043101  042524  020122  EM15:  .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14550  051546  046103  040505  044522
14551  051554  043516  052040  042510
14552  051562  051040  020110  047101
14553  051570  020104  051127  052111
14554  051576  047111  020107  047117
14555  051604  020105  047527  042122
14556  051612  044440  052116  020117
14557  051620  044122  041104  040440
14560  051626  042116  005015
14559  051632  042522  042101  047111  .ASCIZ  /READING IT BACK, CAUSED RHWC TO HAVE WRONG VALUE GIVEN IN "BAD"/
```

```
14560  051640  020107  052111  041040
14561  051646  041501  026113  041440
14562  051654  052501  042523  020104
14563  051662  044122  041527  052040
14564  051670  020117  040510  042526
14565  051676  053440  047522  043516
14566  051704  053040  046101  042525
14567  051712  043440  053111  047105
14568  051720  044440  020116  041042
14569  051726  042101  000042
14570
14571  051732  043101  042524  020122  EM16:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14572  051740  046103  040505  044522
14573  051746  043516  052040  042510
14574  051754  051040  020110  047101
14575  051762  020104  051127  052111
14576  051770  047111  020107  047117
14577  051776  020105  047527  042122
14578  052004  044440  052116  020117
14579  052012  044122  041104  040440
14580  052020  042116  005015
14581  052024  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14582  052032  020107  052111  041040
14583  052040  041501  026113  041440
14584  052046  052501  042523  020104
14585  052054  044122  051503  020062
14586  052062  047524  044040  053101
14587  052070  020105  051127  047117
14588  052076  020107  040526  052514
14589  052104  020105  044507  042526
14590  052112  020116  047111  021040
14591  052120  040502  021104     000
14592
14593  052125     101  052106  051105  EM17:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
14594  052132  041440  042514  051101
14595  052140  047111  020107  044124
14596  052146  020105  044122  040440
14597  052154  042116  053440  044522
14598  052162  044524  043516  052040
14599  052170  047527  053440  051117
14600  052176  020104  047111  047524
14601  052204  051040  042110  020102
14602  052212  047101  006504     012
14603  052217     122  040505  044504          .ASCIZ  /READING IT BACK, CAUSED RHDB TO HAVE WRONG VALUE GIVEN IN "BAD"/
14604  052224  043516  044440  020124
14605  052232  040502  045503  020054
14606  052240  040503  051525  042105
14607  052246  051040  042110  020102
14608  052254  047524  044040  053101
14609  052262  020105  051127  047117
14610  052270  020107  040526  052514
14611  052276  020105  044507  042526
14612  052304  020116  047111  021040
14613  052312  040502  021104     000
14614
14615  052317     101  052106  051105  EM20:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
```

```
 14616   052324   041440   042514   051101
 14617   052332   047111   020107   044124
 14618   052340   020105   044122   040440
 14619   052346   042116   053440   044522
 14620   052354   044524   043516   052040
 14621   052362   047527   053440   051117
 14622   052370   020104   047111   047524
 14623   052376   051040   042110   020102
 14624   052404   047101   006504      012
 14625   052411      122   040505   044504          .ASCIZ  /READING ONE BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
 14626   052416   043516   047440   042516
 14627   052424   041040   041501   026113
 14628   052432   041440   052501   042523
 14629   052440   020104   044122   051503
 14630   052446   020062   047524   044040
 14631   052454   053101   020105   051127
 14632   052462   047117   020107   040526
 14633   052470   052514   020105   044507
 14634   052476   042526   020116   047111
 14635   052504   021040   040502   021104
 14636   052512      000
 14637
 14638   052513      101   052106   051105  EM21:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 14639   052520   041440   042514   051101
 14640   052526   047111   020107   044124
 14641   052534   020105   044122   040440
 14642   052542   042116   053440   044522
 14643   052550   044524   043516   052040
 14644   052556   047527   053440   051117
 14645   052564   020104   047111   047524
 14646   052572   051040   042110   020102
 14647   052600   047101   006504      012
 14648   052605      122   040505   044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHDB TO HAVE WRONG VALUE GIVEN IN "BAD"/
 14649   052612   043516   044440   020124
 14650   052620   040502   045503   052040
 14651   052626   044527   042503   020054
 14652   052634   040503   051525   042105
 14653   052642   051040   042110   020102
 14654   052650   047524   044040   053101
 14655   052656   020105   051127   047117
 14656   052664   020107   040526   052514
 14657   052672   020105   044507   042526
 14658   052700   020116   047111   021040
 14659   052706   040502   021104      000
 14660
 14661   052713      101   052106   051105  EM22:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 14662   052720   041440   042514   051101
 14663   052726   047111   020107   044124
 14664   052734   020105   044122   040440
 14665   052742   042116   053440   044522
 14666   052750   044524   043516   052040
 14667   052756   047527   053440   051117
 14668   052764   020104   047111   047524
 14669   052772   051040   042110   020102
 14670   053000   047101   006504      012
 14671   053005      122   040505   044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
```

```
 14672  053012  043516  044440  020124
 14673  053020  040502  045503  052040
 14674  053026  044527  042503  020054
 14675  053034  040503  051525  042105
 14676  053042  051040  041510  031123
 14677  053050  052040  020117  040510
 14678  053056  042526  053440  047522
 14679  053064  043516  053040  046101
 14680  053072  042525  043440  053111
 14681  053100  047105  044440  020116
 14682  053106  041042  042101  000042
 14683
 14684  053114  043101  042524  020122  EM23:   .ASCII  /AFTER CLEARING THE RH AND WRITING EIGHT WORD INTO RHDB AND/<15><12>
 14685  053122  046103  040505  044522
 14686  053130  043516  052040  042510
 14687  053136  051040  020110  047101
 14688  053144  020104  051127  052111
 14689  053152  047111  020107  044505
 14690  053160  044107  020124  047527
 14691  053166  042122  044440  052116
 14692  053174  020117  044122  041104
 14693  053202  040440  042116  005015
 14694  053210  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
 14695  053216  020107  052111  041040
 14696  053224  041501  026113  041440
 14697  053232  052501  042523  020104
 14698  053240  044122  051503  020062
 14699  053246  047524  044040  053101
 14700  053254  020105  051127  047117
 14701  053262  020107  040526  052514
 14702  053270  020105  044507  042526
 14703  053276  020116  047111  021040
 14704  053304  040502  021104     000
 14705
 14706  053311     124  042510  051040  EM24:   .ASCII  /THE RH WAS CLEARED AND A PATTERN OF 8 WORDS WERE WRITTEN INTO RHDB/<15>
 14707  053316  020110  040527  020123
 14708  053324  046103  040505  042522
 14709  053332  020104  047101  020104
 14710  053340  020101  040520  052124
 14711  053346  051105  020116  043117
 14712  053354  034040  053440  051117
 14713  053362  051504  053440  051105
 14714  053370  020105  051127  052111
 14715  053376  042524  020116  047111
 14716  053404  047524  051040  042110
 14717  053412  006502     012
 14718  053415     122  040505  044504          .ASCII  /READING RHDB FOR THE "N" TH. TIME GAVE WRONG VALUE IN RHDB/<15><12>
 14719  053422  043516  051040  042110
 14720  053430  020102  047506  020122
 14721  053436  044124  020105  047042
 14722  053444  020042  044124  020056
 14723  053452  044524  042515  043440
 14724  053460  053101  020105  051127
 14725  053466  047117  020107  040526
 14726  053474  052514  020105  047111
 14727  053502  051040  042110  006502
```

```
14728  053510  012
14729  053511  042  021116  044440          .ASCIZ  /"N" IS GIVEN IN WORD NO/
14730  053516  020123  044507  042526
14731  053524  020116  047111  053440
14732  053532  051117  020104  047516
14733  053540  000
14734
14735  053541  124  042510  051040  EM25:   .ASCII  /THE RH WAS CLEARED AND A PATTERN OF 8 WORDS WERE WRITTEN INTO RHDB/<15>
14736  053546  020110  040527  020123
14737  053554  046103  040505  042522
14738  053562  020104  047101  020104
14739  053570  020101  040520  052124
14740  053576  051105  020116  043117
14741  053604  034040  053440  051117
14742  053612  051504  053440  051105
14743  053620  020105  051127  052111
14744  053526  042524  020116  047111
14745  053634  047524  051040  042110
14746  053642  006502  012
14747  053645  101  052106  051105          .ASCIZ  /AFTER READING ALL 8 WORDS, FOLLOWING REGISTER CONTAINED WRONG VALUE GIV
14748  053652  051040  040505  044504
14749  053660  043516  040440  046114
14750  053666  034040  053440  051117
14751  053674  051504  020054  047506
14752  053702  046114  053517  047111
14753  053710  020107  042522  044507
14754  053716  052123  051105  041440
14755  053724  047117  040524  047111
14756  053732  042105  053440  047522
14757  053740  043516  053040  046101
14758  053746  042525  043440  053111
14759  053754  047105  044440  020116
14760  053762  041042  042101  000042
14761
14762  053770  042523  052124  047111  EM33:   .ASCIZ  /SETTING RH CLEAR (RHCS2-BIT #5) CAUSED ERROR REGISTER 1 TO HAVE WRONG V
14763  053776  020107  044122  041440
14764  054004  042514  051101  024040
14765  054012  044122  051503  026462
14766  054020  044502  020124  032443
14767  054026  020051  040503  051525
14768  054034  042105  042440  051122
14769  054042  051117  051040  043505
14770  054050  051511  042524  020122
14771  054056  020061  047524  044040
14772  054064  053101  020105  051127
14773  054072  047117  020107  040526
14774  054100  052514  000105
14775
14776  054104  047101  051040  020110  EM34:   .ASCII  /AN RH CLEAR WAS GIVEN RHER1 WAS CHECKED TO HAVE ZERO. PAT (RHCS2-BIT #4
14777  054112  046103  040505  020122
14778  054120  040527  020123  044507
14779  054126  042526  020116  044122
14780  054134  051105  020061  040527
14781  054142  020123  044103  041505
14782  054150  042513  020104  047524
14783  054156  044040  053101  020105
```

```
 14784   054164   042532   047522   020056
 14785   054172   040520   020124   051050
 14786   054200   041510   031123   041055
 14787   054206   052111   021440   024464
 14788   054214   053440   051501   051440
 14789   054222   052105   005015
 14790   054226   047524   044440   053116          .ASCIZ  /TO INVERT PARITY CHECKING. RHER1 WAS READ BUT DID NOT CONTAIN WHAT IS I
 14791   054234   051105   020124   040520
 14792   054242   044522   054524   041440
 14793   054250   042510   045503   047111
 14794   054256   027107   051040   042510
 14795   054264   030522   053440   051501
 14796   054272   051040   040505   020104
 14797   054300   052502   020124   044504
 14798   054306   020104   047516   020124
 14799   054314   047503   052116   044501
 14800   054322   020116   044127   052101
 14801   054330   044440   020123   047111
 14802   054336   021040   047507   042117
 14803   054344   000042
 14804
 14805   054346   044122   041440   042514  EM35:   .ASCII  /RH CLEAR WAS GIVEN. RHER1 WAS CHECKED. PAT (RHCS2-BIT #4) WAS SET/<15><
 14806   054354   051101   053440   051501
 14807   054362   043440   053111   047105
 14808   054370   020056   044122   051105
 14809   054376   020061   040527   020123
 14810   054404   044103   041505   042513
 14811   054412   027104   050040   052101
 14812   054420   024040   044122   051503
 14813   054426   026462   044502   020124
 14814   054434   032043   020051   040527
 14815   054442   020123   042523   006524
 14816   054450      012
 14817   054451      101   042116   051040          .ASCIZ  /AND RHER1 WAS READ. RHCS1 SHOULD HAVE RDY, SC AND MCPE SET/
 14818   054456   042510   030522   053440
 14819   054464   051501   051040   040505
 14820   054472   027104   051040   041510
 14821   054500   030523   051440   047510
 14822   054506   046125   020104   040510
 14823   054514   042526   051040   054504
 14824   054522   020054   041523   040440
 14825   054530   042116   046440   050103
 14826   054536   020105   042523   000124
 14827
 14828   054544   044122   041440   042514  EM36:   .ASCII  /RH CLEAR WAS GIVEN. RHER1 WAS CHECKED. PAT (RHCS2-BIT #4) WAS SET/<15><
 14829   054552   051101   053440   051501
 14830   054560   043440   053111   047105
 14831   054566   020056   044122   051105
 14832   054574   020061   040527   020123
 14833   054602   044103   041505   042513
 14834   054610   027104   050040   052101
 14835   054616   024040   044122   051503
 14836   054624   026462   044502   020124
 14837   054632   032043   020051   040527
 14838   054640   020123   042523   006524
 14839   054646      012
```

```
14840  054647    122  042510  030522          .ASCII  /RHER1 WAS READ. "1" WAS WRITTEN INTO "TRE" IN RHCS1/<15><12>
14841  054654 053440  051501  051040
14842  054662 040505  027104  021040
14843  054670 021061  053440  051501
14844  054676 053440  044522  052124
14845  054704 047105  044440  052116
14846  054712 020117  052042  042522
14847  054720 020042  047111  051040
14848  054726 041510  030523  005015
14849  054734 047117  041440  042510          .ASCIZ  /ON CHECKING RHCS1 IT DID NOT CONTAIN WHAT IS IN GOOD/
14850  054742 045503  047111  020107
14851  054750 044122  051503  020061
14852  054756 052111  042040  042111
14853  054764 047040  052117  041440
14854  054772 047117  040524  047111
14855  055000 053440  040510  020124
14856  055006 051511  044440  020116
14857  055014 047507  042117     000
14858
14859  055021    122  020110  040527  EM37:   .ASCII  /RH WAS CLEARED. RHER1 WAS CHECKED. PAT (RHCS2-BIT #4) WAS SET/<15><12>
14860  055026 020123  046103  040505
14861  055034 042522  027104  051040
14862  055042 042510  030522  053440
14863  055050 051501  041440  042510
14864  055056 045503  042105  020056
14865  055064 040520  020124  051050
14866  055072 041510  031123  041055
14867  055100 052111  021440  024464
14868  055106 053440  051501  051440
14869  055114 052105  005015
14870  055120 044122  051105  020061          .ASCII  /RHER1 WAS READ. "1" WAS WRITTEN IN "TRE" IN RHCS1/<15><12>
14871  055126 040527  020123  042522
14872  055134 042101  020056  030442
14873  055142 020042  040527  020123
14874  055150 051127  052111  042524
14875  055156 020116  047111  021040
14876  055164 051124  021105  044440
14877  055172 020116  044122  051503
14878  055200 006461     012
14879  055203    101  020116  044122          .ASCII  /AN RH CLEAR WAS GIVEN BUT FOLLOWING REGISTER DID NOT/<15><12>
14880  055210 041440  042514  051101
14881  055216 053440  051501  043440
14882  055224 053111  047105  041040
14883  055232 052125  043040  046117
14884  055240 047514  044527  043516
14885  055246 051040  043505  051511
14886  055254 042524  020122  044504
14887  055262 020104  047516  006524
14888  055270     012
14889  055271    103  047117  040524          .ASCIZ  /CONTAIN WHAT IS IN "GOOD"/
14890  055276 047111  053440  040510
14891  055304 020124  051511  044440
14892  055312 020116  043442  047517
14893  055320 021104     000
14894
14895  055323    101  052106  051105  EM45:   .ASCII  /AFTER AN RH CLEAR "1" WAS WRITTEN IN THE DISK (TAPE) ADDRESS/<15><12>
```

```
 14896  055330  040440  020116  044122
 14897  055336  041440  042514  051101
 14898  055344  021040  021061  053440
 14899  055352  051501  053440  044522
 14900  055360  052124  047105  044440
 14901  055366  020116  044124  020105
 14902  055374  044504  045523  024040
 14903  055402  040524  042520  020051
 14904  055410  042101  051104  051505
 14905  055416  006523     012
 14906  055421     122  043505  051511          .ASCII  /REGISTER. PAT IN RHCS1 WAS SET. ON READING DISK (TAPE) ADDRESS/<15><12>
 14907  055426  042524  027122  050040
 14908  055434  052101  044440  020116
 14909  055442  044122  051503  020061
 14910  055450  040527  020123  042523
 14911  055456  027124  047440  020116
 14912  055464  042522  042101  047111
 14913  055472  020107  044504  045523
 14914  055500  024040  040524  042520
 14915  055506  020051  042101  051104
 14916  055514  051505  006523     012
 14917  055521     122  043505  051511          .ASCIZ  /REGISTER IT DID NOT CONTAIN "1"/
 14918  055526  042524  020122  052111
 14919  055534  042040  042111  047040
 14920  055542  052117  041440  047117
 14921  055550  040524  047111  021040
 14922  055556  021061     000
 14923
 14924  055561     101  052106  051105  EM46:   .ASCII  /AFTER AN RH CLEAR "1" WAS WRITTEN IN THE DISK (TAPE) ADDRESS REGISTER/<
 14925  055566  040440  020116  044122
 14926  055574  041440  042514  051101
 14927  055602  021040  021061  053440
 14928  055610  051501  053440  044522
 14929  055616  052124  047105  044440
 14930  055624  020116  044124  020105
 14931  055632  044504  045523  024040
 14932  055640  040524  042520  020051
 14933  055646  042101  051104  051505
 14934  055654  020123  042522  044507
 14935  055662  052123  051105  005015
 14936  055670  040520  020124  047111          .ASCII  /PAT IN RHCS1 WAS SET AFTER READING DISK (TAPE) ADDRESS REGISTER THE FOL
 14937  055676  051040  041510  030523
 14938  055704  053440  051501  051440
 14939  055712  052105  040440  052106
 14940  055720  051105  051040  040505
 14941  055726  044504  043516  042040
 14942  055734  051511  020113  052050
 14943  055742  050101  024505  040440
 14944  055750  042104  042522  051523
 14945  055756  051040  043505  051511
 14946  055764  042524  020122  044124
 14947  055772  020105  047506  046114
 14948  056000  053517  047111  006507
 14949  056006     012
 14950  056007     122  043505  051511          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 14951  056014  042524  020122  044504
```

```
 14952  056022  020104  047516  020124
 14953  056030  047503  052116  044501
 14954  056036  020116  044127  052101
 14955  056044  044440  020123  047111
 14956  056052  021040  047507  042117
 14957  056060  000042
 14958
 14959  056062  047101  051040  020110  EM54:   .ASCII  /AN RH CLEAR (RHCS2 BIT #5) WAS GIVEN. A16 (RHCS1 BIT #8) WAS SET/<15><1
 14960  056070  046103  040505  020122
 14961  056076  051050  041510  031123
 14962  056104  041040  052111  021440
 14963  056112  024465  053440  051501
 14964  056120  043440  053111  047105
 14965  056126  020056  030501  020066
 14966  056134  051050  041510  030523
 14967  056142  041040  052111  021440
 14968  056150  024470  053440  051501
 14969  056156  051440  052105  005015
 14970  056164  047117  051040  040505          .ASCIZ  /ON READING THE FOLLOWING REGISTER IT DID NOT CONTAIN WHAT IS IN "GOOD"/
 14971  056172  044504  043516  052040
 14972  056200  042510  043040  046117
 14973  056206  047514  044527  043516
 14974  056214  051040  043505  051511
 14975  056222  042524  020122  052111
 14976  056230  042040  042111  047040
 14977  056236  052117  041440  047117
 14978  056244  040524  047111  053440
 14979  056252  040510  020124  051511
 14980  056260  044440  020116  043442
 14981  056266  047517  021104     000
 14982
 14983  056273     101  020116  044122  EM56:   .ASCII  /AN RH CLEAR (RHCS2 BIT #5) WAS GIVEN. A16 (RHCS1 BIT #8) WAS SET/<15><1
 14984  056300  041440  042514  051101
 14985  056306  024040  044122  051503
 14986  056314  020062  044502  020124
 14987  056322  032443  020051  040527
 14988  056330  020123  044507  042526
 14989  056336  027116  040440  033061
 14990  056344  024040  044122  051503
 14991  056352  020061  044502  020124
 14992  056360  034043  020051  040527
 14993  056366  020123  042523  006524
 14994  056374     012
 14995  056375     101  046114  055040          .ASCIZ  /ALL ZEROS WERE WRITTEN IN RHBAE. THEN THE FOLLOWING REGISTER DID NOT HA
 14996  056402  051105  051517  053440
 14997  056410  051105  020105  051127
 14998  056416  052111  042524  020116
 14999  056424  047111  051040  041110
 15000  056432  042501  020056  044124
 15001  056440  047105  052040  042510
 15002  056446  043040  046117  047514
 15003  056454  044527  043516  051040
 15004  056462  043[illegible]  051511  042524
 15005  056470  02[illegible]  044504  020104
 15006  056476  04[illegible]  020124  040510
 15007  056504  042[illegible]  053440  040510
```

```
 15008  056512  020124  051511  044440
 15009  056520  020116  043442  047517
 15010  056526  021104     000
 15011
 15012  056531     101  020116  044122  EM60:   .ASCII  /AN RH CLEAR (RHCS2 BIT #5) WAS GIVEN. A17 (RHCS1 BIT #9) WAS SET/<15><1
 15013  056536  041440  042514  051101
 15014  056544  024040  044122  051503
 15015  056552  020062  044502  020124
 15016  056560  032443  020051  040527
 15017  056566  020123  044507  042526
 15018  056574  027116  040440  033461
 15019  056602  024040  044122  051503
 15020  056610  020061  044502  020124
 15021  056616  034443  020051  040527
 15022  056624  020123  042523  006524
 15023  056632     012
 15024  056633     117  020116  042522          .ASCIZ  /ON READING THE FOLLOWING REGISTER IT DID NOT CONTAIN WHAT IS IN "GOOD"/
 15025  056640  042101  047111  020107
 15026  056646  044124  020105  047506
 15027  056654  046114  053517  047111
 15028  056662  020107  042522  044507
 15029  056670  052123  051105  044440
 15030  056676  020124  044504  020104
 15031  056704  047516  020124  047503
 15032  056712  052116  044501  020116
 15033  056720  044127  052101  044440
 15034  056726  020123  047111  021040
 15035  056734  047507  042117  000042
 15036
 15037  056742  047101  051040  020110  EM62:   .ASCII  /AN RH CLEAR (RHCS2 BIT #8) WAS GIVEN. A17 (RHCS1 BIT #9) WAS SET/<15><1
 15038  056750  046103  040505  020122
 15039  056756  051050  041510  031123
 15040  056764  041040  052111  021440
 15041  056772  024470  053440  051501
 15042  057000  043440  053111  047105
 15043  057006  020056  030501  020067
 15044  057014  051050  041510  030523
 15045  057022  041040  052111  021440
 15046  057030  024471  053440  051501
 15047  057036  051440  052105  005015
 15048  057044  046101  020114  042532          .ASCIZ  /ALL ZEROS WERE WRITTEN IN RHBAE THEN THE FOLLOWING REGISTER DID NOT HAV
 15049  057052  047522  020123  042527
 15050  057060  042522  053440  044522
 15051  057066  052124  047105  044440
 15052  057074  020116  044122  040502
 15053  057102  020105  044124  047105
 15054  057110  052040  042510  043040
 15055  057116  046117  047514  044527
 15056  057124  043516  051040  043505
 15057  057132  051511  042524  020122
 15058  057140  044504  020104  047516
 15059  057146  020124  040510  042526
 15060  057154  053440  040510  020124
 15061  057162  051511  044440  020116
 15062  057170  047507  042117  005015
 15063  057176     000
```

```
15064
15065  057177  000101  020116  044122  EM64:   .ASCII  /AN RH CLEAR (RHCS2-BIT #5) WAS GIVEN. IE (RHCS1 BIT #6) WAS SET/<15><12
15066  057204  041440  042514  051101
15067  057212  024040  044122  051503
15068  057220  026462  044502  020124
15069  057226  032443  020051  040527
15070  057234  020123  044507  042526
15071  057242  027116  044440  020105
15072  057250  051050  041510  030523
15073  057256  041040  052111  021440
15074  057264  024466  053440  051501
15075  057272  051440  052105  005015
15076  057300  047117  051040  040505        .ASCIZ  /ON READING THE FOLLOWING REGISTER IT DID NOT CONTAIN WHAT IS IN "GOOD"/
15077  057306  044504  043516  052040
15078  057314  042510  043040  046117
15079  057322  047514  044527  043516
15080  057330  051040  043505  051511
15081  057336  042524  020122  052111
15082  057344  042040  042111  047040
15083  057352  052117  041440  047117
15084  057360  040524  047111  053440
15085  057366  040510  020124  051511
15086  057374  044440  020116  043442
15087  057402  047517  021104  000000
15088
15089  057407  000101  020116  044122  EM66:   .ASCII  /AN RH CLEAR (RHCS2-BIT #5) WAS GIVEN. IE (RHCS1 BIT #6) WAS SET/<15><12
15090  057414  041440  042514  051101
15091  057422  024040  044122  051503
15092  057430  026462  044502  020124
15093  057436  032443  020051  040527
15094  057444  020123  044507  042526
15095  057452  027116  044440  020105
15096  057460  051050  041510  030523
15097  057466  041040  052111  021440
15098  057474  024466  053440  051501
15099  057502  051440  052105  005015
15100  057510  046101  020114  042532        .ASCIZ  /ALL ZEROS WERE WRITTEN IN RHCS3 THEN THE FOLLOWING REGISTER DID NOT CON
15101  057516  047522  020123  042527
15102  057524  042522  053440  044522
15103  057532  052124  047105  044440
15104  057540  020116  044122  051503
15105  057546  020063  044124  047105
15106  057554  052040  042510  043040
15107  057562  046117  047514  044527
15108  057570  043516  051040  043505
15109  057576  051511  042524  020122
15110  057604  044504  020104  047516
15111  057612  020124  047503  052116
15112  057620  044501  020116  044127
15113  057626  052101  044440  020123
15114  057634  047111  021040  047507
15115  057642  042117  000042
15116
15117  057646  044122  041440  042514  EM70:   .ASCII  /RH CLEAR WAS GIVEN. TWO SUCCESSIVE "GO" (RHCS1-BIT #0) WAS GIVEN/<15><1
15118  057654  051101  053440  051501
15119  057662  043440  053111  047105
```

```
15120  057670  020056  053524  020117
15121  057676  052523  041503  051505
15122  057704  044523  042526  021040
15123  057712  047507  020042  051050
15124  057720  041510  030523  041055
15125  057726  052111  021440  024460
15126  057734  053440  051501  043440
15127  057742  053111  047105  005015
15128  057750  044527  044124  052517          .ASCII  /WITHOUT WAITING FOR THE FIRST "GO" TO COMPLETE. THE FOLLOWING REGISTER/
15129  057756  020124  040527  052111
15130  057764  047111  020107  047506
15131  057772  020122  044124  020105
15132  060000  044506  051522  020124
15133  060006  043442  021117  052040
15134  060014  020117  047503  050115
15135  060022  042514  042524  020056
15136  060030  044124  020105  047506
15137  060036  046114  053517  047111
15138  060044  020107  042522  044507
15139  060052  052123  051105  005015
15140  060060  044504  020104  047516          .ASCIZ  /DID NOT CONTAIN WHAT IS IN "GOOD"/
15141  060066  020124  047503  052116
15142  060074  044501  020116  044127
15143  060102  052101  044440  020123
15144  060110  047111  021040  047507
15145  060116  042117  000042
15146
15147  060122  044122  041440  042514  EM76:   .ASCII  /RH CLEAR WAS GIVEN A 2 WORD WRITE WAS DONE FROM A LOCATION/<15><12>
15148  060130  051101  053440  051501
15149  060136  043440  053111  047105
15150  060144  040440  031040  053440
15151  060152  051117  020104  051127
15152  060160  052111  020105  040527
15153  060166  020123  047504  042516
15154  060174  043040  047522  020115
15155  060202  020101  047514  040503
15156  060210  044524  047117  005015
15157  060216  040524  042507  020104          .ASCII  /TAGED WRFROM AND WITH BAI (RHCS2 BIT #3) SET AT THE END/<15><12>
15158  060224  051127  051106  046517
15159  060232  040440  042116  053440
15160  060240  052111  020110  040502
15161  060246  020111  051050  041510
15162  060254  031123  041040  052111
15163  060262  021440  024463  051440
15164  060270  052105  040440  020124
15165  060276  044124  020105  047105
15166  060304  006504  000012
15167  060307  000117  020106  044124          .ASCIZ  /OF THE WRITE THE FOLLOWING REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
15168  060314  020105  051127  052111
15169  060322  020105  044124  020105
15170  060330  047506  046114  053517
15171  060336  047111  020107  042522
15172  060344  044507  052123  051105
15173  060352  042040  042111  047040
15174  060360  052117  041440  047117
15175  060366  040524  047111  053440
```

```
 15176  060374  040510  020124  051511
 15177  060402  044440  020116  043442
 15178  060410  047517  021104     000
 15179
 15180  060415     122  020110  046103  EM104:  .ASCII  /RH CLEAR WAS GIVEN AN IPCK BIT SHOWN IN "IPCK" WAS SET/<15><12>
 15181  060422  040505  020122  040527
 15182  060430  020123  044507  042526
 15183  060436  020116  047101  044440
 15184  060444  041520  020113  044502
 15185  060452  020124  044123  053517
 15186  060460  020116  047111  021040
 15187  060466  050111  045503  020042
 15188  060474  040527  020123  042523
 15189  060502  006524     012
 15190  060505     132  051105  051517          .ASCII  /ZEROS WERE MOVED INTO RHDB. ON READING RHDB THE FOLLOWING/<15><12>
 15191  060512  053440  051105  020105
 15192  060520  047515  042526  020104
 15193  060526  047111  047524  051040
 15194  060534  042110  027102  047440
 15195  060542  020116  042522  042101
 15196  060550  047111  020107  044122
 15197  060556  041104  052040  042510
 15198  060564  043040  046117  047514
 15199  060572  044527  043516  005015
 15200  060600  042522  044507  052123          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 15201  060606  051105  042040  042111
 15202  060614  047040  052117  041440
 15203  060622  047117  040524  047111
 15204  060630  053440  040510  020124
 15205  060636  051511  044440  020116
 15206  060644  043442  047517  021104
 15207  060652     000
 15208
 15209  060653     122  020110  046103  EM106:  .ASCII  /RH CLEAR WAS GIVEN AN IPCK BIT SHOWN IN "IPCK" WAS SET/<15><12>
 15210  060660  040505  020122  040527
 15211  060666  020123  044507  042526
 15212  060674  020116  047101  044440
 15213  060702  041520  020113  044502
 15214  060710  020124  044123  053517
 15215  060716  020116  047111  021040
 15216  060724  050111  045503  020042
 15217  060732  040527  020123  042523
 15218  060740  006524     012
 15219  060743     132  051105  051517          .ASCII  /ZEROS WERE MOVED INTO RHDB. RHDB WAS READ. AN RH CLEAR WAS GIVEN/<15><1
 15220  060750  053440  051105  020105
 15221  060756  047515  042526  020104
 15222  060764  047111  047524  051040
 15223  060772  042110  027102  051040
 15224  061000  042110  020102  040527
 15225  061006  020123  042522  042101
 15226  061014  020056  047101  051040
 15227  061022  020110  046103  040505
 15228  061030  020122  040527  020123
 15229  061036  044507  042526  006516
 15230  061044     012
 15231  061045     124  042510  043040          .ASCIZ  /THE FOLLOWING REGISTER DID NOT CONTAIN WHAT IS GIVEN IN "GOOD"/
```

```
 15232  061052  046117  047514  044527
 15233  061060  043516  051040  043505
 15234  061066  051511  042524  020122
 15235  061074  044504  020104  047516
 15236  061102  020124  047503  052116
 15237  061110  044501  020116  044127
 15238  061116  052101  044440  020123
 15239  061124  044507  042526  020116
 15240  061132  047111  021040  047507
 15241  061140  042117  000042
 15242
 15243  061144  044122  041440  042514  EM114:  .ASCII  /RH CLEAR WAS GIVEN.  A WRITE CHECK/<15><12>
 15244  061152  051101  053440  051501
 15245  061160  043440  053111  047105
 15246  061166  020056  040440  053440
 15247  061174  044522  042524  041440
 15248  061202  042510  045503  005015
 15249  061210  040527  020123  047504          .ASCIZ  /WAS DONE. THE FOLLOWING REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 15250  061216  042516  020056  044124
 15251  061224  020105  047506  046114
 15252  061232  053517  047111  020107
 15253  061240  042522  044507  052123
 15254  061246  051105  042040  042111
 15255  061254  047040  052117  041440
 15256  061262  047117  040524  047111
 15257  061270  053440  040510  020124
 15258  061276  051511  044440  020116
 15259  061304  043442  047517  021104
 15260  061312     000
 15261
 15262  061313     122  020110  046103  EM116:  .ASCII  /RH CLEAR WAS GIVEN.  A WRITE CHECK/<15><12>
 15263  061320  040505  020122  040527
 15264  061326  020123  044507  042526
 15265  061334  027116  020040  020101
 15266  061342  051127  052111  020105
 15267  061350  044103  041505  006513
 15268  061356     012
 15269  061357     127  051501  042040          .ASCII  /WAS DONE THEN ANOTHER RH CLEAR WAS GIVEN. THE FOLLOWING REGISTER/<15><1
 15270  061364  047117  020105  044124
 15271  061372  047105  040440  047516
 15272  061400  044124  051105  051040
 15273  061406  020110  046103  040505
 15274  061414  020122  040527  020123
 15275  061422  044507  042526  027116
 15276  061430  052040  042510  043040
 15277  061436  046117  047514  044527
 15278  061444  043516  051040  043505
 15279  061452  051511  042524  006522
 15280  061460     012
 15281  061461     104  042111  047040          .ASCIZ  /DID NOT CONTAIN WHAT IS IN "GOOD"/
 15282  061466  052117  041440  047117
 15283  061474  040524  047111  053440
 15284  061502  040510  020124  051511
 15285  061510  044440  020116  043442
 15286  061516  047517  021104     000
 15287
```

```
 15288  061523     117  020116  020101  EM124:  .ASCII  /ON A SILO TEST TO TEST DBL RHCS3 BIT #10 THE FOLLOWING/<15><12>
 15289  061530  044523  047514  052040
 15290  061536  051505  020124  047524
 15291  061544  052040  051505  020124
 15292  061552  041104  020114  044122
 15293  061560  051503  020063  044502
 15294  061566  020124  030443  020060
 15295  061574  044124  020105  047506
 15296  061602  046114  053517  047111
 15297  061610  006507     012
 15298  061613     122  043505  051511          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 15299  061620  042524  020122  044504
 15300  061626  020104  047516  020124
 15301  061634  047503  052116  044501
 15302  061642  020116  044127  052101
 15303  061650  044440  020123  047111
 15304  061656  021040  047507  042117
 15305  061664  000042
 15306  061666  042502  047506  042522  EM132:  .ASCII  /BEFORE DATA TRANSFER COMMAND WAS TO/<15><12>
 15307  061674  042040  052101  020101
 15308  061702  051124  047101  043123
 15309  061710  051105  041440  046517
 15310  061716  040515  042116  053440
 15311  061724  051501  052040  006517
 15312  061732     012
 15313  061733     102  020105  044507          .ASCII  /BE GIVEN THE RP DRIVE STATUS REGISTER DID/<15><12>
 15314  061740  042526  020116  044124
 15315  061746  020105  050122  042040
 15316  061754  044522  042526  051440
 15317  061762  040524  052524  020123
 15318  061770  042522  044507  052123
 15319  061776  051105  042040  042111
 15320  062004  005015
 15321  062006  047516  020124  047503          .ASCIZ  /NOT CONTAIN WHAT IS IN "GOOD"/
 15322  062014  052116  044501  020116
 15323  062022  044127  052101  044440
 15324  062030  020123  047111  021040
 15325  062036  047507  042117  000042
 15326  062044  042502  047506  042522  EM133:  .ASCII  /BEFORE DATA TRANSFER COMMAND WAS TO/<15><12>
 15327  062052  042040  052101  020101
 15328  062060  051124  047101  043123
 15329  062066  051105  041440  046517
 15330  062074  040515  042116  053440
 15331  062102  051501  052040  006517
 15332  062110     012
 15333  062111     102  020105  044507          .ASCII  /BE GIVEN THE RS DRIVE STATUS REGISTER DID/<15><12>
 15334  062116  042526  020116  044124
 15335  062124  020105  051522  042040
 15336  062132  044522  042526  051440
 15337  062140  040524  052524  020123
 15338  062146  042522  044507  052123
 15339  062154  051105  042040  042111
 15340  062162  005015
 15341  062164  047516  020124  047503          .ASCIZ  /NOT CONTAIN WHAT IS IN "GOOD"/
 15342  062172  052116  044501  020116
 15343  062200  044127  052101  044440
```

```
 15344  062206  020123  047111  021040
 15345  062214  047507  042117  000042
 15346  062222  042502  047506  042522  EM134:  .ASCII  /BEFORE DATA TRANSFER COMMAND WAS TO BE GIVEN/<15><12>
 15347  062230  042040  052101  020101
 15348  062236  051124  047101  043123
 15349  062244  051105  041440  046517
 15350  062252  040515  042116  053440
 15351  062260  051501  052040  020117
 15352  062266  042502  043440  053111
 15353  062274  047105  005015
 15354  062300  044124  020105  040515          .ASCII  /THE MAG TAPE DRIVE STATUS REGISTER DID NOT/<15><12>
 15355  062306  020107  040524  042520
 15356  062314  042040  044522  042526
 15357  062322  051440  040524  052524
 15358  062330  020123  042522  044507
 15359  062336  052123  051105  042040
 15360  062344  042111  047040  052117
 15361  062352  005015
 15362  062354  047503  052116  044501          .ASCIZ  /CONTAIN WHAT IS IN RHDS1-GOOD/
 15363  062362  020116  044127  052101
 15364  062370  044440  020123  047111
 15365  062376  051040  042110  030523
 15366  062404  043455  047517  000104
 15367  062412  040527  020123  040527  EM135:  .ASCII  /WAS WAITING FOR A BIT TO SET IN A REGISTER/<15><12>
 15368  062420  052111  047111  020107
 15369  062426  047506  020122  020101
 15370  062434  044502  020124  047524
 15371  062442  051440  052105  044440
 15372  062450  020116  020101  042522
 15373  062456  044507  052123  051105
 15374  062464  005015
 15375  062466  044502  020124  047111          .ASCIZ  /BIT IN QUESTION IS IN "BIT WAITED"/<15><12>
 15376  062474  050440  042525  052123
 15377  062502  047511  020116  051511
 15378  062510  044440  020116  041042
 15379  062516  052111  053440  044501
 15380  062524  042524  021104  005015
 15381  062532     000
 15382  062533     122  043505  051511          .ASCIZ  /REGISTER ADDRESS IN QUESTION IS IN "REG. ADDR"/
 15383  062540  042524  020122  042101
 15384  062546  051104  051505  020123
 15385  062554  047111  050440  042525
 15386  062562  052123  047511  020116
 15387  062570  051511  044440  020116
 15388  062576  051042  043505  020056
 15389  062604  042101  051104  000042
 15390  062612  040527  020123  040527  EM136:  .ASCII  /WAS WAITING FOR A BIT TO RESET IN A REGISTER/<15><12>
 15391  062620  052111  047111  020107
 15392  062626  047506  020122  020101
 15393  062634  044502  020124  047524
 15394  062642  051040  051505  052105
 15395  062650  044440  020116  020101
 15396  062656  042522  044507  052123
 15397  062664  051105  005015
 15398  062670  044502  020124  047111          .ASCIZ  /BIT IN QUESTION IS IN "BIT WAITED"/<15><12>
 15399  062676  050440  042525  052123
```

```
15400   062704  047511  020116  051511
15401   062712  044440  020116  041042
15402   062720  052111  053440  044501
15403   062726  042524  021104  005015
15404   062734     000
15405   062735     122  043505  051511          .ASCIZ  /REGISTER ADDRESS IN QUESTION IS IN "REG. ADDR"/
15406   062742  042524  020122  042101
15407   062750  051104  051505  020123
15408   062756  047111  050440  042525
15409   062764  052123  047511  020116
15410   062772  051511  044440  020116
15411   063000  051042  043505  020056
15412   063006  042101  051104  000042
15413   063014  047117  053440  044522  EM137:  .ASCII  /ON WRITING AND READING THE REGISTER IN "REG. ADDR"/<15><12>
15414   063022  044524  043516  040440
15415   063030  042116  051040  040505
15416   063036  044504  043516  052040
15417   063044  042510  051040  043505
15418   063052  051511  042524  020122
15419   063060  047111  021040  042522
15420   063066  027107  040440  042104
15421   063074  021122  005015
15422   063100  052111  042040  042111          .ASCIZ  /IT DID NOT CONTAIN EXPECTED VALUE/
15423   063106  047040  052117  041440
15424   063114  047117  040524  047111
15425   063122  042440  050130  041505
15426   063130  042524  020104  040526
15427   063136  052514  000105
15428   063142  044122  041440  042514  EM140:  .ASCII  /RH CLEAR WAS GIVEN AN IPCK BIT SHOWN IN "IPCK" WAS SET/<15><12>
15429   063150  051101  053440  051501
15430   063156  043440  053111  047105
15431   063164  040440  020116  050111
15432   063172  045503  041040  052111
15433   063200  051440  047510  047127
15434   063206  044440  020116  044442
15435   063214  041520  021113  053440
15436   063222  051501  051440  052105
15437   063230  005015
15438   063232  042532  047522  020123          .ASCII  /ZEROS WERE MOVED INTO RHDB. ON READING  THE FOLLOWING/<15><12>
15439   063240  042527  042522  046440
15440   063246  053117  042105  044440
15441   063254  052116  020117  044122
15442   063262  041104  020056  047117
15443   063270  051040  040505  044504
15444   063276  043516  020040  044124
15445   063304  020105  047506  046114
15446   063312  053517  047111  006507
15447   063320     012
15448   063321     122  043505  051511          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
15449   063326  042524  020122  044504
15450   063334  020104  047516  020124
15451   063342  047503  052116  044501
15452   063350  020116  044127  052101
15453   063356  044440  020123  047111
15454   063364  021040  047507  042117
15455   063372  000042
```

```
 15456  063374  043101  042524  020122  EM141:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15457  063402  046103  040505  044522
 15458  063410  043516  052040  042510
 15459  063416  051040  020110  047101
 15460  063424  020104  051127  052111
 15461  063432  047111  020107  053524
 15462  063440  020117  047527  042122
 15463  063446  044440  052116  020117
 15464  063454  044122  041104  040440
 15465  063462  042116  005015
 15466  063466  042522  042101  047111          .ASCIZ  /READING IT BACK TWICE, CAUSED RHCS1 TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15467  063474  020107  052111  041040
 15468  063502  041501  020113  053524
 15469  063510  041511  026105  041440
 15470  063516  052501  042523  020104
 15471  063524  044122  051503  020061
 15472  063532  047524  044040  053101
 15473  063540  020105  051127  047117
 15474  063546  020107  040526  052514
 15475  063554  020105  044507  042526
 15476  063562  020116  047111  021040
 15477  063570  040502  021104     000
 15478  063575     101  052106  051105  EM142:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15479  063602  041440  042514  051101
 15480  063610  047111  020107  044124
 15481  063616  020105  044122  040440
 15482  063624  042116  053440  044522
 15483  063632  044524  043516  052040
 15484  063640  047527  053440  051117
 15485  063646  020104  047111  047524
 15486  063654  051040  042110  020102
 15487  063662  047101  006504     012
 15488  063667     122  040505  044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHCS3 TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15489  063674  043516  044440  020124
 15490  063702  040502  045503  052040
 15491  063710  044527  042503  020054
 15492  063716  040503  051525  042105
 15493  063724  051040  041510  031523
 15494  063732  052040  020117  040510
 15495  063740  042526  053440  047522
 15496  063746  043516  053040  046101
 15497  063754  042525  043440  053111
 15498  063762  047105  044440  020116
 15499  063770  041042  042101  000042
 15500  063776  043101  042524  020122  EM143:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15501  064004  046103  040505  044522
 15502  064012  043516  052040  042510
 15503  064020  051040  020110  047101
 15504  064026  020104  051127  052111
 15505  064034  047111  020107  053524
 15506  064042  020117  047527  042122
 15507  064050  044440  052116  020117
 15508  064056  044122  041104  040440
 15509  064064  042116  005015
 15510  064070  042522  042101  047111          .ASCIZ  /READING IT BACK TWICE, CAUSED RHBA TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15511  064076  020107  052111  041040
```

```
 15512  064104  041501  020113  053524
 15513  064112  041511  026105  041440
 15514  064120  052501  042523  020104
 15515  064126  044122  040502  052040
 15516  064134  020117  040510  042526
 15517  064142  053440  047522  043516
 15518  064150  053040  046101  042525
 15519  064156  043440  053111  047105
 15520  064164  044440  020116  041042
 15521  064172  042101  000042
 15522  064176  043101  042524  020122  EM144:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15523  064204  046103  040505  044522
 15524  064212  043516  052040  042510
 15525  064220  051040  020110  047101
 15526  064226  020104  051127  052111
 15527  064234  047111  020107  053524
 15528  064242  020117  047527  042122
 15529  064250  044440  052116  020117
 15530  064256  044122  041104  040440
 15531  064264  042116  005015
 15532  064270  042522  042101  047111          .ASCIZ  /READING IT BACK TWICE, CAUSED RHBAE TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15533  064276  020107  052111  041040
 15534  064304  041501  020113  053524
 15535  064312  041511  026105  041440
 15536  064320  052501  042523  020104
 15537  064326  044122  040502  020105
 15538  064334  047524  044040  053101
 15539  064342  020105  051127  047117
 15540  064350  020107  040526  052514
 15541  064356  020105  044507  042526
 15542  064364  020116  047111  021040
 15543  064372  040502  021104     000
 15544  064377     101  052106  051105  EM145:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15545  064404  041440  042514  051101
 15546  064412  047111  020107  044124
 15547  064420  020105  044122  040440
 15548  064426  042116  053440  044522
 15549  064434  044524  043516  052040
 15550  064442  047527  053440  051117
 15551  064450  020104  047111  047524
 15552  064456  051040  042110  020102
 15553  064464  047101  006504     012
 15554  064471     122  040505  044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHWC TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15555  064476  043516  044440  020124
 15556  064504  040502  045503  052040
 15557  064512  044527  042503  020054
 15558  064520  040503  051525  042105
 15559  064526  051040  053510  020103
 15560  064534  047524  044040  053101
 15561  064542  020105  051127  047117
 15562  064550  020107  040526  052514
 15563  064556  020105  044507  042526
 15564  064564  020116  047111  021040
 15565  064572  040502  021104     000
 15566
 15567  064577     101  042040  053105  EM146:  .ASCII  /A DEVICE BASE ADDRESS DID NOT TIME OUT BUT/<15><12>
```

```
 15568  064604  041511  020105  040502
 15569  064612  042523  040440  042104
 15570  064620  042522  051523  042040
 15571  064626  042111  047040  052117
 15572  064634  052040  046511  020105
 15573  064642  052517  020124  052502
 15574  064650  006524     012
 15575  064653     124  042510  041440          .ASCII  /THE CORRESPONDING VECTOR DID TIME OUT./<15><12>
 15576  064660  051117  042522  050123
 15577  064666  047117  044504  043516
 15578  064674  053040  041505  047524
 15579  064702  020122  044504  020104
 15580  064710  044524  042515  047440
 15581  064716  052125  006456     012
 15582  064723     110  052111  041440          .ASCIZ  /HIT CONTINUE TO REPEAT VECTOR TIMEOUT/
 15583  064730  047117  044524  052516
 15584  064736  020105  047524  051040
 15585  064744  050105  040505  020124
 15586  064752  042526  052103  051117
 15587  064760  052040  046511  047505
 15588  064766  052125     000
 15589  064771     101  042040  053105  EM147:  .ASCII  /A DEVICE BASE ADDRESS DID NOT TIME OUT/<15><12>
 15590  064776  041511  020105  040502
 15591  065004     123  040440  042104
 15592  065012  [illegible]22  051523  042040
 15593  065020  042111  047040  052117
 15594  065026  052040  046511  020105
 15595  065034  052517  006524     012
 15596  065041     102  052125  047040          .ASCIZ  /BUT NO UNIT NUMBERS HAD APPROPRIATE DRIVE TYPES/
 15597  065046  020117  047125  052111
 15598  065054  047040  046525  042502
 15599  065062  051522  044040  042101
 15600  065070  040440  050120  047522
 15601  065076  051120  040511  042524
 15602  065104  042040  044522  042526
 15603  065112  052040  050131  051505
 15604  065120     000
 15605
 15606
 15607  065121     120  004503  044122  DH1:    .ASCII  /PC     RHDT/
 15608  065126  052104
 15609
 15610  065130  041520  052411  044516  DH5:    .ASCII  /PC     UNIBUS ADDRESS ON WHICH TIME OUT OCCURRED/
 15611  065136  052502  020123  042101
 15612  065144  051104  051505  020123
 15613  065152  047117  053440  044510
 15614  065160  044103  052040  046511
 15615  065166  020105  052517  020124
 15616  065174  041517  052503  051122
 15617  065202  042105
 15618
 15619  065204  041520  052011  051505  DH6:    .ASCII  /PC     TEST NO RHCS2   RHCS2/<15><12>
 15620  065212  020124  047516  051040
 15621  065220  041510  031123  051011
 15622  065226  041510  031123  005015
 15623  065234  004411  047507  042117          .ASCIZ  /               GOOD    BAD/
```

```
 15624  065242  041011  042101     000
 15625
 15626  065247     120  004503  042524  DH7:    .ASCII  /PC     TEST    RHDB    RHDB/<15><12>
 15627  065254  052123  051011  042110
 15628  065262  004502  044122  041104
 15629  065270  005015
 15630  065272  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
 15631  065300  042117  041011  042101
 15632  065306     000
 15633
 15634  065307     120  004503  042524  DH11:   .ASCII  /PC     TEST    RHCS1   RHCS1/<15><12>
 15635  065314  052123  051011  041510
 15636  065322  030523  051011  041510
 15637  065330  030523  005015
 15638  065334  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
 15639  065342  042117  041011  042101
 15640  065350     000
 15641
 15642  065351     120  004503  042524  DH12:   .ASCII  /PC     TEST    RHCS3   RHCS3/<15><12>
 15643  065356  052123  051011  041510
 15644  065364  031523  051011  041510
 15645  065372  031523  005015
 15646  065376  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
 15647  065404  042117  041011  042101
 15648  065412     000
 15649
 15650  065413     120  004503  042524  DH13:   .ASCII  /PC     TEST    RHBA    RHBA/<15><12>
 15651  065420  052123  051011  041110
 15652  065426  004501  044122  040502
 15653  065434  005015
 15654  065436  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
 15655  065444  042117  041011  042101
 15656  065452     000
 15657
 15658  065453     120  004503  042524  DH14:   .ASCII  /PC     TEST    RHBAE   RHBAE/<15><12>
 15659  065460  052123  051011  041110
 15660  065466  042501  051011  041110
 15661  065474  042501  005015
 15662  065500  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
 15663  065506  042117  041011  042101
 15664  065514     000
 15665
 15666  065515     120  004503  042524  DH15:   .ASCII  /PC     TEST    RHWC    RHWC/<15><12>
 15667  065522  052123  051011  053510
 15668  065530  004503  044122  041527
 15669  065536  005015
 15670  065540  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
 15671  065546  042117  041011  042101
 15672  065554     000
 15673
 15674  065555     120  004503  042524  DH24:   .ASCII  /PC     TEST    WORD    RHDB    RHDB/<15><12>
 15675  065562  052123  053411  051117
 15676  065570  004504  044122  041104
 15677  065576  051011  042110  006502
 15678  065604     012
 15679  065605     011  047516  047011          .ASCIZ  /       NO      NO      GOOD    BAD/
```

```
 15680  065612  004517  047507  042117
 15681  065620  041011  042101     000
 15682
 15683  065625     120  004503  042524  DH33:   .ASCII  /PC     TEST    RHER1   RHER1/<15><12>
 15684  065632  052123  051011  042510
 15685  065640  030522  051011  042510
 15686  065646  030522  005015
 15687  065652  047011  004517  047507          .ASCIZ  /      NO      GOOD    BAD/
 15688  065660  042117  041011  042101
 15689  065666     000
 15690
 15691  065667     120  004503  042524  DH45:   .ASCII  /PC     TEST    RHDA    RHDA/<15><12>
 15692  065674  052123  051011  042110
 15693  065702  004501  044122  040504
 15694  065710  005015
 15695  065712  047011  004517  047507          .ASCIZ  /      NO      GOOD    BAD/
 15696  065720  042117  041011  042101
 15697  065726     000
 15698
 15699  065727     120  004503  042524  DH104:  .ASCII  /PC     TEST    IPCK    RHCS3   RHCS3/<15><12>
 15700  065734  052123  044411  041520
 15701  065742  004513  044122  051503
 15702  065750  004463  044122  051503
 15703  065756  006463     012
 15704  065761     011  047516  004411          .ASCIZ  /      NO              GOOD    BAD/
 15705  065766  047507  042117  041011
 15706  065774  042101     000
 15707
 15708  065777     120  004503  042524  DH105:  .ASCII  /PC     TEST    IPCK    RHCS2   RHCS2/<15><12>
 15709  066004  052123  044411  041520
 15710  066012  004513  044122  051503
 15711  066020  004462  044122  051503
 15712  066026  006462     012
 15713  066031     011  047516  004411          .ASCIZ  /      NO              GOOD    BAD/
 15714  066036  047507  042117  041011
 15715  066044  042101     000
 15716
 15717  066047     120  004503  042524  DH106:  .ASCII  /PC     TEST    IPCK    RHCS1   RHCS1/<15><12>
 15718  066054  052123  044411  041520
 15719  066062  004513  044122  051503
 15720  066070  004461  044122  051503
 15721  066076  006461     012
 15722  066101     011  047516  004411          .ASCIZ  /      NO              GOOD    BAD/
 15723  066106  047507  042117  041011
 15724  066114  042101     000
 15725
 15726  066117     120  004503  042524  DH111:  .ASCII  /PC     TEST    IPCK    RHBA    RHBA/<15><12>
 15727  066124  052123  044411  041520
 15728  066132  004513  044122  040502
 15729  066140  051011  041110  006501
 15730  066146     012
 15731  066147     011  047516  004411          .ASCIZ  /      NO              GOOD    BAD/
 15732  066154  047507  042117  041011
 15733  066162  042101     000
 15734
 15735  066165     120  004503  042524  DH112:  .ASCII  /PC     TEST    IPCK    RHBAE   RHBAE/<15><12>
```

```
 15736  066172  052123  044411  041520
 15737  066200  004513  044122  040502
 15738  066206  004505  044122  040502
 15739  066214  006505     012
 15740  066217     011  047516  004411          .ASCIZ  /      NO            GOOD    BAD/
 15741  066224  047507  042117  041011
 15742  066232  042101     000
 15743
 15744  066235     120  004503  042524  DH113:  .ASCII  /PC    TEST    IPCK    RHWC    RHWC/<15><12>
 15745  066242  052123  044411  041520
 15746  066250  004513  044122  041527
 15747  066256  051011  053510  006503
 15748  066264     012
 15749  066265     011  047516  004411          .ASCIZ  /      NO            GOOD    BAD/
 15750  066272  047507  042117  041011
 15751  066300  042101     000
 15752  066303     120  004505  042524  DH132:  .ASCII  /PE    TEST    RHDS1   RHDS1/<15><12>
 15753  066310  052123  051011  042110
 15754  066316  030523  051011  042110
 15755  066324  030523  005015
 15756  066330  047011  004517  047507          .ASCIZ  /      NO      GOOD BAD/
 15757  066336  042117  041040  042101
 15758  066344     000
 15759  066345     120  004503  042524  DH135:  .ASCII  /PC    TEST    PC OF   BIT WAIT  REG/<15><12>
 15760  066352  052123  050011  020103
 15761  066360  043117  041011  052111
 15762  066366  053440  044501  020124
 15763  066374  051040  043505  005015
 15764  066402  047011  004517  040527          .ASCIZ  /      NO      WAT     FOR       ADDR/
 15765  066410  004524  047506  004522
 15766  066416  020040  042101  051104
 15767  066424     000
 15768  066425     120  004503  042524  DH137:  .ASCII  /PC    TEST    REG.    GOOD    BAD/<15><12>
 15769  066432  052123  051011  043505
 15770  066440  004456  047507  042117
 15771  066446  041011  042101  005015
 15772  066454  047011  004517  042101          .ASCIZ  /      NO      ADDR    DATA    DATA/
 15773  066462  051104  042011  052101
 15774  066470  004501  040504  040524
 15775  066476     000
 15776
 15777  066477     120  020103  020040  DH146:  .ASCIZ  /PC      BASE    VECTOR/
 15778  066504  020040  041040  051501
 15779  066512  020105  020040  053040
 15780  066520  041505  047524  000122
 15781  066526  041520  020040  020040  DH147:  .ASCIZ  /PC      BASE ADDRESS/
 15782  066534  020040  040502  042523
 15783  066542  040440  042104  042522
 15784  066550  051523     000
 15785
 15786
 15787          066554                          .EVEN
 15788
 15789  066554  001116  001126  000000  DT1:    .WORD   $ERRPC,$BDDAT,0
 15790  066562  001116  004046  004050  DT5:    .WORD   $ERRPC,RSCS1,RPCS1,TMCS1,MIXCS1,0
 15791  066570  004052  004054  000000
```

```
15792  066576  001116  004656  001124  DT6:    .WORD   $ERRPC,TSTNM,$GDDAT,$BDDAT,0
15793  066604  001126  000000
15794  066610  001116  004656  004664  DT24:   .WORD   $ERRPC,TSTNM,SILONM,$GDDAT,$BDDAT,0
15795  066616  001124  001126  000000
15796  066624  001116  004656  004662  DT104:  .WORD   $ERRPC,TSTNM,IP,$GDDAT,$BDDAT,0
15797  066632  001124  001126  000000
15798  066640  001116  004656  004662  DT105:  .WORD   $ERRPC,TSTNM,IP,$GDDAT,$BDDAT,0
15799  066646  001124  001126  000000
15800  066654  001116  004656  004662  DT106:  .WORD   $ERRPC,TSTNM,IP,$GDDAT,$BDDAT,0
15801  066662  001124  001126  000000
15802  066670  001116  004656  004672  DT135:  .WORD   $ERRPC,TSTNM,WAITBT,WAITPC,WAITRE,0
15803  066676  004666  004670  000000
15804  066704  001116  004656  001122  DT137:  .WORD   $ERRPC,TSTNM,$BDADR,$GDDAT,$BDDAT,0
15805  066712  001124  001126  000000
15806  066720  001116  005324  005326  DT146:  .WORD   $ERRPC,TESTDV,TESTVC,0
15807  066726  000000
15808  066730  001116  005324  000000  DT147:  .WORD   $ERRPC,TESTDV,0
15809
15810
15811
15812  066736      000      000         DF1:    .BYTE   0,0
15813  066740      000      000     000 DF5:    .BYTE   0,0,0,0,0
15814  066743      000      000
15815  066745      000      000     000 DF6:    .BYTE   0,0,0,0
15816  066750      000
15817  066751      000      000     000 DF24:   .BYTE   0,0,0,0,0
15818  066754      000      000
15819  066756      000      000     000 DF104:  .BYTE   0,0,0,0,0
15820  066761      000      000
15821  066763      000      000     000 DF105:  .BYTE   0,0,0,0,0
15822  066766      000      000
15823  066770      000      000     000 DF106:  .BYTE   0,0,0,0,0
15824  066773      000      000
15825  066775      000      000     000 DF135:  .BYTE   0,0,0,0,0
15826  067000      000      000
15827  067002      000      000     000 DF137:  .BYTE   0,0,0,0,0
15828  067005      000      000
15829  067007      000      000     000 DF146:  .BYTE   0,0,0
15830  067012      000      000         DF147:  .BYTE   0,0
15831                                       .EVEN
15832          000001                       .END
```

```
ACL    = 000040          2777#
ACU    = 100000          2712#
AOE    = 001000          2644#
APE    = 100000          2586#
AS     = 000016          2809#
ATA    = 100000          2632#  12941
ATABLE  004674           2944#
ATTENT  005244           3028#
AT0    = 000001          2664#
AT1    = 000002          2665#
AT2    = 000004          2666#
AT3    = 000010          2667#
AT4    = 000020          2668#
AT5    = 000040          2669#
AT6    = 000100          2670#
AT7    = 000200          2671#
A16    = 000400          2606#  7365   7370
A17    = 001000          2607#  7485   7490
BA     = 000004          2803#
BAI    = 000010          2559#  7832   7858   10446  10492  11247  11293  11934  11980  12652  12698
BASEAD  004704           2951#
BASECH  042254          13379#
BASEMA  004712           2954#
BASEVC  004714           2957#
BASINX  004766           2971#
BASM    004722           2960#
BASP    004716           2958#
BASPAD  004706           2952#
BAST    004720           2959#
BASTAD  004710           2953#
BEGIN   005522           1395   30974  14272
BEGIN2  005532           1401   3099#
BFEVEN  003000           2828#  9964   9989   10097  10102  10324  10329  10438  10443  10671  10676  10784  10789
                        11011  11016  11239  11244  11358  11363  11471  11476  11698  11703  11926  11931  12050
                        12055  12169  12174  12406  12411
BFODD   003002           2830#  10211  10216  10556  10561  10898  10903  11124  11129  11584  11589  11812  11817
                        12287  12293  12525  12530  12644  12649
BIT0   = 000001          1369#  7193   7201   7390
BIT00  = 000001          1359#  1363   4026
BIT01  = 000002          1358#  1368
BIT02  = 000004          1357#  1367
BIT03  = 000010          1356#  1366
BIT04  = 000020          1355#  1365   13231
BIT05  = 000040          1354#  1364   13234
BIT06  = 000100          1353#  1363
BIT07  = 000200          1352#  1362
BIT08  = 000400          1351#  1361   13476
BIT09  = 001000          1350#  1360   13484  13876
BIT1   = 000002          1368#  7510
BIT10  = 002000          1349#  3708   3710   3716   3742   13861
BIT11  = 004000          1348#  13491
BIT12  = 010000          1347#  3708   3716   3742
BIT13  = 020000          1346#  13868
BIT14  = 040000          1345#  3717   3743   3796   3821   3875   3901   3954   3979   4033   4059   4114   4140
                        13462
BIT15  = 100000          1344#
```

DERHAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
BIT2   = 000004          1367#
BIT3   = 000010          1366#
BIT4   = 000020          1365#   3868
BIT5   = 000040          1364#   3868
BIT6   = 000100          1363#
BIT7   = 000200          1362#   3868
BIT8   = 000400          1361#   3717    3743    3796    3821    3868    3875    3901    3954    3979    4033    4059    4114
                         4140
BIT9   = 001000          1360#   3717    3743    3796    3821    3868    3875    3901    3954    3979    4033    4059    4114
                         4140
BOT    = 000002          2617#  13234   13242   13272
BPIB   = 001400          2741#  13299   13339
BPTVEC= 000014           1376#
BSGIVA  004724           2963#
BSGIVV  004744           2967#
BUFEW   003000           2827#   9529*   9531    9762*   9764
BUFOW   003002           2829#   9063*   9065    9296*   9298
CC     = 000036          2818#
CHECK   040344          12924#
CHECKC  040402          12928   12932#
CHECKT  040366          12929#
CH7    = 010000           2751#
CLDISK  040312           3704    3784    3862    3942    4020    4101    4186    4375    4551    4819    5213    5633    6055
                         6476    6913    7185    7360    7480    7600    7721    7825    7970    8045    8206    8281    8443
                         8583    8745    8885    9050    9119    9283    9352    9516    9585    9749    9818    9978   10091
                        10205   10318   10432   10550   10665   10778   10892   11005   11118   11233   11352   11465   11579
                        11692   11806   11920   12040   12159   12277   12396   12515   12634   12908#
CLR    = 000040          2561#   3719    3745    3798    3823    3877    3903    3956    3981    4035    4061    4116    4142
                         7013   12908
CLRBIT  004654           2929#   3710*   3723    3750    3868*   3881    3908    4026*   4039    4066    4107*   4120    4147
CMNDRP  001572           3358   13175#
CMNDRS  041502           3376   13127#
CMNDTM  041710           3338   12043   12162   12280   12399   12518   12637   13230#
COMAND  004642           2920#  13127*  13148   13175*  13202   13230*  13297   13347
COMND   004640           2919#   3670*   7731    7834    9059    9068    9292    9301    9525    9534    9758    9767    9992
                        10105   10219   10332   10447   10564   10679   10792   10906   11019   11132   11248   11366   11479
                        11592   11706   11820   11935   12043   12059   12162   12178   12280    2296   12399   12415   12518
                        12534   12637   12654
CR     = 000015          1284#  13636   13646
CRLF   = 000200          1285#   3252   13607   13646   13904
CSF    = 000002          2699#
CSU    = 000010          2701#
CSI    = 000000          2801#
CSZ    = 000010          2806#
DA     = 000006          2805#
DB     = 000022          2811#
DBL    = 002000          2581#   7778    9098    9331    9564    9797   10110   10569   10797   11137   11567   11711   12063
                        12300   12419
DC     = 000034          2817#
DCK    = 100000          2650#
DCL    = 000100          2778#
DCLEAR  004142           2896#  13281
DDISP  = 177570          1291#   1474    3128
DEN1   = 000400          2738#
DEN2   = 001000          2739#
DEN4   = 002000          2740#
```

```
DEVPNT  005300          3049#   3213    3245*   3261
DE1     = 000040        2622#
DFF20   = 000002        2618#
DFF5    = 000001        2616#
DF1     066736          1534    1546    1557    1570    15812#
DF104   066756          2214    2247    2253    2259    2265    15819#
DF105   066763          2220    2241    15821#
DF106   066770          2235    2443    15823#
DF135   066775          2413    2424    15825#
DF137   067002          2436    15827#
DF146   067007          2525    15829#
DF147   067012          2534    15830#
DF24    066751          1792    15817#
DF5     066740          1583    15813#
DF6     066745          1595    1608    1622    1636    1650    1664    1678    1692    1706    1720    1734    1748    1762
                        1776    1806    1812    1821    1830    1839    1848    1860    1874    1888    1904    1920    1926
                        1932    1938    1944    1950    1967    1987    1990    1996    2002    2008    2014    2025    2031
                        2042    2048    2059    2065    2076    2082    2093    2099    2110    2116    2129    2135    2141
                        2147    2153    2159    2172    2178    2184    2190    2196    2202    2275    2285    2296    2302
                        2308    2314    2320    2326    2336    2342    2348    2354    2360    2366    2379    2392    2402
                        2456    2471    2486    2501    2516    15815#
DH1     065121          1531    1543    1554    1567    15607#
DH104   065727          2212    2245    15699#
DH105   065777          2218    2239    15708#
DH106   066047          2233    2441    15717#
DH11    065307          1631    1810    1883    1899    1915    1979    2023    2046    2057    2080    2091    2114    2127
                        2170    2294    2340    2451    15634#
DH111   066117          2251    15726#
DH112   066165          2257    15735#
DH113   066235          2263    15744#
DH12    065351          1645    1816    1930    1994    2097    2108    2139    2182    2283    2306    2334    2466    15642#
DH13    065413          1659    1825    1936    2000    2145    2188    2312    2352    2481    15650#
DH132   066303          2374    2387    2400    15752#
DH135   066345          2407    2418    15759#
DH137   066425          2430    15768#
DH14    065453          1673    1834    1942    2006    2029    2040    2063    2074    2151    2194    2318    2358    2496
                        15658#
DH146   066477          2521    15777#
DH147   066526          2531    15781#
DH15    065515          1687    1843    1948    2012    2157    2200    2324    2364    2511    15666#
DH24    065565          1786    15674#
DH33    065625          1855    1869    15683#
DH45    065667          1962    15691#
DH5     065130          1579    15610#
DH6     065204          1590    1617    1701    1729    1757    1771    1801    1924    1988    2133    2176    2273    2300
                        2346    15619#
DH7     065247          1603    1715    1743    15626#
DIGB    = 000004        2619#
DISPLA  001142          1474#   3128*   3136*   13505*  13860*
DISPRE  000174          1392#   3136
DLT     = 100000        2572#
DL64    = 000020        2621#
DMD     = 000001        2654#
DPEEW   = 020000        2584#
DPEOW   = 040000        2585#
DPR     = 000400        2625#   12942   13130   13182   13232
```

```
DRY    = 000200          2624#  12942   13130   13182   13232   13288   13331   13355
DS     = 000012          2807#
DSWR   = 177570          1290#   1473    3127
DT     = 000026          2817#
DTE    = 010000          2647#
DTSY   = 001000          2660#
DT1      066554          1533    1545    1556    1569   15789#
DT104    066624          2213    2246    2252    2258    2264   15796#
DT105    066640          2219    2240   15798#
DT106    066654          2234    2442   15800#
DT135    066670          2412    2423   15802#
DT137    066704          2435   15804#
DT146    066720          2524   15806#
DT147    066730          2533   15808#
DT24     066610          1791   15794#
DT5      066562          1582   15790#
DT6      066576          1594    1607    1621    1635    1649    1663    1677    1691    1705    1719    1733    1747    1761
                         1775    1805    1811    1820    1829    1838    1847    1859    1873    1887    1903    1919    1935
                         1931    1937    1943    1949    1966    1983    1989    1995    2001    2007    2013    2024    2030
                         2041    2047    2058    2064    2075    2081    2092    2098    2109    2115    2128    2134    2140
                         2146    2152    2158    2171    2177    2183    2189    2195    2201    2274    2284    2295    2301
                         2307    2313    2319    2325    2335    2341    2347    2353    2359    2365    2378    2391    2401
                         2455    2470    2485    2500    2515   15792#
DVA    = 004000          2609#   3708    3709    3716    3742    4264    4706    5088    5501    5924    6345    6778    7018
                         7370    7437    7490    7557    7610    7677    7738    7839    7980    8051    8216    8287    8454
                         8521    8589    8756    8823    8891    9125    9358    9591    9824   10016   10130   10243   10356
                        10472   10589   10703   10817   10930   11043   11157   11273   11390   11503   11617   11731   11844
                        11960   12083   12201   12320   12439   12557   12678   12934
ECH    = 000100          2641#
ECI    = 004000          2725#  13200
EC1    = 000044          2821#
EC2    = 000046          2822#
EMTVEC= 000030           1379#   3112*   3113*
EM1      046700          1527   14286#
EM10     050372          1611   14439#
EM104    060415          2205    2217   15180#
EM106    060653          2223    2238    2247    2250    2256    2262   15209#
EM11     050565          1625   14461#
EM114    061144          2268    2278   15243#
EM116    061313          2288    2299    2305    2311    2317    2323   15262#
EM12     050760          1639   14483#
EM124    061523          2329    2339    2345    2351    2357    2363   15288#
EM13     051153          1653   14505#
EM132    061666          2368   15306#
EM133    062044          2381   15326#
EM134    062222          2394   15346#
EM135    062412          2404   15367#
EM136    062612          2415   15390#
EM137    063014          2426   15413#
EM14     051345          1657   14527#
EM140    063142          2440   15428#
EM141    063374          2445   15456#
EM142    063575          2460   15478#
EM143    063776          2475   15500#
EM144    064176          2490   15522#
EM145    064377          2505   15544#
```

```
EM146   064577          2518   15567#
EM147   064771          2528   15589#
EM15    051540          1681   14549#
EM16    051732          1695   14571#
EM17    052126          1709   14593#
EM2     047052          1537   14306#
EM20    052317          1723   14615#
EM21    052513          1737   14638#
EM22    052713          1751   14661#
EM23    053114          1765   14684#
EM24    053311          1779   14706#
EM25    053541          1795    1809    1815    1824    1833    1842   14735#
EM3     047254          1549   14330#
EM33    053770          1851   14762#
EM34    054104          1863   14776#
EM35    054346          1877   14805#
EM36    054544          1891   14828#
EM37    055021          1907    1923    1929    1935    1941    1947   14859#
EM4     047410          1560   14347#
EM45    055323          1953   14895#
EM46    055561          1970    1987    1993    1999    2005    2011   14924#
EM5     047707          1573   14382#
EM54    056062          2017    2028   14959#
EM56    056273          2034    2045   14983#
EM6     050062          1586   14402#
EM60    056531          2051    2062   15012#
EM62    056742          2068    2079   15037#
EM64    057177          2085    2096    2102    2113   15065#
EM66    057407         15089#
EM7     050200          1598   14417#
EM70    057646          2119    2132    2138    2144    2150    2156   15117#
EM76    060122          2162    2175    2181    2187    2193    2199   15147#
EPAR  = 000010          2733#  13300   13340
ERFLG$  005240          3022#
ERR   = 040000          2631#
ERRVEC= 000004          1372#   3125    3126*   3137*   3161*   3162*   3265*   3314*   3579*   3580*  13467   13468*  13470*
                       13473*
ER1   = 000014          2808#
ER2   = 000040          2819#
ER3   = 000042          2820#
EXT1  = 000001          2679#
EXT10 = 000010          2682#
EXT2  = 000002          2680#
EXT20 = 000020          2683#
EXT4  = 000004          2681#
EXT40 = 000040          2684#
FBAE    005466          3083#   3341*   3361*   3379*   3671
FBASEA  005426          3079#   3332*   3398    3605    3661
FC    = 000006          2804#
FCS3    005476          3084#   3343*   3363*   3381*   3672
FDEVIC  005366          3074#   3326*   3336    3346    3348    3350    3352    3354    3356    3366    3368    3370    3372
                        3374    3404    3419    3426    3433    3435*   3441    3443*   3449    3451*   3457    3459*   3465
                        3467*   3473    3475*   3481    3483*   3489    3491*   3497    3499*   3607    3616    3623    3630
                        3637
FDRIVR  005456          3082#   3338*   3358*   3376*   3670
FEN   = 000200          2705#
```

DERHAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
FER    = 000020          2639#
FIRST    005242          3023#   3149    3158*
FMT1   = 000020          2734#
FMT2   = 000040          2735#
FMT22  = 010000          2726#  13200
FMT4   = 000100          2736#
FMT8   = 000200          2737#
FORBAE   004776          2975#
FRHNM    005436          3080#   3333*   3334*   3335*   3392    3594   12762
.SLAVE   005446          3081#   3312*   3410    3613    3669
FTYPE    005406          3077#   3330*
FUNITN   005376          3076#   3328*   3329*   3511    3651    3667
FUTABL   004134          2892#
FVECTR   005416          3078#   3331*   3517    3657    3668
GIVE     005006          2982#
GNS    = ****** U        1391    3156    3221    3225    3236    3242    3248    3390    3396    3402    3408    3423    3430
                         3438    3446    3454    3462    3470    3478    3486    3494    3502    3509    3515    3521    3525
                         3588    3592    3599    3603    3611    3620    3627    3634    3641    3649    3655   12756   12766
                        12772   12778   12782   12845   12851   12869   12873   12879   12884   12890   12896   13050   13056
                        13063   13067   13071   13078   13082   13382   13388   13399   13405   13411   13415   13421   13427
                        13434   13908   13916   13924   13932   13940   13948   13956   13964   13972   13980   13988   13996
                        14004   14012   14020   14028   14036   14044   14052   14221   14222   14223   14224   14225   14228
                        14229   14230   14231
GO     = 000001          2603#   3708    3716    3742    7733    7734   13151   13178   13205   13246   13282   13305   13351
GRV    = 000010          2620#
HCE    = 000200          2642#
HCI    = 002000          2724#
HCRC   = 000400          2643#
HERADD   005516          3093#  12886
HT     = 000011          1282#  13605   13646
IAE    = 002000          2645#
IE     = 000100          2580#   2604#   7605    7610    7630
ILF    = 000001          2635#
ILLEGL   004206          2914#
ILR    = 000002          2636#
IOTVEC = 000020          1377#   3110*   3111*
IP       004662          2936#   7973*   8209*   8446*   8748*  15796   15798   15800
IPCK0  = 000001          2576#   7975
IPCK1  = 000002          2577#   8211
IPCK2  = 000004          2578#   8448
IPCK3  = 000010          2579#   8750
IR     = 000100          2562#   3789    4191    4244    4381    4433    4556    4592    4644    4685    4824    5065    5219
                         5477    5639    5900    6061    6321    6482    6753    7043    7242    7858    8022    8074    8259
                         8310    8497    8571    8612    8799    8843    8914    9075    9148    9308    9381    9541    9614
                         9774    9847   10035   10149   10262   10375   10492   10608   10722   10836   10949   11062   11176
                        11293   11409   11522   11636   11750   11863   11980   12102   12220   12339   12458   12576   12698
IXE    = 004000          2709#
KIPAR0 = 172340          1435#
KIPAR1 = 172342          1436#
KIPAR2 = 172344          1437#
KIPAR3 = 172346          1438#
KIPAR4 = 172350          1439#
KIPAR5 = 172352          1440#
KIPAR6 = 172354          1441#
KIPAR7 = 172356          1442#
KIPDR0 = 172300          1424#
```

DERHAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
KIPDR1= 172302          1425#
KIPDR2= 172304          1426#
KIPDR3= 172306          1427#
KIPDR4= 172310          1428#
KIPDR5= 172312          1429#
KIPDR6= 172314          1430#
KIPDR7= 172316          1431#
LA    = 000020          2810#
LBT   = 002000          2627#  12941
LERADD  005514          3092#  12892
LF    = 000012          1283#  13640  13646
LOPCT   005000          2976#
LOPCT1  005002          2977#
LTRY    005320          3051#   3228*   3532*   3543*
MCLK  = 000002          2655#
MCPE  = 020000          2610#   3708    3716    3742
MDPE  = 000400          2565#   8022    8258    8497    8541    8799    8843
MEMERR  005520          3094#  12898
MHS   = 001000          2707#
MINX  = 000004          2656#
MIXCS1  004054          2836#  15790
MMVEC = 000250          1413#
MOL   = 010000          2629#  12942  13130  13182  13232
MPE   = 000400          2564#
MR    = 000024          2812#
MRD   = 000020          2658#
MSE   = 000020          2702#
MSTCK = 000010          2657#
MWR   = 000040          2659#
MXF   = 001000          2566#
NED   = 010000          2569#   3293
NEM   = 004000          2568#
NHS   = 002000          2708#
NML   = 000300          2742#  13299  13339
NMRHS   004764          2970#
NOGO    004644          2921#   7730*  13149  13203  13349  13461*
NOPERA  004134          2893#
NTYPNT  005362          3069#   3281*   3284    3288    3326    3550*
OCYL  = 100000          2780#
OF    = 000032          2815#
OFREV = 000200          2723#
OFSETC  004166          2906#
OF100 = 000004          2718#
OF200 = 000010          2719#
OF25  = 000001          2716#
OF400 = 000020          2720#
OF50  = 000002          2717#
OF800 = 000040          2721#
OPERSE  040654         13045#  13688
OPI   = 020000          2648#
OR    = 000200          2563#   4216    4406    4582    4592    4618    4644    4855    4864    5250    5260    5672    5682
                        6094    6104    6517    6527    7757    8497    8799    9075    9308    9541    9774   11409   11522
                       11636   11750   11863   11980
PAR   = 000010          2638#
PARE70= 000114          3095#   3159*   3160*
PARITY  037736          3159   12866#
```

```
PAT    = 000020          2560#   6937    7197    7242
PC     =%000007          1303#   3147*   3704*   3784*   3862*   3942*   4020*   4101*   4186*   4375*   4551*   4819*   5213*
                         5633*   6055*   6476*   6913*   7185*   7360*   7480*   7600*   7721*   7825*   7970*   8045*   8206*
                         8281*   8443*   8583*   8745*   8885*   9050*   9119*   9283*   9352*   9516*   9585*   9749*   9818*
                         9978*  10091*  10205*  10318*  10432*  10550*  10665*  10778*  10892*  11005*  11118*  11233*  11352*
                        11465*  11578*  11692*  11806*  11920*  12040*  12159*  12277*  12396*  12515*  12634*  12804*  12807*
                        12817*  12822   12912*  12948*  13114*  13614*  13621*  13628*  13642*  13644*  13672*  13686*  13870*
                        14108*  14271
PCJSR   040342          12919#  12924*  12925*  12929*  12930*
PE     = 020000          2570#
PERMX   005264          3038#
PERMXV  005274          3043#
PERNUM  005276          3045#
PERRP   005260          3036#
PERRPV  005270          3041#
PERRS   005256          3035#   3211
PERRSV  005266          3040#
PERTU   005262          3037#
PERTUV  005272          3042#
PGE    = 002000          2567#   7757
PIP    = 020000          2630#  13250   13261   13309   13320
PIRQ   = 177772          1289#
PIRQVE= 000240           1383#
PKACK   004172          2908#  13177
PLU    = 020000          2711#
POINTT  005360          3068#   3280*
PRE    = 000020          2776#
PRESEN  004056          2837#
PRITEM  045100          3148*   3674*  13047*  13896#  14057*  14074   14078*
PROG   = 001000          2626#
PR0    = 000000          1306#
PR1    = 000040          1307#
PR2    = 000100          1308#
PR3    = 000140          1309#
PR4    = 000200          1310#
PR5    = 000240          1311#
PR6    = 000300          1312#
PR7    = 000340          1313#
PS     = 177776          1286#   1287    3141*  12753*  13046*
PSEL   = 002000          2608#
PSU    = 000001          2773#
PSW    = 177776          1287#
PUTREG  041442          13101#
PWRVEC= 000024           1378#   3116*   3117*  14239*  14240*  14249*  14255*  14267*  14268*
RDCHR  = 104406         13759   14228#
RDLIN  = 104407         13809   14225#
RDOCT  = 104410          3227    3244    3253   13073   13084   13390   13407   14230#
RDY    = 000200          2605#   3708    3709    3716    3742    4264    4706    5088    5501    5924    6345    6778    7018
                         7370    7437    7490    7557    7610    7677    7839    7980    8051    8216    8287    8454    8521
                         8589    8756    8823    8891    9125    9358    9591    9824   10016   10130   10243   10356   10472
                        10589   10703   10817   10930   11043   11157   11273   11390   11503   11617   11731   11844   11960
                        12083   12201   12320   12439   12557   12678   12934   13156   13210   13367
READAT  004160          2903#  10680   10793   10907   11020   11133   11249
READIN  004174          2909#
RECALI  004140          2895#
REFOR   004162          2904#
```

```
REINTQ  004424          2918#
RELEAS  004144          2897#
RESVEC= 000010          1373#
RETCL   004170          2907#
REVRED  004200          2911#  12060  12179  12297  12416  12535  12655  13297
REVWRT  004204          2913#  11367  11480  11593  11707  11821  11936
REWIND  004176          2910#  13245
RHAS    004076          2867#  13990
RHBA    004064          2860#   4037*  4038   4043   4063*  4064   4069   4301   4473   4745   5131   5546   5969
                        6390    6825   7096   7285   7727*  7833*  7899   8123   8359   8661   8963   9057*  9065*
                        9197    9290*  9298*  9430   9523*  9531*  9663   9756*  9764*  9896   9989*  10102* 10216*
                       10329*  10443* 10561* 10676* 10789* 10903* 11016* 11129* 11244* 11363* 11476* 11589* 11703*
                       11817*  11931* 12055* 12174* 12292* 12411* 12530* 12649* 13918
RHBAE   004130          2882#   3671*  4118*  4119   4124   4144*  4145   4150   4319   4491   4763   5151   5567
                        5990    6411   6847   7120   7304   7392   7410*  7416   7512   7530*  7536   7917   8145
                        8381    8683   8985   9219   9452   9685   9918
RHCA    004114          2876#  13197*  13982
RHCC    004100          2868#  13107   14022
RHCCA   004116          2877#
RHCS1   004060          2858#   3272   3308*  3549*  3554*  3660   3721*  3722   3727   3747*  3748   3753   4266
                        4708    5090   5503   5926   6347   6780   6964   6982*  6989   7020   7224   7365*  7372
                        7439    7485*  7492   7559   7605*  7612   7679   7733*  7734*  7740   7841   7982   8053
                        8218    8289   8456   8523   8591   8758   8825   8893   9127   9360   9593   9826   10014*
                       10018   10128* 10132  10241* 10245  10354* 10358  10470* 10474  10587* 10591  10701* 10705
                       10815*  10819  10928* 10932  11041* 11045  11155* 11159  11271* 11275  11388* 11392  11501*
                       11505   11615* 11619  11729* 11733  11842* 11846  11958* 11962  12081* 12085  12199* 12203
                       12318*  12322  12437* 12441  12555* 12559  12676* 12680  12768  12910* 13148* 13151* 13155
                       13179*  13202* 13205* 13209  13245* 13246* 13283* 13304* 13305* 13347* 13351* 13366  13384
                       13417   13934
RHCS2   004070          2864#   3278*  3293   3328   3553*  3555   3719*  3745*  3798*  3800*  3801   3805   3823*
                        3825*   3826   3830   3877*  3903*  3956*  3981*  4035*  4061*  4116*  4142*  4194   4215
                        4247    4384   4405   4436   4559   4581   4595   4617   4647   4688   4827   4854   4867
                        5068    5222   5249   5263   5480   5642   5671   5685   5903   6064   6093   6107   6324
                        6485    6516   6530   6756   6937*  7013*  7014*  7046   7197*  7245   7760   7832*  7861
                        8025    8077   8261   8313   8500   8544   8615   8802   8846   8917   9078   9151   9311
                        9384    9544   9617   9777   9850   10038  10152  10265  10378  10446* 10495  10611  10725
                       10839   10952  11065  11179  11247* 11296  11412  11525  11639  11753  11866  11934* 11983
                       12105   12223  12342  12461  12579  12652* 12701  12908* 12909* 13926
RHCS3   004132          2883#   3672*  3879*  3880   3885   3905*  3906   3911   4284   4455   4727   5111   5525
                        5948    6369   6803   7072   7266   7632   7650*  7656   7780   7881   7975*  8003   8101
                        8211*   8239   8337   8448*  8478   8565   8639   8750*  8780   8867   8941   9100   9175
                        9333    9408   9566   9641   9799   9874   9999   10112  10226  10339  10454  10571  10686
                       10799   10913  11026  11139  11255  11373  11486  11599  11713  11827  11942  12065  12184
                       12302   12421  12540  12660
RHDA    004066          2862#   7193*  7203   13145*
RHDB    004102          2870#   4211*  4228   4401*  4418   4577*  4613*  4629   4669   4847*  4891   4913   4935
                        4957    4979   5001   5023   5045   5243*  5288   5312   5336   5360   5384   5408   5432
                        5456    5665*  5711   5735   5759   5783   5807   5831   5855   5879   6086*  6132   6156
                        6180    6204   6228   6252   6276   6300   6509*  6556   6581   6606   6631   6656   6681
                        6706    6731   7977*  7997   8213*  8233   8450*  8451*  8472   8519   8752*  8753*  8774
                        8821   13391
RHDST   004066          2863#  13198*  13974
RHDS1   004072          2865#  13132   13184  13233  13242  13249  13260  13271  13287  13308  13319  13330  13355
                       13942
RHDT    004106          2872#   3286   3301   3330   14046
RHEC1   004124          2880#  14030
```

```
RHEC2   004126          2881#  14038
RHER1   004074          2866#   6921    6943   13950
RHER2   004120          2878#  13958
RHER3   004122          2879#  13966
RHFC    004066          2861#  13302*  13346*
RHLA    004100          2869#  14014
RHMR    004104          2871#  14006
RHOF    004112          2875#  13200*  13998
RHSN    004110          2873#  14054
RHTC    004112          2874#   3300*  13231*  13299*  13300*  13339*  13340*
RHWC    004062          2859#   3958*   3959    3963    3983*   3984    3988    4337    4509    4781    5171    5588    6011
                        6432    6869    7144    7323    7726*   7831*   7935    8167    8403    8705    9007    9056*   9064*
                        9241    9289*   9297*   9474    9522*   9530*   9707    9755*   9763*   9940    9991*   10058   10104*
                       10172   10218*  10285   10331*  10398   10445*  10516   10563*  10631   10678*  10745   10791*  10859
                       10905*  10972   11018*  11085   11131*  11199   11246*  11317   11365*  11432   11478*  11545   11591*
                       11659   11705*  11773   11819*  11886   11933*  12004   12057*  12125   12176*  12243   12294*  12362
                       12413*  12481   12532*  12599   12651*  12722   12911*  13105   13107   13341   13910
RMR    = 000004         2637#
RPCS1   004050          2834#  15790
RPVEC   002716          2542#   3143   13401   13408*  13423   13437*
RPVECT  043056          3144   13431#  13437
RSCS1   004046          2833#  15790
R0     =%000000          1294#   3143*   3144*   3145*   3211*   3214    3231*   3245*   3254*   3255    3261*   3266    7731*
                         7834*   9059*   9068*   9292*   9301*   9525*   9534*   9758*   9767*   9992*  10105*  10219*  10332*
                        10447*  10564*  10679*  10792*  10906*  11019*  11132*  11248*  11366*  11479*  11592*  11706*  11820*
                        11935*  12059*  12178*  12296*  12415*  12534*  12654*  12814*  12817   12970   12972*  12973   12975
                        12976   12977   13020*  13102   13105*  13108   13113*  13127   13163*  13175   13217*  13230   13374*
                        13391*  13393*  13394*  13522   13532*  13536   13552   13553   13566*  13597   13598*  13599   13602*
                        13806   13810*  13811   13814   13834*  13837*  14060   14061*  14062*  14069*  14070*  14071*  14072*
                        14073*  14074   14076   14078   14079   14084   14090   14092*  14093   14107*  14198   14199*  14200
                        14201*  14202*  14203*  14204*  14241   14266*
R1     =%000001          1295#   3212*   3215*   3262*   3267    4846*   4847    4848*   5241*   5243    5244*   5662*   5665
                         5666*   6084*   6086    6088*   6506*   6509    6511*   9983*   9987*  10096*  10100*  10210*  10214*
                        10323*  10327*  10437*  10441*  10555*  10559*  10670*  10674*  10783*  10787*  10897*  10901*  11010*
                        11014*  11123*  11127*  11238*  11242*  11357*  11361*  11470*  11474*  11583*  11587*  11697*  11701*
                        11811*  11815*  11925*  11929*  12049*  12053*  12168*  12172*  12286*  12290*  12405*  12409*  12524*
                        12528*  12643*  12647*  12932   12937   13103   13106*  13108*  13112*  13392*  13395*  13523   13536*
                        13537   13541   13565*  13807   13812*  13820*  13822*  13824*  13827*  13830   13833*  14089   14090*
                        14095   14099   14101   14106*  14242   14265*
R2     =%000002          1296#   3213*   3214*   3271*   3276*   3287*   3288*   3289*   3290    3311*   3312*   3325*   3326*
                         3328*   3329*   3330*   3331*   3332*   3333*   3334*   3335*   3336    3338*   3341*   3343*   3346
                         3348    3350    3352    3354    3356    3358*   3361*   3363*   3366    3368    3370    3372    3374
                         3376*   3379*   3381*   3392    3398    3404    3410    3419    3426    3433    3435*   3441    3443*
                         3449    3451*   3457    3459*   3465    3467*   3473    3475*   3481    3483*   3489    3491*   3497
                         3499*   3511    3517    3583*   3584*   3585*   3594    3605    3607    3613    3616    3623    3630
                         3637    3651    3657    3661    3667    3668    3669    3670    3671    3672    4845*   4849*   5242*
                         5245*   5664*   5667*   6085*   6089*   6508*   6512*   9984*   9986*  10097*  10099*  10211*  10213*
                        10324*  10326*  10438*  10440*  10556*  10558*  10671*  10673*  10784*  10786*  10898*  10900*  11011*
                        11013*  11124*  11126*  11239*  11241*  11358*  11360*  11471*  11473*  11584*  11586*  11698*  11700*
                        11812*  11814*  11926*  11928*  12050*  12052*  12169*  12171*  12287*  12289*  12406*  12408*  12525*
                        12527*  12644*  12646*  12759*  12760*  12761*  12762   13104   13107*  13109*  13111*  13524   13535*
                        13539*  13542   13549*  13550*  13551   13556*  13564*  13808   13813*  13821*  13823*  13825*  13831
                        13832*  14243   14264*
R3     =%000003          1297#   3272*   3274*   3659*   3664*  12926   12931   12944   12971   12983*  12986*  12988*  12991*
                        13003*  13006*  13008*  13011*  13019*  13525   13533*  13534*  13548*  13551*  13560*  13561*  13563*
                        13755   13756*  13757   13760*  13761   13765   13767   13769*  13771*  14146   14155*  14161*  14162*
```

DERHAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    14165*  14170*  14171*  14172   14181*  14244   14263*
R4      =%000004     1298#   3273*   3274    3275*   3660*   3662*  14147   14149*  14150*  14151*  14152   14153*  14167
                    14169*  14177*  14180*  14245   14262*
R5      =%000005     1299#   3661*   3662    3663*   3714*   3715    3734*   3740*   3741    3761*   3793*   3794    3812*
                     3818*   3819    3838*   3872*   3873    3892*   3898*   3899    3919*   3951*   3952    3970*   3976*
                     3977    3996*   4030*   4031    4050*   4056*   4057    4077*   4111*   4112    4131*   4137*   4138
                     4158*  13526   13528*  13530*  13537*  13541*  13556   13562*  14148   14154*  14156*  14158*  14159*
                    14160*  14161   14179*  14246   14261*
R6      =%000006     1300#   1302    3104*   3105*   3106
R7      =%000007     1301#   1303
SC      = 100000     2612#   3708    3716    3742    7738    7980    8216    8521    8823    11390   11503   11617   11731
                    11844   11960
SC1     = 000100     2685#
SC10    = 001000     2688#
SC2     = 000200     2686#
SC20    = 002000     2689#
SC4     = 000400     2687#
SEECOM   004164      2905#
SERCH    004146      2898#
SETBIT   004652      2928#   3709*   3724    3749    3789*   3802    3827    3867*   3882    3907    3947*   3960    3985
                     4025*   4040    4065    4106*   4121    4146
SETSIZ   006512      3217    3261#
SILONM   004664      2937#   4887*   4909*   4931*   4953*   4975*   4997*   5019*   5041*   5284*   5308*   5332*   5356*
                     5380*   5404*   5428*   5452*   5707*   5731*   5755*   5779*   5803*   5827*   5851*   5875*   6128*
                     6152*   6176*   6200*   6224*   6248*   6272*   6296*   6552*   6577*   6602*   6627*   6652*   6677*
                     6702*   6727*  15794
SKI     = 040000     2779#
SLAVE    004660      2935#   3669*  13231
SLVPR   = 002000     2750#   3302
SN      = 000030     2814#
SND1     006114      3152    3158#
SP      =%000006     1302#   3108*   3125*   3133*   3137    3228    3229    3238*   3245    3250*   3254    3339*   3340*
                     3341    3342*   3343    3359*   3360*   3361    3362*   3363    3377*   3378*   3379    3380*   3381
                     3392*   3398*   3410*   3511*   3517*   3536*   3537*   3538    3539    3555*   3556*   3557    3582*
                     3584    3594*   3605*   3613*   3651*   3657*   3701*   3781*   3859*   3939*   4017*   4098*   4183*
                     4372*   4548*   4816*   5210*   5630*   6052*   6473*   6910*   7182*   7357*   7477*   7597*   7718*
                     7822*   7967*   8203*   8440*   8742*   9047*   9280*   9513*   9746*   9975*  10088*  10202*  10315*
                    10429*  10547*  10662*  10775*  10889*  11002*  11115*  11230*  11349*  11462*  11575*  11689*  11803*
                    11917*  12037*  12156*  12274*  12393*  12512*  12631*  12758*  12760   12762*  12768*  12774*  12811*
                    12847*  12875*  12886*  12892*  12898*  12924   12926*  12927*  12929   12931*  12932*  12933*  12934
                    12941*  12942   12970*  12971*  12972   12977*  13019   13020   13052*  13058*  13074*  13075   13085
                    13086*  13102*  13103*  13104*  13111   13112   13113   13177*  13178*  13179   13281*  13282*  13283
                    13341*  13342*  13343*  13344*  13346   13384*  13394   13401*  13408   13417*  13423*  13467*  13470
                    13472   13473   13501   13502   13506*  13522*  13523*  13524*  13525*  13526*  13527*  13528   13531*
                    13544   13546*  13548   13558   13560   13562   13563   13564   13565   13566   13568*  13569*  13597*
                    13598   13599*  13601   13602   13603*  13605   13607   13609   13615   13617*  13619*  13627*  13631
                    13635   13636   13640   13681*  13682*  13683   13687   13694   13696   13699   13702*  13703*  13704
                    13709   13711   13713*  13714   13732*  13733*  13734*  13735*  13736*  13742*  13755*  13760   13771
                    13772*  13773*  13774*  13804*  13805*  13806*  13807*  13808*  13810   13814*  13816   13818   13826*
                    13827   13829   13830*  13832   13833   13834   13836   13865   13878*  13881*  13897*  13898*  13899
                    13910*  13918*  13926*  13934*  13942*  13950*  13958*  13966*  13974*  13982*  13990*  13998*  14006*
                    14014*  14022*  14030*  14038*  14046*  14054*  14060*  14065*  14089*  14095*  14099*  14106   14107
                    14138*  14139   14140   14141*  14146*  14147*  14148*  14154   14179   14180   14181   14182*  14183*
                    14198*  14199   14209*  14210*  14241*  14242*  14243*  14244*  14245*  14246*  14247*  14248   14256*
                    14260   14261   14262   14263   14264   14265   14266   14271*
SPACFD   004202      2912#  13304
```

```
SR0    = 177572          1417#
SR1    = 177574          1418#
SR2    = 177576          1419#
SR3    = 172516          1420#
SS1      007366          3345    3365    3382    3387#
SS2      006526          3264#   3534    3545
SS3      010476          3536#   3546
SS4      010520          3265    3543#   3564
SS5      010536          3296    3549#
SS6      006602          3279#   3559
SS7      006616          3283#   3552
SS8      010544          3292    3306    3550#
STACK  = 001000          1277#   3108    3701    3781    3859    3939    4017    4098    4183    4372    4548    4816    5210
                         5630    6052    6473    6910    7182    7357    7477    7597    7718    7822    7967    8203    8440
                         8742    9047    9280    9513    9746    9975   10088   10202   10315   10429   10547   10662   10775
                        10889   11002   11115   11230   11349   11462   11575   11689   11803   11917   12037   12156   12274
                        12393   12512   12631
START    005536          3098    3101#
STKLMT= 177774           1288#
ST200    005012          2984#   3097*   3099*   3209
SWR      001140          1473#   3106    3127*   3129    3135*   3717    3743    3796    3821    3875    3901    3954    3979
                         4033    4059    4114    4140   13462   13476   13478   13484   13491   13861   13868   13873   13876
                        13897   14247   14260*
SWREG    000176          1393#   3135
SW0    = 000001          1341#
SW00   = 000001          1331#   1341
SW01   = 000002          1330#   1340
SW02   = 000004          1329#   1339
SW03   = 000010          1328#   1338
SW04   = 000020          1327#   1337
SW05   = 000040          1326#   1336
SW06   = 000100          1325#   1335   13899
SW07   = 000200          1324#   1334
SW08   = 000400          1323#   1333
SW09   = 001000          1322#   1332
SW1    = 000002          1340#
SW10   = 002000          1321#
SW11   = 004000          1320#
SW12   = 010000          1319#
SW13   = 020000          1318#
SW14   = 040000          1317#
SW15   = 100000          1316#
SW2    = 000004          1339#
SW3    = 000010          1338#
SW4    = 000020          1337#
SW5    = 000040          1336#
SW6    = 000100          1335#
SW7    = 000200          1334#
SW8    = 000400          1333#
SW9    = 001000          1332#
S1     = 000001          2730#
S2     = 000002          2731#
S4     = 000004          2732#
TBITVE= 000014           1374#
TC     = 000032          2816#
TDF    = 000040          2703#
```

```
TESTAD  040652          13044#
TESTDV  005324           3053#   3266*   3268    3273    3332    3339    3359    3377   15806   15808
TESTVC  005326           3054#   3267*   3315    3331   15806
TFOUND  005322           3052#   3263*   3311    3325    3333    3529*   3536
TIEOUT  037600           3161    3579   12842#
TIMCNT  040446          12967#  12983   12988   13003   13008
TKVEC = 000060           1381#  13668*  13669*
TMCSI   004052           2835#  15790
TMPSLV  005364           3071#   3299*   3300    3304    3307*   3312
TMPO    005250           3031#   3229*   3257*
TMP1    005252           3032#   3230*   3238    3250    3256*
TMP4    005254           3033#
TN    = 000067           3701#   3702    3781#   3782    3859#   3860    3939#   3940    4017#   4018    4098#   4099    4183#
                         4184    4372#   4373    4548#   4549    4816#   4817    5210#   5211    5630#   5631    6052#   6053
                         6473#   6474    6910#   6911    7182#   7183    7357#   7358    7477#   7478    7597#   7598    7718#
                         7719    7822#   7823    7967#   7968    8203#   8204    8440#   8441    8742#   8743    9047#   9048
                         9280#   9281    9513#   9514    9746#   9747    9975#   9976   10088#  10089   10202#  10203   10315#
                        10316   10429#  10430   10547#  10548   10662#  10663   10775#  10776   10889#  10890   11002#  11003
                        11115#  11116   11230#  11231   11349#  11350   11462#  11463   11575#  11576   11689#  11690   11803#
                        11804   11917#  11918   12037#  12038   12156#  12157   12274#  12275   12393#  12394   12512#  12513
                        12631#  12632
TOTALA  005246           3029#
TPVEC = 000064           1382#
TRAPVE= 000034           1380#   3114*   3115*   3142*
TRE   = 040000           2611#   3308    3549    3554    3708    3716    3742    6982    7738    7980    8216    8521    8823
TRK1  = 004000           2690#
TRK10 = 040000           2693#
TRK2  = 010000           2691#
TRK20 = 100000           2694#
TRK4  = 020000           2692#
TRTVEC= 000014           1375#
TSBAE   005140           3006#
TSBASE  005220           3015#
TSCOMD  005200           3012#
TSCS3   005160           3009#
TSDEVI  005034           2988#
TSINDX  005116           3000#
TSRHNO  005014           2986#
TSSLAV  005074           2996#
TSTNM   004656           2934#   3702*   3782*   3860*   3940*   4018*   4099*   4184*   4373*   4549*   4817*   5211*   5631*
                         6053*   6474*   6911*   7183*   7358*   7478*   7598*   7719*   7823*   7968*   8204*   8441*   8743*
                         9048*   9281*   9514*   9747*   9976*  10089*  10203*  10316*  10430*  10548*  10663*  10776*  10890*
                        11003*  11116*  11231*  11350*  11463*  11576*  11690*  11804*  11918*  12038*  12157*  12275*  12394*
                        12513*  12632*  13052   15792   15794   15796   15798   15800   15802   15804
TSTOTL  005114           2999#
TSTUNT  005506           3086#   3538*   3583   12759   12788
TST1    006152           3207#
TST10   013266           4097#
TST11   013570           4182#
TST12   014116           4371#
TST13   014420           4547#
TST14   015102           4815#
TST15   016012           5209#
TST16   016724           5629#
TST17   017636           6051#
TST2    010622           3540    3577#  12787   12823
```

```
TST20   020552          6472#
TST21   021466          6909#
TST22   022100          7181#
TST23   022374          7356#
TST24   022556          7476#
TST25   022740          7596#
TST26   023122          7717#
TST27   023316          7821#
TST3    011420          3700#
TST30   023600          7966#
TST31   024202          8202#
TST32   024604          8439#
TST33   025332          8741#
TST34   026060          9046#
TST35   026456          9279#
TST36   027054          9512#
TST37   027452          9745#
TST4    011722          3780#
TST40   030050          9974#
TST41   030276         10087#
TST42   030524         10201#
TST43   030752         10314#
TST44   031200         10428#
TST45   031434         10546#
TST46   031662         10661#
TST47   032110         10774#
TST5    012202          3858#
TST50   032336         10888#
TST51   032564         11001#
TST52   033012         11114#
TST53   033240         11229#
TST54   033474         11348#
TST55   033722         11461#
TST56   034150         11574#
TST57   034376         11688#
TST6    012504          3938#
TST60   034624         11802#
TST61   035052         11916#
TST62   035306         12036#
TST63   035544         12045   12155#
TST64   036002         12164   12273#
TST65   036240         12282   12392#
TST66   036476         12401   12511#
TST67   036734         12520   12630#
TST7    012764          4016#
TST70   037200         12639   12751#
TSUNIT  005054          2993#
TSVEC   005120          3003#
TUF     = 000100        2704#
TYPDS   = 104405        3239    3251    3393    3411    3512    3595    3614    3652    12763   12775   12812   14100   14225#
TYPE    = 104401        3154    3219    3223    3234    3240    3246    3252    3388    3394    3400    3406    3421    3428
                        3436    3444    3452    3460    3468    3476    3484    3492    3500    3507    3513    3519    3523
                        3586    3590    3597    3601    3609    3618    3625    3632    3639    3647    3653    12754   12764
                       12770   12776   12780   12810   12813   12843   12849   12867   12871   12877   12882   12888   12894
                       13048   13054   13060   13061   13065   13069   13076   13080   13380   13386   13397   13403   13409
                       13413   13419   13425   13432   13567   13610   13685   13693   13763   13766   13770   13838   13840
```

```
                    13863   13871   13904   13906   13914   13922   13930   13939   13946   13954   13962   13970   13978
                    13986   13994   14002   14010   14018   14026   14034   14042   14050   14059   14081   14083   14086
                    14088   14103   14173   14221#  14269
TYPERR  046022      13902   14058#
TYPNT   005330       3064#   3280    3290
TYPOC = 104402       3399    3518    3606    3658   12769   12848   12853   12876   12881   12897   12893   12899   13053
                    13059   13385   13402   13418   13424   13436   13911   13919   13927   13935   13943   13951   13959
                    13967   13975   13983   13991   13999   14007   14015   14023   14031   14039   14047   14055   14067
                    14096   14222#
TYPON = 104404      14224#
TYPOS = 104403      14223#
UNIT    005D10       2983#   3667*   4192    4245    4382    4434    4557    4593    4645    4686    4825    4865    5066
                     5220    5261    5478    5640    5683    5901    6062    6105    6322    6483    6528    6754    7014
                     7044    7243    7758    7859    8023    8075    8259    8311    8498    8542    8613    8800    8844
                     8915    9076    9149    9309    9382    9542    9615    9775    9848   10036   10150   10263   10376
                    10493   10609   10723   10837   10950   11063   11177   11294   11410   11523   11637   11751   11864
                    11981   12103   12221   12340   12459   12577   12699   12909
UNLOAD  004136       2994#
UNS     = 040000     2649#
USEVEC  005004       2978#
US1     = 000001     2556#
US2     = 000002     2557#
US4     = 000004     2558#
UWR     = 000010     2775#
VECPNT  005310       3050#   3254*   3262
VECTOR  005512       3091#   3668*
VUF     = 000002     2774#
VU3D    = 010000     2710#
VV      = 000100     2623#  12927   12942   13182
WAITBT  004672       2940#  12976*  12984   12989   13004   13009   15802
WAITPC  004666       2938#  12973*  12974*  15802
WAITRE  004670       2939#  12975*  12984   12989   12993   13004   13009   13013   13017   15802
WAIT.T  040450      12969#  14231
WAT     = 104411     4214    4404    4580    4616    4853    5248    5670    6092    6515   13154   13208   13248   13259
                    13270   13286   13307   13318   13329   13354   13365   14231#
WATO    004646       2924#  12981   13258*  13268*  13317*  13327*  13459*
WC      = 000002     2802#  13106
WCE     = 040000     2571#   9075    9308    9541    9774
WCEEW   = 004000     2582#   9564    9797
WCEOW   = 010000     2583#   9098    9331
WCF     = 000040     2640#
WCU     = 000001     2698#
WLE     = 004000     2646#
WORKBS  004770       2972#
WORKNM  004772       2973#
WORKVC  004774       2974#
WORUNT  005510       3088#   3539*   3582   12758   12785*  12788*
WRCHDT  004152       2900#
WRCHEK  004150       2899#   9069    9302    9535    9768
WRFROM  004210       2917#   7727    7833    7937    7976*   7977    8212*   8213    8449*   8450    8451    8751*   8752
                     8753    9055*   9057    9288*   9290    9521*   9523    9754*   9756
WRIDAT  004154       2901#   7732    7835    9060    9293    9526    9759    9993   10106   10220   10333   10448   10565
WRIFOR  004156       2902#
WRL     = 004000     2628#
WRTBIT  004650       2927#   3708*   3788*   3866*   3946*   4024*   4105*
WRU     = 000400     2706#
```

```
WSU    = 000004          2700#
$AUTOB  001134           1470#
$BDADR  001122           1465#  3727*  3753*  3805*  3930*  3885*  3911*  3963*  3988*  4043*  4069*  4124*  4150*
                        12937*  12944*  15804
$BDDAT  001126           1467#  3286*  3290   3301*  3302   3722*  3725   3748*  3751   3801*  3803   3826*  3828
                         3880*  3883   3906*  3909   3959*  3961   3984*  3986   4038*  4041   4064*  4067   4119*
                         4122   4145*  4148   4194*  4195   4228*  4229   4247*  4248   4266*  4267   4284*  4285
                         4301*  4302   4319*  4320   4337*  4338   4384*  4385   4418*  4419   4436*  4437   4455*
                         4456   4473*  4474   4491*  4492   4509*  4510   4559*  4560   4595*  4596   4629*  4630
                         4647*  4648   4669*  4670   4688*  4689   4708*  4709   4727*  4728   4745*  4746   4763*
                         4764   4781*  4782   4827*  4828   4867*  4868   4891*  4892   4913*  4914   4935*  4936
                         4957*  4958   4979*  4980   5001*  5002   5023*  5024   5045*  5046   5068*  5069   5090*
                         5091   5111*  5112   5131*  5132   5151*  5152   5171*  5172   5222*  5223   5263*  5264
                         5288*  5299   5312*  5313   5336*  5337   5360*  5361   5384*  5385   5408*  5409   5432*
                         5433   5456*  5457   5480*  5481   5503*  5504   5525*  5526   5546*  5547   5567*  5568
                         5588*  5589   5642*  5643   5685*  5686   5711*  5712   5735*  5736   5759*  5760   5783*
                         5784   5807*  5808   5831*  5832   5855*  5856   5879*  5880   5903*  5904   5926*  5927
                         5948*  5949   5969*  5970   5990*  5991   6011*  6012   6064*  6065   6107*  6108   6132*
                         6133   6156*  6157   6180*  6181   6204*  6205   6228*  6229   6252*  6253   6276*  6277
                         6300*  6301   6324*  6325   6347*  6348   6369*  6370   6390*  6391   6411*  6412   6432*
                         6433   6485*  6486   6530*  6531   6556*  6557   6581*  6582   6606*  6607   6631*  6632
                         6656*  6657   6681*  6682   6706*  6707   6731*  6732   6756*  6757   6780*  6781   6803*
                         6804   6825*  6826   6847*  6848   6869*  6870   6921*  6922   6943*  6944   6964*  6965
                         6989*  6990   7020*  7021   7046*  7047   7072*  7073   7096*  7097   7120*  7121   7144*
                         7145   7203*  7204   7224*  7225   7245*  7246   7266*  7267   7285*  7286   7304*  7305
                         7323*  7324   7372*  7373   7392*  7393   7416*  7417   7439*  7440   7492*  7493   7512*
                         7513   7536*  7537   7559*  7560   7612*  7613   7632*  7633   7656*  7657   7679*  7680
                         7740*  7741   7760*  7761   7780*  7781   7841*  7842   7861*  7862   7881*  7882   7899*
                         7900   7917*  7918   7935*  7936   7982*  7983   8003*  8004   8025*  8026   8053*  8054
                         8077*  8078   8101*  8102   8123*  8124   8145*  8146   8167*  8168   8218*  8219   8239*
                         8240   8261*  8262   8289*  8290   8313*  8314   8337*  8338   8359*  8360   8391*  8392
                         8403*  8404   8456*  8457   8478*  8479   8500*  8501   8523*  8524   8544*  8545   8565*
                         8566   8591*  8592   8615*  8616   8639*  8640   8661*  8662   8683*  8684   9705*  9706
                         8758*  8759   8780*  8781   8802*  8803   8825*  8826   8846*  8847   8867*  8868   8893*
                         8894   8917*  8918   8941*  8942   8963*  8964   8985*  8986   9007*  9008   9078*  9079
                         9100*  9101   9127*  9128   9151*  9152   9175*  9176   9197*  9198   9219*  9220   9241*
                         9242   9311*  9312   9333*  9334   9360*  9361   9384*  9385   9408*  9409   9430*  9431
                         9452*  9453   9474*  9475   9544*  9545   9566*  9567   9593*  9594   9617*  9618   9641*
                         9642   9663*  9664   9685*  9686   9707*  9708   9777*  9778   9799*  9800   9826*  9827
                         9850*  9851   9874*  9875   9896*  9897   9918*  9919   9940*  9941   9999*  10000  10018*
                        10019  10038*  10039  10058*  10059  10112*  10113  10132*  10133  10152*  10153  10172*  10173
                        10226*  10227  10245*  10246  10265*  10266  10285*  10286  10339*  10340  10359*  10359  10378*
                        10379  10398*  10399  10454*  10455  10474*  10475  10495*  10496  10516*  10517  10571*  10572
                        10591*  10592  10611*  10612  10631*  10632  10686*  10687  10705*  10706  10725*  10726  10745*
                        10746  10799*  10800  10819*  10820  10839*  10840  10859*  10860  10913*  10914  10932*  10933
                        10952*  10953  10972*  10973  11026*  11027  11045*  11046  11065*  11066  11085*  11086  11139*
                        11140  11159*  11160  11179*  11180  11199*  11200  11255*  11256  11275*  11276  11296*  11297
                        11317*  11318  11373*  11374  11392*  11393  11412*  11413  11432*  11433  11486*  11487  11505*
                        11506  11525*  11526  11545*  11546  11599*  11600  11619*  11620  11639*  11640  11659*  11660
                        11713*  11714  11733*  11734  11753*  11754  11773*  11774  11827*  11828  11846*  11847  11866*
                        11867  11986*  11887  11942*  11943  11962*  11963  11983*  11984  12004*  12005  12065*  12066
                        12085*  12086  12105*  12106  12125*  12126  12184*  12185  12203*  12204  12223*  12224  12243*
                        12244  12302*  12303  12322*  12323  12342*  12343  12362*  12363  12421*  12422  12441*  12442
                        12461*  12462  12481*  12482  12540*  12541  12559*  12560  12579*  12580  12599*  12600  12660*
                        12661  12680*  12681  12701*  12702  12722*  12723  12993*  13013*  13017*  13132*  13133  13184*
                        13185  13233*  13234*  13235  15789  15792  15794  15796  15798  15800  15804
```

DERHAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
$BELL   001216          1499#  13693  13779  13863  13884
$CHARC  044040         13612*  13622* 13629  13638* 13643#
$CKSWR= ****** U       14228
$CMTAG  001100          1453#   3103   3104   3112   3118   3119
$CM1    = 000006        1485#   1486#  1487#  1488#  1489#  1490#  1491#
$CM2    = 000014        1485#   1486#  1487#  1488#  1489#  1490#  1491#
$CM3    = 000006        1483#   1485
$CM4    = 000006        1491#   1492#  1493#  1494#  1495#  1496#  1497#
$CNTLC  044543         13685   13779#
$CNTLG  044555         13781#
$CNTLU  044550         13780#
$CRLF   001223          1501#  13060  13611  13646  13779  13843  13871  13884  14059  14083  14088
$DBLK   043614         13533   13567  13575#
$DOAGN  037554         12806   12815  12821#
$DTBL   043604         13536   13571#
$ENDAD  037544          1397   12817#
$ENDCT  037512         12808#
$ENDMG  037563         12810   12825#
$ENULL  037560         12813   12824#
$EOP    037456         12798#
$EOPCT  037504         12805#  12809
$ERFLG  001103          1456#  13449  13480  13482  13488* 13509  13858* 13884
$ERMAX  001115          1462#   3120* 13482  13504* 13509
$ERROR  044744          3112   13857#
$ERRPC  001116          1463#  13865* 13866* 13867  13884  14065  15789  15790  15792  15794  15796  15798  15800
                       15802   15804  15806  15808
$ERRTB  001226          1518#  14073
$ERRTY  045102         13870   13897#
$ERTTL  001112          1460#  12774  12784* 13864* 13884
$ESCAP  001214          1498#   3119* 13503* 13879  13881  13884
$FILLC  001156          1481#  13615  13646
$FILLS  001155          1480#  13646
$GDADR  001120          1464#
$GDDAT  001124          1466#   3715*  3716*  3721   3723*  3724*  3725   3741*  3742*  3747   3749*  3750*  3751
                        3794*   3795*  3800   3802*  3803   3819*  3820*  3825   3827*  3828   3873*  3874*  3879
                        3881*   3882*  3883   3899*  3900*  3905   3907*  3908*  3909   3952*  3953*  3958   3960*
                        3961    3977*  3978*  3983   3985*  3986   4031*  4032*  4037   4039*  4040*  4041   4057*
                        4058*   4063   4065*  4066*  4067   4112*  4113*  4118   4120*  4121*  4122   4138*  4139*
                        4144    4146*  4147*  4148   4191*  4192*  4195   4226*  4229   4244*  4245*  4248   4264*
                        4267    4282*  4285   4299*  4302   4317*  4320   4335*  4338   4381*  4382*  4385   4416*
                        4419    4433*  4434*  4437   4453*  4456   4471*  4474   4489*  4492   4507*  4510   4556*
                        4557*   4560   4592*  4593*  4596   4627*  4630   4644*  4645*  4648   4667*  4670   4685*
                        4686*   4689   4706*  4709   4725*  4728   4743*  4746   4761*  4764   4779*  4782   4824*
                        4825*   4828   4864*  4865*  4868   4889*  4892   4911*  4914   4933*  4936   4955*  4958
                        4977*   4980   4999*  5002   5021*  5024   5043*  5046   5065*  5066*  5069   5088*  5091
                        5109*   5112   5129*  5132   5149*  5152   5169*  5172   5219*  5220*  5223   5260*  5261*
                        5264    5286*  5289   5310*  5313   5334*  5337   5358*  5361   5382*  5385   5406*  5409
                        5430*   5433   5454*  5457   5477*  5478*  5481   5501*  5504   5523*  5526   5544*  5547
                        5565*   5568   5586*  5589   5639*  5640*  5643   5682*  5683*  5686   5709*  5712   5733*
                        5736    5757*  5760   5781*  5784   5805*  5808   5829*  5832   5853*  5856   5877*  5880
                        5900*   5901*  5904   5924*  5927   5946*  5949   5967*  5970   5988*  5991   6009*  6012
                        6061*   6062*  6065   6104*  6105*  6108   6130*  6133   6154*  6157   6179*  6181   6202*
                        6205    6226*  6229   6250*  6253   6274*  6277   6298*  6301   6321*  6322*  6325   6345*
                        6348    6367*  6370   6388*  6391   6409*  6412   6430*  6433   6482*  6483*  6486   6527*
                        6528*   6531   6554*  6557   6579*  6582   6604*  6607   6629*  6632   6654*  6657   6679*
                        6682    6704*  6707   6729*  6732   6753*  6754*  6757   6778*  6781   6801*  6804   6823*
```

```
                    6826    6845*   6848    6867*   6870    6919*   6922    6941*   6944    6962*   6965    6987*   6990
                    7018*   7021    7043*   7044*   7047    7070*   7073    7094*   7097    7118*   7121    7142*   7145
                    7201*   7204    7222*   7225    7242*   7243*   7246    7264*   7267    7283*   7286    7302*   7305
                    7321*   7324    7370*   7373    7390*   7393    7414*   7417    7437*   7440    7490*   7493    7510*
                    7513    7534*   7537    7557*   7560    7610*   7613    7630*   7633    7654*   7657    7677*   7680
                    7738*   7741    7757*   7758*   7761    7778*   7781    7839*   7842    7858*   7859*   7862    7879*
                    7982    7997*   7900    7915*   7918    7933*   7936    7980*   7983    8001*   8004    8022*   8023*
                    8026    8051*   8054    8074*   8075*   8078    8099*   8102    8121*   8124    8143*   8146    8165*
                    8168    8216*   8219    8237*   8240    8258*   8259*   8262    8287*   8290    8310*   8311*   8314
                    8335*   8338    8357*   8360    8379*   8382    8401*   8404    8454*   8457    8476*   8479    8497*
                    8498*   8501    8521*   8524    8541*   8542*   8545    8563*   8566    8589*   8592    8612*   8613*
                    8616    8637*   8640    8659*   8662    8681*   8684    8703*   8706    8756*   8759    8778*   8781
                    8799*   8800*   8803    8823*   8826    8843*   8844*   8847    8865*   8868    8891*   8894    8914*
                    8915*   8918    8939*   8942    8961*   8964    8983*   8986    9005*   9008    9075*   9076*   9079
                    9098*   9101    9125*   9128    9148*   9149*   9152    9173*   9176    9195*   9198    9217*   9220
                    9239*   9242    9308*   9309*   9312    9331*   9334    9358*   9361    9381*   9382*   9385    9406*
                    9409    9428*   9431    9450*   9453    9472*   9475    9541*   9542*   9545    9564*   9567    9591*
                    9594    9614*   9615*   9618    9639*   9642    9661*   9664    9683*   9686    9705*   9708    9774*
                    9775*   9778    9797*   9800    9824*   9827    9847*   9848*   9851    9872*   9875    9894*   9897
                    9916*   9919    9938*   9941    9997*   10000   10016*  10019   10035*  10036*  10039   10056*  10059
                   10110*  10113   10130*  10133   10149*  10150*  10153   10170*  10173   10224*  10227   10243*  10246
                   10262*  10263*  10266   10283*  10286   10337*  10340   10356*  10359   10375*  10376*  10379   10396*
                   10399   10452*  10455   10472*  10475   10492*  10493*  10496   10514*  10517   10569*  10572   10589*
                   10592   10608*  10609*  10612   10629*  10632   10684*  10687   10703*  10706   10722*  10723*  10726
                   10743*  10746   10797*  10800   10817*  10820   10836*  10837*  10840   10857*  10860   10911*  10914
                   10930*  10933   10949*  10950*  10953   10970*  10973   11024*  11027   11043*  11046   11062*  11063*
                   11066   11083*  11086   11137*  11140   11157*  11160   11176*  11177*  11180   11197*  11200   11253*
                   11256   11273*  11276   11293*  11294*  11297   11315*  11318   11371*  11374   11390*  11393   11409*
                   11410*  11413   11430*  11433   11484*  11487   11503*  11506   11522*  11523*  11526   11543*  11546
                   11597*  11600   11617*  11620   11636*  11637*  11640   11657*  11660   11711*  11714   11731*  11734
                   11750*  11751*  11754   11771*  11774   11825*  11828   11844*  11847   11863*  11864*  11967   11884*
                   11887   11940*  11943   11960*  11963   11980*  11981*  11984   12002*  12005   12063*  12066   12083*
                   12086   12102*  12103*  12106   12123*  12126   12182*  12185   12201*  12204   12220*  12221*  12224
                   12241*  12244   12300*  12303   12320*  12323   12339*  12340*  12343   12360*  12363   12419*  12422
                   12439*  12442   12458*  12459*  12462   12479*  12482   12538*  12541   12557*  12560   12576*  12577*
                   12580   12597*  12600   12658*  12661   12678*  12681   12698*  12699*  12702   12720*  12723   13130*
                   13133   13182*  13185   13232*  13235   15792   15794   15796   15798   15800   15804

$GET42  037534          12814#
$GTSWR= ****** U        14227
$HD   = 000000           1260
$HIOCT  044742          13831*  13842#
$ICNT   001104           1457#  13495*  13496   13498*  13508
$ILLUP  046662          14239   14255   14274#
$INTAG  001135           1471#
$ITEMB  001114           1461#  13867*  13884   14062
$LF     001224           1502#  13646   13770   13779   13843   13884
$LPADR  001106           1458#   3121*  13075*  13086   13486*  13501*  13506   13508
$LPERR  001110           1459#   3122*   3713*   3739*   3792*   3817*   3871*   3897*   3950*   3975*   4029*   4055*   4110*
                         4136*  13058   13085*  13486   13502*  13508   13878
$MAIL = ****** U         3139   13501   13599   13873
$MNEW   044573          13784#
$MSWR   044562          13782#
$MXCNT  043376          13499   13508#
$NULL   001154           1479#  13617   13646
$NWTST= 000001           3182#   3184    3572#   3574    3684#   3686    3765#   3767    3842#   3844    3923#   3925    4000#
                         4002    4081#   4083    4166#   4168    4355#   4357    4526#   4528    4799#   4800    5190#   5192
```

```
                        5609#   5611    6031#   6033    6452#   6454    6890#   6892    7167#   7169    7341#   7343    7461#
                        7463    7581#   7583    7701#   7703    7797#   7799    7953#   7955    8189#   8191    8425#   8427
                        8727#   8729    9030#   9032    9263#   9265    9496#   9498    9729#   9731    9963#   9965   10076#
                       10078   10190#  10192   10303#  10305   10416#  10418   10535#  10537   10650#  10652   10763#  10765
                       10877#  10879   10990#  10992   11103#  11105   11217#  11219   11337#  11339   11450#  11452   11563#
                       11565   11677#  11679   11791#  11793   11904#  11906   12024#  12026   12143#  12145   12261#  12263
                       12380#  12382   12499#  12501   12617#  12619   12744#  12746
$OCNT   046432         14145*  14174*  14187#
$OMODE  046434         14140*  14144*  14149   14152*  14163*  14189#
$OVER   043362         13463   13479   13487   13497   13505#
$PASS   001100          1454#  12802*  12803*  12811   12824   13493   13509
$POWER  046670         14270   14277#
$PWRAD  046656         14272#
$PWRDN  046516          3116   14239#  14267
$PWRMG  046652         14270#
$PWRUP  046570         14249   14255#
$QLES   001222          1500#  13646   13763   13779   13840   13843   13884
$RDCHR  044334         13732#  14228
$RDDEC= ****** U       14231
$RDLIN  044424         13755#  14229
$RDOCT  044604         13804#  14230
$RDSZ =  000011        13748#
$REGAD  001160          1483#
$REG0   001162          1485#
$REG1   001164          1486#
$REG2   001166          1487#
$REG3   001170          1488#
$REG4   001172          1489#
$REG5   001174          1490#
$RTNAD  037556         12823#
$R2A  = ****** U       14231
$SAVRE= ****** U       14231
$SAVR6  046666         14248*  14256   14257*  14258*  14276#
$SCOPE  043120          3110   12854   12900   13458#
$SETUP= 000017          3102#   3109    3110    3112    3114    3116    3118    3119    3121   12800   13459   13689   13690
                       13721   13786   13858   13876   13883
$SS1  = 000000          3139#
$STUP = 177777          3102#
$SVLAD  043334         13471   13500#
$SWR  = 167700          1251#   1260    1265    1266    1267    1268    1269    1270    1271    1497    1498    1499    3118
                        3119    3121    3122    3208    3578    3701    3781    3859    3939    4017    4098    4183    4372
                        4548    4816    5210    5630    6052    6473    6910    7182    7357    7477    7597    7718    7822
                        7967    8203    8440    8742    9047    9280    9513    9746    9975   10088   10202   10315   10429
                       10547   10662   10775   10889   11002   11115   11230   11349   11462   11575   11689   11803   11917
                       12037   12156   12274   12393   12512   12631   12752   12795   12801   12816   12822   12824   13450
                       13451   13452   13453   13454   13462   13474   13476   13477   13480   13481   13482   13489   13490
                       13491   13502   13505   13508   13849   13850   13851   13852   13853   13861   13868   13873   13876
                       13884   14273
$SWRMK= 000000          1271    1272   13454   13455   13478
$TIMES  001212          1497#   3118*   3208*   3578*  12752*  12801*  13489*  13496  13499*  13508
$TKB    001146          1476#  13649   13670   13681   13702
$TKCNT  044044         13650#  13665*  13691   13708*  13739   13741*
$TKINT  044064          3147   13665#  13686
$TKQEN= 044063         13654#  13716   13744
$TKQIN  044046         13651#  13666*  13667   13714*  13715*  13716   13718*
$TKQOU  044050         13652#  13667*  13742   13743*  13744   13746*
```

DERHAB.SRC CROSS REFERENCE TABLE -- USER SYMBOLS

```
$TKQSR  044052          13653#  13666   13718   13746
$TKS    001144           1475#  13649   13671*  13698*  13700   13706*
$TKSRV  044134          13668   13681#
$TMP0   001176           1491#
$TMP1   001200           1492#   7997*   8233*   8472*   8774*
$TMP2   001202           1493#   8519*   8821*
$TMP3   001204           1494#
$TMP4   001206           1495#
$TMP5   001210           1496#
$TN   = 000071           1251#   1260    3182    3208#   3540    3572    3578#   3684    3701#   3765    3781#   3842    3859#
                         3923    3939#   4000    4017#   4081    4098#   4166    4183#   4355    4372#   4526    4548#   4798
                         4816#   5190    5210#   5609    5630#   6031    6052#   6452    6473#   6890    6910#   7167    7182#
                         7341    7357#   7461    7477#   7581    7597#   7701    7718#   7797    7822#   7953    7967#   8189
                         8203#   8425    8440#   8727    8742#   9030    9047#   9263    9280#   9496    9513#   9729    9746#
                         9963    9975#   10076   10088#  10190   10202#  10303   10315#  10416   10429#  10535   10547#  10650
                        10662#  10763   10775#  10877   10889#  10990   11002#  11103   11115#  11217   11230#  11337   11349#
                        11450   11462#  11563   11575#  11677   11689#  11791   11803#  11904   11917#  12024   12037#  12044
                        12143   12156#  12163   12261   12274#  12281   12380   12393#  12400   12499   12512#  12519   12617
                        12631#  12638   12744   12752#
$TPB    001152           1478#  13635*  13646
$TPFLG  001157           1482#  13593   13646
$TPS    001150           1477#  13633   13646
$TRAP   046436           3114   14198#
$TRAP2  046460          14209#  14220
$TRP  = 000012          14213#  14222#  14223#  14224#  14225#  14226#  14228   14229#  14230#  14231#  14232#
$TRPAD  046472          14203   14220#
$TSTNM  001102           1455#  12800*  13449   13478   13500*  13505   13509   13860   13884
$TTYIN  044532          13756   13757   13774   13778#
$TYPBN= ****** U        14226
$TYPDS  043400          13521#  14225
$TYPE   043624          13593#  14213   14221
$TYPEC  043774          13614   13621   13628   13633#  13634
$TYPEX  044042          13639   13641   13644#
$TYPOC  046234          14143#  14222
$TYPON  046250          14142   14145#  14224
$TYPOS  046210          14138#  14223
$XTSTR  043140          13465#
$$GET4= 000000          12816#
$OFILL  046433          14139*  14143*  14153   14188#
$40CAT= ****** U        13462   13870
.     = 067014           1387#   1391#   1396#   1398#   1400#   1445#   1452#   1503    2826#   2830#   2917#   2918#   2963#
                         2967#   2986#   2988#   2993#   2996#   3003#   3006#   3009#   3012#   3015#   3049#   3050#   3074#
                         3076#   3077#   3078#   3079#   3080#   3081#   3082#   3083#   3084#   3107    3121    3122    3157#
                         3237#   3249#   3391#   3397#   3403#   3409#   3424#   3471#   3510#   3600#   3604#   3612#   3621#
                         3650#  12824   12828   12846#  12852#  12874#  12880#  12891#  12897#  13064#  13079#  13389#  13400#
                        13406#  13416#  13508   13509   13575#  13646   13649   13653#  13654   13655#  13778#  13779   13786
                        13843   13884   13925#  13933#  13941#  13949#  13957#  13965#  13973#  14029#  14037#  14110#  14251
                        14275   15787#
```

```
ALLREG   1261#  13905  13913  13921  13929  13937  13945  13953  13961  13969  13977  13985  13993  14001  14009
        14017   14025  14033  14041  14049
BITDBL   1261#   9961  10074  10188  10301  10414  10533  10648  10761  10875  10988  11101  11215  11335  11448
        11561   11675  11789  11902  12022  12141  12259  12378  12497  12615
BTBANG   1261#   3683   3764   3841   3922   3999   4080
CHECKD   1261#
CHECKV   1261#
COMMEN      1#   1384#
COMPAR   1261#   4190   4225   4243   4263   4281   4298   4316   4334   4380   4415   4432   4452   4470   4488
         4506   4555   4591   4626   4643   4666   4684   4705   4724   4742   4760   4778   4823   4863   4888
         4910   4932   4954   4976   4998   5020   5042   5064   5087   5108   5128   5148   5168   5218   5259
         5285   5309   5333   5357   5381   5405   5429   5453   5476   5500   5522   5543   5564   5585   5638
         5681   5708   5732   5756   5780   5804   5828   5852   5876   5899   5923   5945   5966   5987   6008
         6060   6103   6129   6153   6177   6201   6225   6249   6273   6297   6320   6344   6366   6387   6408
         6429   6481   6526   6553   6578   6603   6628   6653   6678   6703   6728   6752   6777   6800   6822
         6844   6866   6918   6940   6961   6986   7017   7042   7069   7093   7117   7141   7200   7221   7241
         7263   7282   7301   7320   7369   7389   7413   7436   7489   7509   7533   7556   7609   7629   7653
         7676   7737   7756   7777   7838   7857   7878   7896   7914   7932   7979   8000   8021   8050   8073
         8098   8120   8142   8164   8215   8236   8257   8286   8309   8334   8356   8378   8400   8453   8475
         8496   8520   8540   8562   8588   8611   8636   8658   8680   8702   8755   8777   8798   8822   8842
         8864   8890   8913   8938   8960   8982   9004   9074   9097   9124   9147   9172   9194   9216   9238
         9307   9330   9357   9380   9405   9427   9449   9471   9540   9563   9590   9613   9638   9660   9682
         9704   9773   9796   9823   9846   9871   9893   9915   9937   9996  10015  10034  10055  10109  10129
        10148  10169  10223  10242  10261  10282  10336  10355  10374  10395  10451  10471  10491  10513  10568
        10588  10607  10628  10683  10702  10721  10742  10796  10816  10835  10856  10910  10929  10948  10969
        11023  11042  11061  11082  11136  11156  11175  11196  11252  11272  11292  11314  11370  11389  11408
        11429  11483  11502  11521  11542  11596  11616  11635  11656  11710  11730  11749  11770  11824  11843
        11862  11883  11939  11959  11979  12001  12062  12082  12101  12122  12181  12200  12219  12240  12299
        12319  12338  12359  12418  12438  12457  12478  12537  12556  12575  12596  12657  12677  12697  12719
        13129  13181
ENDCOM      1#   1384#
ERROR    1278#   3317   3561   3728   3754   3806   3831   3886   3912   3964   3989   4044   4070   4125   4151
         4199   4233   4252   4271   4289   4306   4324   4342   4389   4423   4441   4460   4478   4496   4514
         4564   4600   4634   4652   4674   4693   4713   4732   4750   4768   4786   4832   4872   4896   4918
         4940   4962   4984   5006   5028   5050   5073   5095   5116   5136   5156   5176   5227   5268   5293
         5317   5341   5365   5389   5413   5437   5461   5485   5508   5530   5551   5572   5593   5647   5690
         5716   5740   5764   5788   5812   5836   5860   5884   5908   5931   5953   5974   5995   6016   6069
         6112   6137   6161   6185   6209   6233   6257   6281   6305   6329   6352   6374   6395   6416   6437
         6490   6535   6561   6586   6611   6636   6661   6686   6711   6736   6761   6785   6808   6830   6852
         6874   6926   6948   6969   6994   7025   7051   7077   7101   7125   7149   7208   7229   7250   7271
         7290   7309   7328   7377   7397   7421   7444   7497   7517   7541   7564   7617   7637   7661   7684
         7745   7765   7785   7846   7866   7886   7904   7922   7940   7987   8008   8030   8058   8082   8106
         8128   8150   8172   8223   8244   8266   8294   8318   8342   8364   8386   8408   8461   8483   8505
         8528   8549   8570   8596   8620   8644   8666   8688   8710   8763   8785   8807   8830   8851   8872
         8898   8922   8946   8968   8990   9012   9083   9105   9132   9156   9180   9202   9224   9246   9316
         9338   9365   9389   9413   9435   9457   9479   9549   9571   9598   9622   9646   9668   9690   9712
         9782   9804   9831   9855   9879   9901   9923   9945  10004  10023  10043  10063  10117  10137  10157
        10177  10231  10250  10270  10290  10344  10363  10383  10403  10459  10479  10500  10521  10576  10596
        10616  10636  10691  10710  10730  10750  10804  10824  10844  10864  10918  10937  10957  10977  11031
        11050  11070  11090  11144  11164  11184  11204  11260  11280  11301  11322  11378  11397  11417  11437
        11491  11510  11530  11550  11604  11624  11644  11664  11718  11738  11758  11778  11832  11851  11871
        11891  11947  11967  11988  12009  12070  12090  12110  12130  12189  12208  12228  12248  12307  12327
        12347  12367  12426  12446  12466  12486  12545  12564  12584  12604  12665  12685  12706  12727  12938
        12945  12994  13014  13137  13189  13238
ESCAPE      1#   1384#
GETPRI      1#   1384#
```

GETSWR 1# 1384#
MSG 3181# 3184 3572# 3574 3684# 3686 3765# 3767 3842# 3844 3923# 3925 4000# 4002 4081#
4083 4165# 4168 4190# 4200 4224# 4234 4253 4272 4290 4307 4325 4343 4354# 4357
4380# 4390 4414# 4424 4442 4461 4479 4497 4515 4525# 4528 4555# 4565 4591# 4601
4626# 4635 4653 4664# 4675 4694 4714 4733 4751 4769 4787 4797# 4800 4823# 4833
4863# 4873 4886# 4897 4919 4941 4963 4985 5007 5029 5051 5074 5096 5117 5137
5157 5177 5189# 5192 5217# 5228 5258# 5269 5283# 5294 5318 5342 5366 5390 5414
5438 5462 5486 5509 5531 5552 5573 5594 5608# 5611 5637# 5648 5680# 5691 5706#
5717 5741 5765 5789 5813 5837 5861 5885 5909 5932 5954 5975 5996 6017 6030#
6033 6059# 6070 6102# 6113 6127# 6138 6162 6186 6210 6234 6258 6282 6306 6330
6353 6375 6396 6417 6438 6451# 6454 6480# 6491 6525# 6536 6551# 6562 6587 6612
6637 6662 6687 6712 6737 6762 6786 6809 6831 6853 6875 6889# 6892 6917# 6927
6939# 6949 6970 6984# 6995 7026 7052 7078 7102 7126 7150 7166# 7169 7199# 7209
7230 7251 7272 7291 7310 7329 7340# 7343 7367# 7378 7398 7422 7445 7460# 7463
7487# 7498 7518 7542 7565 7580# 7583 7607# 7618 7638 7662 7685 7700# 7703 7736#
7746 7766 7786 7796# 7799 7837# 7847 7867 7887 7905 7923 7941 7952# 7955 7978#
7988 7999# 8009 8031 8059 8083 8107 8129 8151 8173 8188# 8191 8214# 8224 8235#
8245 8267 8295 8319 8343 8365 8387 8409 8424# 8427 8452# 8462 8474# 8484 8506
8529 8550 8571 8597 8621 8645 8667 8689 8711 8726# 8729 8754# 8764 8776# 8786
8808 8831 8852 8873 8899 8923 8947 8969 8991 9013 9029# 9032 9071# 9084 9106
9133 9157 9181 9203 9225 9247 9262# 9265 9304# 9317 9339 9366 9390 9414 9436
9458 9480 9495# 9498 9537# 9550 9572 9599 9623 9647 9669 9691 9713 9728# 9731
9770# 9783 9805 9832 9856 9880 9902 9924 9946 9962# 9965 9995# 10005 10024 10044
10064 10075# 10078 10108# 10118 10138 10158 10178 10189# 10192 10222# 10232 10251 10271 10291
10302# 10305 10335# 10345 10364 10384 10404 10415# 10418 10450# 10460 10480 10501 10522 10534#
10537 10567# 10577 10597 10617 10637 10649# 10652 10682# 10692 10711 10731 10751 10762# 10765
10795# 10805 10825 10845 10865 10876# 10879 10909# 10919 10938 10958 10978 10989# 10992 11022#
11032 11051 11071 11091 11102# 11105 11135# 11145 11165 11185 11205 11216# 11219 11251# 11261
11281 11302 11323 11336# 11339 11369# 11379 11398 11418 11438 11449# 11452 11482# 11492 11511
11531 11551 11562# 11565 11595# 11605 11625 11645 11665 11676# 11679 11709# 11719 11739 11759
11779 11790# 11793 11823# 11833 11852 11872 11892 11903# 11906 11938# 11948 11968 11989 12010
12023# 12026 12061# 12071 12091 12111 12131 12142# 12145 12180# 12190 12209 12229 12249 12260#
12263 12298# 12308 12328 12348 12368 12379# 12382 12417# 12427 12447 12467 12487 12498# 12501
12536# 12546 12565 12585 12605 12616# 12619 12656# 12666 12686 12707 12728 12743# 12746 13128#
13138 13176# 13190

MSG1 4885# 4903 4909# 4925 4931# 4947 4953# 4969 4975# 4991 4997# 5013 5019# 5035 5041#
5057 5063# 5080 5102 5123 5143 5163 5183 5282# 5301 5307# 5325 5331# 5349 5355#
5373 5379# 5397 5403# 5421 5427# 5445 5451# 5469 5475# 5493 5516 5538 5559 5580
5601 5705# 5724 5730# 5748 5754# 5772 5778# 5796 5802# 5820 5826# 5844 5850# 5868
5874# 5892 5898# 5916 5939 5961 5982 6003 6024 6126# 6145 6151# 6169 6175# 6193
6199# 6217 6223# 6241 6247# 6265 6271# 6289 6295# 6313 6319# 6337 6360 6382 6403
6424 6445 6550# 6570 6576# 6595 6601# 6620 6626# 6645 6651# 6670 6676# 6695 6701#
6720 6726# 6745 6751# 6770 6794 6817 6839 6861 6883 6985# 7005 7016# 7036 7062
7088 7112 7136 7160 7368# 7382 7388# 7402 7412# 7426 7435# 7449 7488# 7502 7508#
7522 7532# 7546 7555# 7569 7608# 7622 7628# 7642 7652# 7666 7675# 7689

MSG2 7979# 7992 8000# 8015 8037 8049# 8065 8089 8113 8135 8157 8179 8215# 8228 8236#
8251 8273 8285# 8301 8325 8349 8371 8393 8415 8453# 8466 8475# 8490 8512 8535
8556 8577 8587# 8603 8627 8651 8673 8695 8717 8755# 8768 8777# 8792 8814 8837
8858 8879 8889# 8905 8929 8953 8975 8997 9019 9072# 9090 9112 9123# 9139 9163
9187 9209 9231 9253 9305# 9323 9345 9356# 9372 9396 9420 9442 9464 9486 9538#
9556 9578 9589# 9605 9629 9653 9675 9697 9719 9771# 9789 9811 9822# 9838 9862
9886 9908 9930 9952 9995# 10010 10029 10049 10069 10108# 10123 10143 10163 10183 10222#
10237 10256 10276 10296 10335# 10350 10369 10389 10409 10450# 10466 10486 10507 10528 10567#
10582 10602 10622 10642 10682# 10697 10716 10736 10756 10795# 10810 10830 10850 10870 10909#
10924 10943 10963 10983 11022# 11037 11056 11076 11096 11135# 11150 11170 11190 11210 11251#
11267 11287 11308 11329 11369# 11384 11403 11423 11443 11482# 11497 11516 11536 11556 11595#

```
        11610   11630   11650   11670   11709#  11724   11744   11764   11784   11823#  11838   11857   11877   11897   11938#
        11954   11974   11995   12016   12061#  12076   12096   12116   12136   12180#  12195   12214   12234   12254   12298#
        12313   12333   12353   12373   12417#  12432   12452   12472   12492   12536#  12551   12570   12590   12610   12656#
        12672   12692   12713   12734
MULT        1#   1384#
NEWTST      1#   1384#   3182    3572    3684    3765    3842    3923    4000    4081    4166    4355    4526    4798    5190
         5609    6031    6452    6890    7167    7341    7461    7581    7701    7797    7953    8189    8425    8727    9030
         9263    9496    9729    9963   10076   10190   10303   10416   10535   10650   10763   10877   10990   11103   11217
        11337   11450   11563   11677   11791   11904   12024   12143   12261   12380   12499   12617   12744
PEDPE    1261#   7951    8187    8423    8725
POP         1#   1384#  13018   13111   13562   13832   14260   14261
PUSH        1#   1384#  12969   13101   13521   13806   14241   14247
REPORT      1#   1384#
RHCLEA   1261#   3703    3783    3861    3941    4019    4100    4185    4374    4550    4818    5212    5632    6054    6475
         6912    7184    7359    7479    7599    7720    7824    7969    8044    8205    8280    8442    8582    8744    8884
         9049    9118    9282    9351    9515    9584    9748    9817    9977   10090   10204   10317   10431   10549   10664
        10777   10891   11004   11117   11232   11351   11464   11577   11691   11805   11919   12039   12158   12276   12395
        12514   12633
SAVTST   1261#
SCOPE    1279#   3207    3577    3700    3780    3858    3938    4016    4097    4182    4371    4547    4815    5209    5629
         6051    6472    6909    7181    7356    7476    7596    7717    7821    7966    8202    8439    8741    9046    9279
         9512    9745    9974   10087   10201   10314   10428   10546   10661   10774   10888   11101   11114   11229   11348
        11461   11574   11688   11802   11916   12036   12155   12273   12392   12511   12630   12751   12799
SETPRI      1#   1384#  13735
SETTRA  14213#  14222   14223   14224   14225   14228   14229   14230   14231
SETUP       1#   1384#   3102
SIZDEV   1261#
SKIP        1#   1261#   1384#  12044   12163   12281   12400   12519   12638
SLASH       1#   1384#
SMORE    1261#  13459
SPACE    1384#
STARS       1#   1384#   1444    1448    1503    2537    2543    2595    2597    2796    3182    3206    3572    3576    3684
         3699    3765    3779    3842    3857    3923    3937    4000    4015    4081    4096    4166    4181    4355    4370
         4526    4546    4798    4814    5190    5208    5609    5628    6031    6050    6452    6471    6890    6908    7167
         7180    7341    7355    7461    7475    7581    7595    7701    7716    7797    7820    7953    7965    8189    8201
         8425    8438    8727    8740    9030    9045    9263    9278    9496    9511    9729    9744    9963    9973   10076
        10086   10190   10200   10303   10313   10416   10427   10535   10545   10650   10660   10763   10773   10877   10887
        10990   11000   11103   11113   11217   11228   11337   11347   11450   11460   11563   11573   11677   11687   11791
        11801   11904   11915   12024   12035   12143   12154   12261   12272   12380   12391   12499   12510   12617   12629
        12744   12750   12792   12838   12840   12863   12865   13446   13511   13578   13648   13724   13748   13792   13845
        14115   14192   14237   14253
STARTT   1261#   3701    3781    3859    3939    4017    4098    4183    4372    4548    4816    5210    5630    6052    6473
         6910    7182    7357    7477    7597    7718    7822    7967    8203    8440    8742    9047    9280    9513    9746
         9975   10088   10202   10315   10429   10547   10662   10775   10889   11002   11115   11230   11349   11462   11575
        11689   11803   11917   12037   12156   12274   12393   12512   12631
SWRSU       1#   1384#   3123#
TJUMP    1261#   3540
TRMTRP  14213#
TTSTNO   1261#   3702    3782    3860    3940    4018    4099    4184    4373    4549    4817    5211    5631    6053    6474
         6911    7183    7358    7478    7598    7719    7823    7968    8204    8441    8743    9048    9281    9514    9747
         9976   10089   10203   10316   10430   10548   10663   10776   10890   11003   11116   11231   11350   11463   11576
        11690   11804   11918   12038   12157   12275   12394   12513   12632
TYPBIN      1#   1384#
TYPDEC      1#   1384#  12811
TYPNAM      1#   1384#
TYPNUM      1#   1384#
```

DERHAB.SRC CROSS REFERENCE TABLE -- MACRO NAMES

```
TYPOCS      1#   1384#
TYPOCT      1#   1384#
TYPTXT      1#   1384#   3153   3218   3223   3233   3240   3246   3387   3394   3400   3406   3421   3428   3436
          3444   3452   3460   3468   3476   3484   3492   3500   3506   3513   3519   3523   3586   3590   3597
          3601   3609   3618   3625   3632   3639   3646   3653  12754  12764  12770  12776  12780  12842  12849
         12866  12871  12877  12882  12888  12894  13048  13054  13061  13065  13069  13076  13080  13379  13396
         13397  13403  13409  13413  13419  13425  13431  13906  13914  13922  13930  13938  13946  13954  13962
         13970  13978  13986  13994  14002  14010  14018  14026  14034  14042  14050
WCEOW     1261#  9028   9261   9494   9727
WT        1261#  4213   4403   4579   4615   4852   5247   5669   6091   6514  13153  13207  13247  13257  13269
         13285  13306  13316  13328  13353  13364
$$CMRE    1446#  1485   1486   1487   1488   1489   1490
$$CMTM    1446#  1491   1492   1493   1494   1495   1496
$$ESCA      1#   1384#
$$NEWT      1#   1384#   3182   3572   3684   3765   3842   3923   4000   4081   4166   4355   4526   4798   5190
          5609   6031   6452   6890   7167   7341   7461   7581   7701   7797   7953   8189   8425   8727   9030
          9263   9496   9729   9963  10076  10190  10303  10416  10535  10650  10763  10877  10990  11103  11217
         11337  11450  11563  11677  11791  11904  12024  12143  12261  12380  12499  12617  12744
$$SET    14213#  14222  14223  14224  14225  14228  14229  14230  14231
$$SKIP      1#   1384#
.EQUAT      1#   1251#   1274
.HEADE      1#   1251#
.KT11       1#   1251#   1409
.SETUP      1#   1251#   3101
.SWRHI      1#   1251#   1261
.SWRLO    1251#  1272#
.$ACT1      1#
.$APTB      1#
.$APTH      1#
.$APTY      1#
.$ASTA      1#
.$CATC      1#   1251#   1385
.$CMTA      1#   1251#   1446
.$DB2D      1#
.$DB20      1#
.$DIV       1#
.$EOP       1#   1251#  12790
.$ERRO      1#   1251#  13843
.$ERRT      1#   1251#
.$MULT      1#
.$POWE      1#   1251#  14235
.$RAND      1#
.$RDDE      1#
.$RDOC      1#   1251#  13790
.$READ      1#   1251#  13646
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB20      1#
.$SCOP      1#   1251#  13444
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#   1251#  14190
.$TYPB      1#
.$TYPD      1#   1251#  13509
.$TYPE      1#   1251#  13576
```

```
.$TYPO      1#   1251#  14113
.$4DCA      1#
.1170       1#
```

```
ADD      3275   3340   3342   3360   3362   3378   3380   3529   3663  13074  13343  13541  13603  13827  14073
        14141  14151
ASL      3289   3585   3734   3812   3892   3970   4050   4131   5244   5666  12761  13820  13822  13824  14070
        14071  14072  14202
ASLB    13546
ASR      3334   3537
BCC      3735   3813   3893   3971   4051   4132  13547
BCS      3762   3839   3920   3997   4078   4159
BEQ      3210   3285   3291   3294   3347   3349   3351   3353   3355   3367   3369   3371   3373   3533   3544
         3551   3558   3718   3726   3744   3752   3797   3804   3822   3829   3876   3884   3902   3910   3955
         3962   3980   3987   4034   4042   4060   4068   4115   4123   4141   4149   4198   4232   4251   4270
         4288   4305   4323   4341   4388   4422   4440   4459   4477   4495   4513   4563   4599   4633   4651
         4673   4692   4712   4731   4749   4767   4785   4831   4871   4895   4917   4939   4961   4983   5005
         5027   5049   5072   5094   5115   5135   5155   5175   5226   5267   5292   5316   5340   5364   5388
         5412   5436   5460   5484   5507   5529   5550   5571   5592   5646   5689   5715   5739   5763   5787
         5811   5835   5859   5883   5907   5930   5952   5973   5994   6015   6068   6111   6136   6160   6184
         6208   6232   6256   6280   6304   6328   6351   6373   6394   6415   6436   6489   6534   6560   6585
         6610   6635   6660   6685   6710   6735   6760   6784   6807   6829   6851   6873   6925   6947   6968
         6993   7024   7050   7076   7100   7124   7148   7207   7228   7249   7270   7289   7308   7327   7376
         7396   7420   7443   7496   7516   7540   7563   7616   7636   7660   7683   7744   7764   7784   7845
         7865   7885   7903   7921   7939   7986   8007   8029   8057   8081   8105   8127   8149   8171   8222
         8243   8265   8293   8317   8341   8363   8385   8407   8460   8482   8504   8527   8548   8569   8595
         8619   8643   8665   8687   8709   8762   8784   8806   8829   8850   8871   8897   8921   8945   8967
         8989   9011   9082   9104   9131   9155   9179   9201   9223   9245   9315   9337   9364   9388   9412
         9434   9456   9478   9548   9570   9597   9621   9645   9667   9689   9711   9781   9803   9830   9854
         9878   9900   9922   9944  10003  10022  10042  10062  10116  10136  10156  10176  10230  10249  10269
        10289  10343  10362  10382  10402  10458  10478  10499  10520  10575  10595  10615  10635  10690  10709
        10729  10749  10803  10823  10843  10863  10917  10936  10956  10976  11030  11049  11069  11089  11143
        11163  11183  11203  11259  11279  11300  11321  11377  11396  11416  11436  11490  11509  11529  11549
        11603  11623  11643  11663  11717  11737  11757  11777  11831  11850  11870  11890  11946  11966  11987
        12008  12069  12089  12109  12129  12188  12207  12227  12247  12306  12326  12346  12366  12425  12445
        12465  12485  12544  12563  12583  12603  12664  12684  12705  12726  12786  12815  12936  12943  13005
        13010  13136  13188  13237  13477  13479  13481  13485  13494  13606  13641  13740  13815  13859  13862
        13877  13880  14080  14085  14091  14102  14168
BGE     13497
BGT     12806  13555  13712  13817  14175
BHI     13483
BIC      3329   3556   3716   3723   3742   3750   3795   3820   3874   3881   3900   3908   3953   3978   4032
         4039   4058   4066   4113   4120   4139   4147  10014  10128  10241  10354  10470  10587  10701  10815
        10928  11041  11155  11271  11388  11501  11615  11729  11842  11958  12081  12199  12318  12437  12555
        12676  12803  12910  12933  12941  13234  13300  13340  13393  13682  13703  13713  13826  13898  14165
BIS      3719   3724   3749   3798   3802   3827   3877   3882   3907   3956   3960   3985   4035   4040   4065
         4116   4121   4146   4192   4245   4382   4434   4557   4593   4645   4686   4825   4865   5066   5220
         5261   5478   5640   5683   5901   6062   6105   6322   6483   6528   6754   6937   7044   7197   7243
         7733   7734   7758   7832   7859   7975   8023   8075   8211   8259   8311   8448   8498   8542   8613
         8750   8800   8844   8915   9076   9149   9309   9382   9542   9615   9775   9348  10036  10150  10263
        10376  10446  10493  10609  10723  10837  10950  11063  11177  11247  11234  11410  11523  11637  11751
        11864  11934  11981  12103  12221  12340  12459  12577  12652  12699  12927  13151  13178  13205  13246
        13282  13299  13305  13339  13351  13394  13549  13550  14170  14171
BISB    14062
BIT      3293   3302   3717   3743   3796   3821   3875   3901   3954   3979   4033   4059   4114   4140  12984
        12989  13004  13009  13242  13462  13476  13484  13491  13861  13968  13876
BITB    13629
BLOS    13758
BLT     13538  13554  13620  13710  13819  14176
BMI     13545
```

DERHAB.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
BNE     3107  3130  3150  3216  3258  3277  3303  3305  3337  3357  3375  3405  3420  3427  3434
        3442  3450  3458  3466  3474  3482  3490  3498  3608  3617  3624  3631  3638  3665  4850
        5246  5668  6090  6513  9988 10101 10215 10328 10442 10560 10675 10788 10902 11015 11128
       11243 11362 11475 11588 11702 11816 11930 12045 12054 12164 12173 12282 12291 12401 12410
       12520 12529 12639 12648 12982 12985 12987 12990 12992 13007 13012 13110 13150 13204 13243
       13298 13350 13396 13463 13492 13543 13600 13608 13616 13630 13637 13684 13692 13697 13705
       13717 13745 13762 13768 13869 13900 14063 14075 14094 14166 14259
BPL    13529 13559 13594 13634 13701 13874 14164
BR      3098  3132  3151  3155  3220  3224  3235  3241  3247  3309  3316  3323  3345  3365  3382
        3389  3395  3401  3407  3422  3429  3437  3445  3448  3453  3456  3461  3464  3469  3472
        3477  3480  3485  3488  3493  3496  3501  3504  3508  3514  3520  3524  3587  3591  3598
        3602  3610  3615  3619  3622  3626  3629  3633  3636  3640  3643  3648  3654 12755 12765
       12771 12777 12781 12844 12850 12868 12872 12878 12883 12889 12895 12928 12997 13049 13055
       13062 13066 13070 13077 13081 13381 13387 13398 13404 13410 13414 13420 13426 13433 13465
       13471 13474 13487 13490 13540 13557 13596 13613 13623 13632 13639 13695 13764 13828 13841
       13901 13907 13915 13923 13931 13939 13947 13955 13963 13971 13979 13987 13995 14003 14011
       14019 14027 14035 14043 14051 14068 14077 14097 14104 14142 14157 14178 14251 14275
CLC     3733  3811  3891  3969  4049  4130
CLR     3099  3105  3118  3119  3148  3231  3263  3278  3299  3674  4211  4846  7410  7530  7650
        7976  8212  8449  8751  9063  9296  9529  9762 12753 12784 12800 12801 12911 13046 13145
       13197 13198 13268 13327 13459 13461 13489 13503 13532 13535 13665 13698 13734 13735 13812
       13813 14057 14061 14155 14257
CLRB   13488 13561 13612 13638 13769 13837
CMP     3106  3129  3284  3290  3304  3336  3346  3348  3350  3352  3354  3356  3366  3368  3370
        3372  3374  3404  3419  3426  3433  3441  3449  3457  3465  3473  3481  3489  3497  3557
        3607  3616  3623  3630  3637  3725  3751  3803  3828  3883  3909  3961  3986  4041  4067
        4122  4148  4195  4229  4248  4267  4285  4302  4320  4338  4385  4419  4437  4456  4474
        4492  4510  4560  4596  4630  4648  4670  4689  4709  4728  4746  4764  4782  4828  4868
        4892  4914  4936  4958  4980  5002  5024  5046  5069  5091  5112  5132  5152  5172  5223
        5264  5289  5313  5337  5361  5385  5409  5433  5457  5481  5504  5526  5547  5568  5589
        5643  5686  5712  5736  5760  5784  5808  5832  5856  5880  5904  5927  5949  5970  5991
        6012  6065  6108  6133  6157  6181  6205  6229  6253  6277  6301  6325  6348  6370  6391
        6412  6433  6486  6531  6557  6582  6607  6632  6657  6682  6707  6732  6757  6781  6804
        6826  6848  6870  6922  6944  6965  6990  7021  7047  7073  7097  7121  7145  7204  7225
        7246  7267  7286  7305  7324  7373  7393  7417  7440  7493  7513  7537  7560  7613  7633
        7657  7680  7741  7761  7781  7842  7862  7882  7900  7918  7936  7983  8004  8026  8054
        8078  8102  8124  8146  8168  8219  8240  8262  8290  8314  8338  8360  8382  8404  8457
        8479  8501  8524  8545  8566  8592  8616  8640  8662  8684  8706  8759  8781  8803  8826
        8847  8868  8894  8918  8942  8964  8986  9008  9079  9101  9128  9152  9176  9198  9220
        9242  9312  9334  9361  9385  9409  9431  9453  9475  9545  9567  9594  9618  9642  9664
        9686  9708  9778  9800  9827  9851  9875  9897  9919  9941 10000 10019 10039 10059 10113
       10133 10153 10173 10227 10246 10266 10286 10340 10359 10379 10399 10455 10475 10496 10517
       10572 10592 10612 10632 10687 10706 10726 10746 10800 10820 10840 10860 10914 10933 10953
       10973 11027 11046 11066 11086 11140 11160 11180 11200 11256 11276 11297 11318 11374 11393
       11413 11433 11487 11506 11526 11546 11600 11620 11640 11660 11714 11734 11754 11774 11828
       11847 11867 11887 11943 11963 11984 12005 12043 12066 12086 12106 12126 12162 12185 12204
       12224 12244 12280 12303 12323 12343 12363 12399 12422 12442 12462 12482 12518 12541 12560
       12580 12600 12637 12661 12681 12702 12723 12954 12942 13133 13185 13235 13297 13472 13496
       13553 13683 13691 13696 13704 13709 13711 13716 13744 13757 13899 14074 14076
CMPB   13478 13482 13605 13607 13615 13636 13640 13761 13787 13816 13818
DEC     3215  3257  3276  3532  3543  3550  3664  4849  5245  5667  6089  6512  9987 10100 10214
       10327 10441 10559 10674 10787 10901 11014 11127 11242 11361 11474 11587 11701 11815 11929
       12053 12172 12290 12409 12528 12647 12785 12804 12986 12991 13006 13011 13109 13395 13741
       14069
DECB   13619 13622 14163 14174
EMT     1278
```

```
HALT    1391   3322   3383   3505   3562   3644  13429  13438  13595  13875  14250  14274
INC     3256   3307   3335   3553   4848  12802  13495  13539  13708  13715  13743  13864  14169  14177  14258
INCB   13500  13642  13858
IOT     1379
JMP     1385   1401   3152   3217   3292   3296   3306   3412   3425   3432   3440   3534   3540   3545   3546
        3565   3559   3564  12787  12822  12854  12900  13688  13902
JSR     3147   3704   3784   3862   3942   4020   4101   4186   4375   4551   4819   5213   5633   6055   6476
        6913   7185   7360   7480   7600   7721   7731   7825   7834   7970   8045   8206   8281   8443   8583
        6745   8865   9050   9059   9068   9119   9283   9292   9301   9352   9516   9525   9534   9585   9743
        9758   9767   9818   9978   9992  10091  10105  10205  10219  10318  10332  10432  10447  10550  10564
       10665  10679  10778  10792  10892  10906  11005  11019  11118  11132  11233  11248  11352  11366  11465
       11479  11578  11592  11692  11706  11806  11820  11920  11935  12040  12059  12159  12178  12277  12296
       12396  12415  12515  12534  12634  12654  12817  13614  13621  13628  13686  13870
MOV     3097   3104   3108   3110   3111   3112   3113   3114   3115   3116   3117   3121   3122   3125   3126
        3127   3128   3133   3135   3136   3137   3141   3142   3143   3144   3145   3158   3159   3160   3161
        3162   3208   3211   3212   3213   3214   3228   3229   3230   3238   3245   3250   3254   3261   3262
        3266   3266   3267   3271   3272   3273   3274   3280   3281   3286   3287   3300   3301   3308   3311
        3312   3314   3325   3326   3328   3330   3331   3332   3333   3338   3339   3341   3343   3358   3359
        3361   3363   3376   3377   3378   3381   3392   3398   3410   3435   3443   3451   3459   3467   3475
        3483   3491   3499   3511   3517   3536   3538   3539   3549   3554   3555   3578   3579   3590   3592
        3583   3594   3606   3613   3651   3657   3659   3660   3661   3662   3667   3668   3669   3670   3671
        3672   3701   3702   3708   3709   3710   3713   3714   3715   3721   3722   3727   3739   3740   3741
        3746   3747   3748   3753   3781   3782   3788   3789   3792   3793   3794   3800   3801   3805   3817
        3818   3819   3823   3825   3826   3830   3859   3860   3866   3867   3868   3871   3872   3873   3879
        3880   3886   3897   3898   3899   3903   3905   3906   3911   3939   3940   3946   3947   3950   3951
        3956   3958   3959   3963   3975   3976   3977   3981   3983   3984   3988   4017   4018   4024   4025
        4026   4029   4030   4031   4037   4038   4043   4055   4056   4057   4061   4063   4064   4069   4098
        4099   4105   4106   4107   4110   4111   4112   4118   4119   4124   4136   4137   4138   4142   4144
        4145   4150   4183   4184   4191   4194   4226   4228   4244   4247   4264   4266   4282   4284   4299
        4301   4317   4319   4335   4337   4372   4373   4381   4384   4401   4416   4418   4433   4436   4453
        4455   4471   4473   4489   4491   4507   4509   4548   4549   4556   4559   4577   4592   4595   4613
        4627   4629   4644   4647   4667   4669   4685   4688   4706   4708   4725   4727   4743   4745   4761
        4763   4779   4781   4816   4817   4824   4827   4845   4847   4864   4867   4887   4889   4891   4909
        4911   4913   4931   4933   4935   4953   4955   4957   4975   4977   4979   4997   4999   5001   5019
        5021   5023   5041   5043   5045   5065   5068   5088   5090   5109   5111   5129   5131   5149   5151
        5169   5171   5210   5211   5219   5222   5241   5242   5243   5260   5263   5284   5286   5288   5308
        5310   5312   5332   5334   5336   5356   5358   5360   5380   5382   5384   5404   5406   5408   5428
        5430   5432   5452   5454   5456   5477   5480   5501   5503   5523   5525   5544   5546   5565   5667
        5586   5588   5630   5631   5639   5642   5662   5664   5665   5682   5685   5707   5709   5711   5731
        5733   5735   5755   5757   5759   5779   5781   5783   5803   5805   5807   5827   5829   5831   5951
        5853   5855   5875   5877   5879   5900   5903   5924   5926   5946   5948   5967   5969   5988   6152
        6006   6011   6052   6053   6061   6064   6084   6085   6086   6104   6107   6128   6130   6132   6272
        6154   6156   6176   6178   6180   6200   6202   6204   6224   6226   6228   6248   6250   6252   6411
        6274   6276   6296   6298   6300   6321   6324   6345   6347   6367   6369   6388   6390   6409   6547
        6430   6432   6473   6474   6482   6485   6506   6508   6509   6527   6530   6552   6554   6556   6577
        6579   6581   6602   6604   6606   6627   6629   6631   6652   6654   6656   6677   6679   6681   7025
        6704   6706   6727   6729   6731   6753   6756   6778   6780   6801   6803   6823   6825   6945   6947
        6867   6869   6910   6911   6919   6921   6941   6943   6962   6964   6982   6987   6989   7013   7014
        7018   7020   7043   7046   7070   7072   7094   7096   7118   7120   7142   7144   7182   7183   7193
        7201   7203   7222   7224   7242   7245   7264   7266   7283   7285   7302   7304   7321   7323   7357
        7358   7365   7370   7372   7390   7392   7414   7416   7437   7439   7477   7478   7485   7490   7498
        7510   7512   7534   7536   7557   7559   7597   7598   7605   7610   7612   7630   7632   7654   7656
        7677   7679   7718   7719   7726   7727   7730   7738   7740   7757   7760   7778   7780   7822   7837
        7831   7833   7839   7841   7858   7861   7879   7881   7897   7899   7915   7917   7933   7936   7967
        7968   7973   7977   7980   7982   7997   8001   8003   8022   8025   8051   8053   8074   8077   8099
        8101   8121   8123   8143   8145   8165   8167   8203   8204   8209   8213   8216   8218   8233   8237
```

```
                8239    8258    8261    8287    8289    8310    8313    8335    8337    8357    8359    8379    8381    8401    8403
                8440    8441    8446    8450    8451    8454    8456    8472    8476    8478    8497    8500    8519    8521    8523
                8541    8544    8563    8565    8589    8591    8612    8615    8637    8639    8659    8661    8681    8683    8703
                8705    8742    8743    8748    8752    8753    8756    8758    8774    8778    8780    8799    8802    8821    8823
                8825    8843    8846    8865    8867    8891    8893    8914    8917    8939    8941    8961    8963    8983    8985
                9005    9007    9047    9048    9055    9056    9057    9064    9065    9075    9078    9098    9100    9125    9127
                9148    9151    9173    9175    9195    9197    9217    9219    9239    9241    9280    9281    9288    9289    9290
                9297    9298    9308    9311    9331    9333    9358    9360    9381    9384    9406    9408    9428    9430    9450
                9452    9472    9474    9513    9514    9521    9522    9523    9530    9531    9541    9544    9564    9566    9591
                9593    9614    9617    9639    9641    9661    9663    9683    9685    9705    9707    9746    9747    9754    9755
                9756    9763    9764    9774    9777    9797    9799    9824    9826    9847    9850    9872    9874    9894    9896
                9916    9918    9938    9940    9975    9976    9983    9984    9986    9989    9991    9997    9999    10016   10018
               10035   10038   10056   10058   10088   10089   10096   10097   10099   10102   10104   10110   10112   10130   10132
               10149   10152   10170   10172   10202   10203   10210   10211   10213   10216   10218   10224   10226   10243   10245
               10262   10265   10283   10285   10315   10316   10323   10324   10326   10329   10331   10337   10339   10356   10358
               10375   10378   10396   10398   10429   10430   10437   10438   10440   10443   10445   10452   10454   10472   10474
               10492   10495   10514   10516   10547   10548   10555   10556   10558   10561   10563   10569   10571   10589   10591
               10608   10611   10629   10631   10662   10663   10670   10671   10673   10676   10678   10684   10686   10703   10705
               10722   10725   10743   10745   10775   10776   10783   10784   10786   10789   10791   10797   10799   10817   10819
               10836   10839   10857   10859   10889   10890   10897   10898   10900   10903   10905   10911   10913   10930   10932
               10949   10952   10970   10972   11002   11003   11010   11011   11013   11016   11018   11024   11026   11043   11045
               11062   11065   11083   11085   11115   11116   11123   11124   11126   11129   11131   11137   11139   11157   11159
               11176   11179   11197   11199   11230   11231   11238   11239   11241   11244   11246   11253   11255   11273   11275
               11293   11296   11315   11317   11349   11350   11357   11358   11360   11363   11365   11371   11373   11390   11392
               11409   11412   11430   11432   11462   11463   11470   11471   11473   11476   11478   11484   11486   11503   11505
               11522   11525   11543   11545   11575   11576   11583   11584   11586   11589   11591   11597   11599   11617   11619
               11636   11639   11657   11659   11689   11690   11697   11698   11700   11703   11705   11711   11713   11731   11733
               11750   11753   11771   11773   11803   11804   11811   11812   11814   11817   11819   11825   11827   11[illegible]   11846
               11863   11866   11884   11886   11917   11918   11925   11926   11928   11931   11933   11940   11942   11960   11962
               11980   11983   12002   12004   12037   12038   12049   12050   12052   12055   12057   12063   12065   12083   12085
               12102   12105   12123   12125   12156   12157   12168   12169   12171   12174   12176   12182   12184   12201   12203
               12220   12223   12241   12243   12274   12275   12286   12287   12289   12292   12294   12300   12302   12320   12322
               12339   12342   12360   12362   12393   12394   12405   12406   12408   12411   12413   12419   12421   12439   12441
               12458   12461   12479   12481   12512   12513   12524   12525   12527   12530   12532   12538   12540   12557   12559
               12576   12579   12597   12599   12631   12632   12643   12644   12646   12649   12651   12658   12660   12678   12680
               12698   12701   12720   12722   12752   12758   12759   12762   12768   12774   12788   12807   12811   12814   12886
               12892   12898   12908   12909   12924   12926   12929   12931   12932   12937   12944   12970   12971   12972   12973
               12975   12976   12977   12983   12988   12993   13003   13008   13013   13017   13019   13020   13047   13052   13058
               13076   13085   13086   13102   13103   13104   13105   13106   13107   13108   13111   13112   13113   13127   13130
               13132   13148   13175   13177   13179   13192   13184   13200   13202   13230   13231   13232   13233   13245   13258
               13281   13283   13302   13304   13317   13341   13346   13347   13384   13391   13392   13401   13408   13417   13423
               13437   13467   13468   13470   13473   13486   13498   13499   13501   13502   13505   13506   13522   13523   13524
               13525   13526   13527   13528   13533   13536   13556   13562   13563   13564   13565   13566   13568   13569   13597
               13598   13602   13617   13666   13667   13668   13669   13671   13706   13718   13732   13733   13736   13746   13755
               13756   13771   13772   13773   13774   13804   13805   13806   13807   13808   13810   13811   13830   13831   13832
               13833   13834   13860   13865   13878   13881   13897   13910   13918   13926   13934   13942   13950   13958   13966
               13974   13982   13990   13998   14006   14014   14022   14030   14038   14046   14054   14060   14065   14078   14079
               14084   14089   14090   14092   14095   14099   14106   14107   14138   14146   14147   14148   14154   14161   14179
               14180   14181   14182   14183   14198   14199   14203   14209   14210   14239   14240   14241   14242   14243   14244
               14245   14246   14247   14248   14249   14255   14256   14260   14261   14262   14263   14264   14265   14266   14267
               14268   14271

MOVB            3120   13504   13531   13534   13548   13551   13560   13599   13627   13635   13681   13702   13714   13742   13760
               13765   13814   13867   14139   14140   14143   14144   14145   14149   14152   14153   14172   14201

NEG            13342   13344   13530   14150
NOP            12818   12819   12820
RESET          12816
```

```
ROL      3761  3838  3919  3996  4077  4158  6088  6511 13821 13823 13825 14156 14158 14159 14160
        14162
RTI      3134 13021 13087 13507 13570 13604 13707 13719 13737 13747 13775 13835 13883 14184 14211
        14273
RTS     12912 12948 13114 13163 13217 13374 13644 13672 14108 14204
SEC      3760  3837  3918  3995  4076  4157  6087  6510
SUB      3288  3584 12760 12847 12875 12925 12930 12974 13537 13866
TRAP    14213 14222 14223 14224 14225 14228 14229 14230 14231
TST      3149  3209  3255  3268  3315 12981 13149 13203 13349 13469 13493 13542 13552 13601 13609
        13631 13670 13687 13694 13699 13739 13829 13836 13873 13879 14101 14167 14200
TSTB    13480 13544 13558 13593 13633 13700 14093
.ASCII   1500  1501 14286 14306 14330 14347 14359 14382 14402 14417 14439 14461 14483 14505 14527
        14549 14571 14593 14615 14638 14661 14684 14706 14718 14735 14776 14805 14828 14840 14859
        14870 14879 14895 14906 14924 14936 14959 14983 15012 15037 15065 15089 15117 15128 15147
        15157 15180 15190 15209 15219 15243 15262 15269 15288 15306 15313 15326 15333 15346 15354
        15367 15390 15413 15428 15438 15456 15478 15500 15522 15544 15567 15575 15589 15607 15610
        15619 15626 15634 15642 15650 15658 15666 15674 15683 15691 15699 15708 15717 15726 15735
        15744 15752 15759 15768
.ASCIZ   1499  1502  3157  3222  3226  3237  3243  3249  3391  3397  3403  3409  3424  3431  3439
        3447  3455  3463  3471  3479  3487  3495  3503  3510  3516  3522  3526  3589  3593  3600
        3604  3612  3621  3628  3635  3642  3650  3656 12757 12767 12773 12779 12783 12825 12846
        12852 12870 12874 12880 12885 12891 12897 13051 13057 13064 13068 13072 13079 13083 13383
        13389 13400 13406 13412 13416 13422 13428 13435 13779 13780 13781 13782 13784 13909 13917
        13925 13933 13941 13949 13957 13965 13973 13981 13989 13997 14005 14013 14021 14029 14037
        14045 14053 14109 14277 14298 14319 14342 14369 14375 14394 14413 14427 14449 14471 14493
        14515 14537 14559 14581 14603 14625 14648 14671 14694 14729 14747 14762 14790 14817 14849
        14889 14917 14950 14970 14995 15024 15048 15076 15100 15140 15167 15200 15231 15249 15281
        15298 15321 15341 15362 15375 15382 15398 15405 15422 15448 15466 15488 15510 15532 15554
        15582 15596 15623 15630 15638 15646 15654 15662 15670 15679 15687 15695 15704 15713 15722
        15731 15740 15749 15756 15764 15772 15777 15781
.BLKB   13653 13778
.BLKW    2830  2917  2918  2963  2967  2986  2988  2993  2996  3003  3006  3009  3012  3015  3049
        3050  3074  3076  3077  3078  3079  3080  3081  3082  3083  3084 13575
.BYTE    1455  1456  1461  1462  1470  1471  1479  1480  1481  1482  2944 12824 13776 13777 14185
        14186 14187 14188 15812 15813 15815 15817 15819 15821 15823 15825 15827 15829 15830
.DSABL  13721
.ENABL      1  1251 13649
.END    15832
.ENDC    1256  1268  1270  1271  1272  1278  1370  1384  1396  1421  1432  1443  1445  1449  1453
        1455  1483  1491  1497  1498  1499  1500  1504  2538  2544  2596  2598  2797  3102  3108
        3109  3112  3114  3116  3118  3119  3121  3123  3139  3157  3183  3184  3206  3207  3208
        3209  3222  3226  3237  3243  3249  3391  3397  3403  3409  3424  3431  3439  3447  3455
        3463  3471  3479  3487  3495  3503  3510  3516  3522  3526  3573  3574  3576  3577  3578
        3579  3589  3593  3600  3604  3612  3621  3628  3635  3642  3650  3656  3695  3696  3694
        3699  3700  3701  3766  3767  3774  3779  3780  3781  3843  3844  3852  3857  3858  3859
        3924  3925  3932  3937  3938  3939  4001  4002  4010  4015  4016  4017  4082  4083  4091
        4096  4097  4098  4167  4168  4181  4182  4183  4193  4227  4246  4265  4283  4300  4318
        4336  4356  4357  4370  4371  4372  4383  4417  4435  4454  4472  4490  4508  4527  4528
        4546  4547  4548  4558  4594  4628  4646  4668  4687  4707  4726  4744  4762  4780  4799
        4800  4814  4815  4816  4826  4866  4890  4912  4934  4956  4978  5000  5022  5044  5067
        5089  5110  5130  5150  5170  5191  5192  5208  5209  5210  5221  5262  5287  5311  5335
        5359  5383  5407  5431  5455  5479  5502  5524  5545  5566  5587  5610  5611  5628  5629
        5630  5641  5684  5710  5734  5758  5782  5806  5830  5854  5878  5902  5925  5947  5968
        5989  6010  6032  6033  6050  6051  6052  6063  6106  6131  6155  6179  6203  6227  6251
        6275  6299  6323  6346  6368  6389  6410  6431  6453  6454  6471  6472  6473  6484  6529
        6555  6580  6605  6630  6655  6680  6705  6730  6755  6779  6802  6824  6846  6968  6991
```

DERHAB.SRC CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
        6892  6908  6909  6910  6920  6942  6963  6988  7019  7045  7071  7095  7119  7143  7168
        7169  7180  7181  7182  7202  7223  7244  7265  7284  7303  7322  7342  7343  7355  7356
        7357  7371  7391  7415  7438  7462  7463  7475  7476  7477  7491  7511  7535  7558  7582
        7583  7595  7596  7597  7611  7631  7655  7678  7702  7703  7716  7717  7718  7739  7759
        7778  7798  7799  7820  7821  7822  7840  7860  7890  7898  7916  7934  7954  7955  7965
        7966  7967  7981  8002  8024  8044  8052  8076  8100  8122  8144  8166  8190  8191  8201
        8202  8203  8217  8238  8260  8280  8288  8312  8336  8358  8380  8402  8426  8427  8438
        8439  8440  8455  8477  8499  8522  8543  8564  8582  8590  8614  8638  8660  8682  8704
        8728  8729  8740  8741  8742  8757  8779  8801  8824  8845  8866  8884  8892  8916  8940
        8962  8984  9006  9031  9032  9045  9046  9047  9077  9099  9126  9150  9174  9196  9218
        9240  9264  9265  9278  9279  9280  9310  9332  9359  9383  9407  9429  9451  9473  9497
        9498  9511  9512  9513  9543  9565  9592  9616  9640  9662  9684  9706  9730  9731  9744
        9745  9746  9776  9798  9825  9849  9873  9895  9917  9939  9964  9965  9973  9974  9975
        9981  9994  9995  9998 10017 10034 10037 10055 10057 10077 10078 10086 10087 10088 10094
       10107 10108 10111 10131 10148 10151 10169 10171 10191 10192 10200 10201 10202 10208 10221
       10222 10225 10244 10261 10264 10282 10284 10304 10305 10313 10314 10315 10321 10334 10336
       10338 10357 10374 10377 10395 10397 10417 10418 10427 10428 10429 10435 10449 10450 10453
       10473 10491 10494 10513 10515 10536 10537 10545 10546 10547 10553 10566 10567 10570 10590
       10607 10610 10628 10630 10651 10652 10660 10661 10662 10668 10679 10681 10682 10685 10704
       10721 10724 10742 10744 10764 10765 10773 10774 10775 10781 10792 10794 10795 10798 10818
       10835 10838 10856 10858 10878 10879 10887 10888 10889 10895 10906 10908 10909 10912 10931
       10948 10951 10969 10971 10991 10992 11000 11001 11002 11008 11019 11021 11022 11025 11044
       11061 11064 11082 11084 11104 11105 11113 11114 11115 11121 11132 11134 11135 11138 11158
       11175 11179 11196 11198 11218 11219 11228 11229 11230 11236 11248 11250 11251 11254 11274
       11292 11295 11314 11316 11338 11339 11347 11348 11349 11355 11366 11368 11369 11372 11389
       11391 11408 11411 11429 11431 11451 11452 11460 11461 11462 11468 11479 11481 11482 11485
       11502 11504 11521 11524 11542 11544 11564 11565 11573 11574 11575 11581 11592 11594 11595
       11598 11616 11618 11635 11638 11656 11658 11678 11679 11687 11688 11689 11695 11706 11708
       11709 11712 11730 11732 11749 11752 11770 11772 11792 11793 11801 11802 11803 11809 11820
       11822 11823 11826 11843 11845 11862 11865 11883 11885 11905 11906 11915 11916 11917 11923
       11935 11937 11938 11941 11959 11961 11979 11982 12001 12003 12025 12026 12035 12036 12037
       12047 12058 12061 12064 12084 12101 12104 12122 12124 12144 12145 12154 12155 12156 12166
       12177 12180 12183 12202 12219 12222 12240 12242 12262 12263 12272 12273 12274 12284 12295
       12298 12301 12321 12338 12341 12359 12361 12381 12382 12391 12392 12393 12403 12414 12417
       12420 12440 12457 12460 12478 12480 12500 12501 12510 12511 12512 12522 12533 12536 12539
       12558 12575 12578 12596 12598 12618 12619 12629 12630 12631 12641 12653 12656 12659 12679
       12697 12700 12719 12721 12745 12746 12750 12751 12752 12753 12757 12767 12773 12779 12783
       12793 12794 12795 12797 12800 12806 12809 12810 12814 12816 12822 12824 12825 12828 12839
       12841 12846 12852 12864 12866 12870 12874 12880 12885 12891 12897 13051 13057 13064 13068
       13072 13079 13083 13131 13183 13383 13389 13400 13406 13412 13416 13422 13428 13435 13447
       13450 13455 13462 13464 13475 13478 13479 13480 13482 13484 13491 13495 13500 13501 13505
       13508 13509 13512 13579 13599 13649 13680 13689 13720 13721 13725 13736 13748 13749 13756
       13758 13761 13763 13779 13780 13786 13793 13799 13843 13846 13849 13858 13865 13870 13871
       13872 13873 13883 13884 13909 13917 13925 13933 13941 13949 13957 13965 13973 13981 13989
       13997 14005 14013 14021 14029 14037 14045 14053 14116 14193 14199 14202 14221 14222 14223
       14224 14225 14226 14227 14228 14229 14230 14231 14238 14247 14248 14254 14260 14261 14271
       14273 14280
.EQUIV  1278  1279  1287  1302  1303  1332  1333  1334  1335  1336  1337  1338  1339  1340  1341
        1360  1361  1362  1363  1364  1365  1366  1367  1368  1369
.EVEN   3157  3222  3226  3237  3243  3249  3391  3397  3403  3409  3424  3431  3439  3447  3455
        3463  3471  3479  3487  3495  3503  3510  3516  3522  3526  3589  3593  3600  3604  3612
        3621  3628  3635  3642  3650  3656 12757 12767 12773 12779 12783 12846 12852 12870 12874
       12880 12885 12891 12897 13051 13057 13064 13068 13072 13079 13083 13383 13389 13400 13406
       13412 13416 13422 13428 13435 13655 13909 13917 13925 13933 13941 13949 13957 13965 13973
       13981 13989 13997 14005 14013 14021 14029 14037 14045 14053 14110 14279 15787 15831
.IF      1252  1268  1269  1270  1271  1272  1276  1342  1370  1394  1421  1432  1443  1444  1448
```

1452	1454	1483	1491	1497	1498	1499	1503	1504	2537	2543	2595	2597	2796	3102
3103	3108	3110	3112	3114	3116	3118	3119	3121	3139	3156	3182	3184	3206	3208
3209	3221	3225	3236	3242	3248	3390	3396	3402	3408	3423	3430	3438	3446	3454
3462	3470	3478	3486	3494	3502	3509	3515	3521	3525	3572	3574	3576	3578	3579
3588	3592	3599	3603	3611	3620	3627	3634	3641	3649	3655	3684	3686	3693	3699
3701	3765	3767	3774	3779	3781	3842	3844	3851	3857	3859	3923	3925	3932	3937
3939	4000	4002	4009	4015	4017	4081	4083	4090	4096	4098	4166	4168	4181	4183
4193	4227	4246	4265	4283	4300	4318	4336	4355	4357	4370	4372	4383	4417	4435
4454	4472	4490	4508	4526	4528	4546	4548	4558	4594	4628	4646	4668	4687	4707
4726	4744	4762	4780	4798	4800	4814	4816	4826	4866	4890	4912	4934	4956	4978
5000	5022	5044	5067	5089	5110	5130	5150	5170	5190	5192	5208	5210	5221	5262
5287	5311	5335	5359	5383	5407	5431	5455	5479	5502	5524	5545	5566	5587	5609
5611	5628	5630	5641	5684	5710	5734	5758	5782	5806	5830	5854	5878	5902	5925
5947	5968	5989	6010	6031	6033	6050	6052	6063	6106	6131	6155	6179	6203	6227
6251	6275	6299	6323	6346	6368	6389	6410	6431	6452	6454	6471	6473	6484	6529
6555	6580	6605	6630	6655	6680	6705	6730	6755	6779	6802	6824	6846	6868	6890
6892	6908	6910	6920	6942	6963	6988	7019	7045	7071	7095	7119	7143	7167	7169
7180	7182	7202	7223	7244	7265	7284	7303	7322	7341	7343	7355	7357	7371	7391
7415	7438	7461	7463	7475	7477	7491	7511	7535	7558	7581	7583	7595	7597	7611
7631	7655	7678	7701	7703	7716	7718	7739	7759	7779	7797	7799	7820	7822	7840
7860	7880	7898	7916	7934	7953	7955	7965	7967	7981	8002	8024	8044	8052	8076
8100	8122	8144	8166	8189	8191	8201	8203	8217	8238	8260	8280	8288	8312	8336
8358	8380	8402	8425	8427	8438	8440	8455	8477	8499	8519	8522	8543	8564	8590
8614	8638	8660	8682	8704	8727	8729	8740	8742	8757	8779	8801	8821	8824	8845
8866	8892	8916	8940	8962	8984	9006	9030	9032	9045	9047	9077	9099	9126	9150
9174	9196	9218	9240	9263	9265	9278	9280	9310	9332	9359	9383	9407	9429	9451
9473	9496	9498	9511	9513	9543	9565	9592	9616	9640	9662	9684	9706	9729	9731
9744	9746	9776	9798	9825	9849	9873	9895	9917	9939	9963	9965	9973	9975	9981
9992	9994	9995	9998	10015	10017	10034	10037	10057	10076	10078	10086	10088	10094	10105
10107	10108	10111	10129	10131	10148	10151	10171	10190	10192	10200	10202	10208	10219	10221
10222	10225	10242	10244	10261	10264	10284	10303	10305	10313	10315	10321	10332	10334	10335
10338	10355	10357	10374	10377	10397	10416	10418	10427	10429	10435	10447	10449	10450	10453
10471	10473	10491	10494	10513	10515	10535	10537	10545	10547	10553	10564	10566	10567	10570
10588	10590	10607	10610	10630	10650	10652	10660	10662	10668	10679	10681	10682	10685	10702
10704	10721	10724	10744	10763	10765	10773	10775	10781	10792	10794	10795	10798	10816	10818
10835	10838	10858	10877	10879	10887	10889	10895	10906	10908	10909	10912	10929	10931	10948
10951	10971	10990	10992	11000	11002	11008	11019	11021	11022	11025	11042	11044	11061	11064
11084	11103	11105	11113	11115	11121	11132	11134	11135	11138	11156	11158	11175	11178	11198
11217	11219	11228	11230	11236	11248	11250	11251	11254	11272	11274	11292	11295	11314	11316
11337	11339	11347	11349	11355	11366	11369	11372	11389	11391	11408	11411	11431	11450	11452
11460	11462	11468	11479	11482	11485	11502	11504	11521	11524	11544	11563	11565	11573	11575
11581	11592	11595	11598	11616	11618	11635	11638	11658	11677	11679	11687	11689	11695	11706
11709	11712	11730	11732	11749	11752	11772	11791	11793	11801	11803	11809	11820	11823	11826
11843	11845	11862	11865	11885	11904	11906	11915	11917	11923	11935	11938	11941	11959	11961
11979	11982	12001	12003	12024	12026	12035	12037	12043	12058	12059	12064	12082	12084	12101
12104	1212[illegible]	12143	12145	12154	12156	12162	12177	12178	12183	12200	12202	12219	12222	12242
12261	1226[illegible]	12272	12274	12280	12295	12296	12301	12319	12321	12338	12341	12361	12380	12382
12391	12393	12399	12414	12415	12420	12438	12440	12457	12460	12480	12499	12501	12510	12512
12518	12533	12534	12539	12556	12558	12575	12578	12598	12617	12619	12629	12631	12637	12653
12654	12659	12677	12679	12697	12700	12719	12721	12744	12746	12750	12752	12753	12756	12766
12772	12778	12782	12792	12793	12794	12795	12796	12797	12799	12805	12808	12810	12814	12816
12822	12824	12825	12838	12840	12845	12851	12863	12865	12869	12873	12879	12884	12890	12896
13050	13056	13063	13067	13071	13078	13082	13131	13183	13382	13388	13399	13405	13411	13415
13421	13427	13434	13446	13449	13454	13459	13462	13474	13476	13477	13478	13480	13481	13482
13491	13493	13501	13502	13507	13508	13509	13511	13578	13599	13648	13650	13678	13693	13699
13721	13724	13725	13735	13748	13756	13757	13761	13762	13778	13779	13786	13792	13795	13811

```
            13845  13848  13858  13861  13868  13870  13871  13873  13876  13883  13884  13908  13916  13924  13932
            13940  13948  13956  13964  13972  13980  13988  13996  14004  14012  14020  14028  14036  14044  14052
            14115  14192  14198  14202  14213  14222  14223  14224  14225  14226  14227  14228  14229  14230  14231
            14237  14247  14248  14253  14260  14261  14269  14271  14273  14277
.IFF         1268   1270   1271   1272   1278   1445   1449   1452   1454   1483   1504   2538   2544   2596   2598
             2797   3108   3183   3184   3207   3208   3209   3573   3574   3577   3578   3579   3685   3686   3700
             3701   3766   3767   3780   3781   3843   3844   3858   3859   3924   3925   3938   3939   4001   4002
             4016   4017   4082   4083   4097   4098   4167   4168   4182   4183   4356   4357   4371   4372   4527
             4528   4547   4548   4799   4800   4815   4816   5191   5192   5209   5210   5610   5611   5629   5630
             6032   6033   6051   6052   6453   6454   6472   6473   6891   6892   6909   6910   7168   7169   7181
             7182   7342   7343   7356   7357   7462   7463   7476   7477   7582   7583   7596   7597   7702   7703
             7717   7718   7798   7799   7821   7822   7954   7955   7966   7967   8190   8191   8202   8203   8426
             8427   8439   8440   8728   8729   8741   8742   9031   9032   9046   9047   9264   9265   9279   9280
             9497   9498   9512   9513   9730   9731   9745   9746   9964   9965   9974   9975  10077  10078  10087
            10088  10191  10192  10201  10202  10304  10305  10314  10315  10417  10418  10428  10429  10536  10537
            10546  10547  10651  10652  10661  10662  10764  10765  10774  10775  10878  10879  10888  10889  10991
            10992  11001  11002  11104  11105  11114  11115  11218  11219  11229  11230  11338  11339  11348  11349
            11451  11452  11461  11462  11564  11565  11574  11575  11678  11679  11688  11689  11792  11793  11802
            11803  11905  11906  11916  11917  12025  12026  12036  12037  12144  12145  12155  12156  12262  12263
            12273  12274  12381  12382  12392  12393  12500  12501  12511  12512  12618  12619  12630  12631  12745
            12746  12751  12752  12753  12793  12796  12800  12806  12809  12824  12839  12841  12864  12866  13447
            13475  13478  13479  13482  13508  13512  13579  13649  13721  13725  13728  13736  13748  13749  13757
            13762  13778  13793  13846  13848  13861  13883  13884  14116  14193  14199  14238  14254  14271
.IFT         3157   3222   3226   3237   3243   3249   3391   3397   3403   3409   3424   3431   3439   3447   3455
             3463   3471   3479   3487   3495   3503   3510   3516   3522   3526   3589   3593   3600   3604   3612
             3621   3628   3635   3642   3650   3656  12757  12767  12773  12779  12783  12846  12852  12870  12874
            12880  12885  12891  12897  13051  13057  13064  13068  13072  13079  13083  13383  13389  13400  13406
            13412  13416  13422  13428  13435  13490  13727  13732  13816  13836  13843  13871  13909  13917  13925
            13933  13941  13949  13957  13965  13973  13981  13989  13997  14005  14013  14021  14029  14037  14045
            14053
.IFTF        3157   3222   3226   3237   3243   3249   3391   3397   3403   3409   3424   3431   3439   3447   3455
             3463   3471   3479   3487   3495   3503   3510   3516   3522   3526   3589   3593   3600   3604   3612
             3621   3628   3635   3642   3650   3656  12757  12767  12773  12779  12783  12846  12852  12870  12874
            12880  12885  12891  12897  13051  13057  13064  13068  13072  13079  13083  13383  13389  13400  13406
            13412  13416  13422  13428  13435  13488  13721  13725  13728  13812  13820  13842  13870  13909  13917
            13925  13933  13941  13949  13957  13965  13973  13981  13989  13997  14005  14013  14021  14029  14037
            14045  14053
.IIF         1251   1256   1260   1265   1266   1267   1268   1271   1272   1273   1274   1391   1503   3109   3112
             3118   3119   3121   3122   3710   3723   3750   3790   3802   3828   3868   3881   3908   3948   3960
             3986   4026   4039   4066   4107   4120   4147   4192   4204   4205   4214   4227   4239   4240   4245
             4258   4259   4265   4277   4278   4283   4295   4300   4312   4313   4318   4330   4331   4336   4348
             4349   4382   4394   4395   4404   4417   4429   4434   4447   4448   4454   4466   4467   4472   4484
             4485   4490   4502   4503   4508   4520   4521   4557   4570   4571   4580   4593   4606   4607   4616
             4628   4640   4645   4658   4659   4668   4681   4686   4700   4701   4707   4720   4721   4726   4739
             4744   4757   4762   4775   4780   4793   4825   4837   4838   4853   4865   4879   4880   4890   4905
             4912   4927   4934   4949   4956   4971   4978   4993   5000   5015   5022   5037   5044   5059   5066
             5082   5083   5089   5104   5105   5110   5125   5130   5145   5150   5165   5170   5185   5220   5233
             5234   5248   5261   5276   5277   5287   5303   5311   5327   5335   5351   5359   5375   5383   5399
             5407   5423   5431   5447   5455   5471   5478   5495   5496   5502   5518   5519   5524   5540   5545
             5561   5566   5582   5587   5603   5640   5653   5654   5670   5683   5699   5700   5710   5726   5734
             5750   5758   5774   5782   5798   5806   5822   5830   5846   5854   5870   5878   5894   5901   5918
             5919   5925   5941   5942   5947   5963   5968   5984   5989   6005   6010   6026   6062   6075   6076
             6092   6105   6120   6121   6131   6147   6155   6171   6179   6195   6203   6219   6227   6243   6251
             6267   6275   6291   6299   6315   6322   6339   6340   6346   6362   6363   6368   6384   6389   6405
             6410   6426   6431   6447   6483   6496   6497   6515   6528   6544   6545   6555   6572   6590   6597
             6605   6622   6630   6647   6655   6672   6680   6697   6705   6722   6730   6747   6754   6772   6773
```

```
 6779  6796  6797  6802  6819  6824  6841  6846  6863  6868  6885  6920  6931  6942  6954
 6963  6975  6976  6988  7006  7007  7019  7038  7039  7044  7064  7065  7071  7090  7095
 7114  7119  7138  7143  7162  7202  7216  7217  7223  7237  7238  7243  7258  7259  7265
 7279  7284  7298  7303  7317  7322  7336  7371  7383  7384  7391  7404  7405  7415  7431
 7438  7455  7456  7491  7503  7504  7511  7524  7525  7535  7551  7558  7575  7576  7611
 7623  7624  7631  7644  7645  7655  7671  7678  7695  7696  7739  7752  7753  7758  7772
 7773  7779  7792  7840  7853  7854  7859  7873  7874  7880  7893  7898  7911  7916  7929
 7934  7947  7957  7958  7961  7978  7979  7981  7992  7993  7994  8002  8016  8017  8021
 8023  8038  8039  8052  8069  8070  8075  8093  8094  8100  8117  8122  8139  8144  8161
 8166  8183  8193  8194  8197  8214  8215  8217  8228  8229  8230  8238  8252  8253  8257
 8259  8274  8275  8288  8305  8306  8311  8329  8330  8336  8353  8358  8375  8380  8397
 8402  8419  8429  8430  8431  8433  8451  8453  8455  8466  8468  8469  8472  8477  8491
 8492  8496  8498  8513  8514  8518  8522  8536  8537  8542  8557  8558  8564  8578  8579
 8590  8607  8608  8613  8631  8632  8638  8655  8660  8677  8682  8699  8704  8721  8731
 8732  8733  8735  8753  8755  8757  8768  8770  8771  8774  8779  8793  8794  8798  8800
 8815  8816  8820  8824  8838  8839  8844  8859  8860  8866  8880  8881  8892  8909  8910
 8915  8933  8934  8940  8957  8962  8979  8984  9001  9006  9023  9074  9076  9091  9092
 9096  9099  9113  9114  9126  9143  9144  9149  9167  9168  9174  9191  9196  9213  9218
 9235  9240  9257  9307  9309  9324  9325  9329  9332  9346  9347  9359  9376  9377  9382
 9400  9401  9407  9424  9429  9446  9451  9468  9473  9490  9540  9542  9557  9558  9565
 9579  9580  9592  9609  9610  9615  9633  9634  9640  9657  9662  9679  9684  9701  9706
 9723  9773  9775  9790  9791  9798  9812  9813  9825  9842  9843  9848  9866  9867  9873
 9890  9895  9912  9917  9934  9939  9956  9968  9972  9991  9992  9996  9998 10009 10011
10017 10028 10030 10031 10034 10036 10048 10050 10051 10055 10057 10068 10070 10081 10085
10104 10105 10109 10111 10122 10124 10125 10128 10131 10142 10144 10145 10148 10150 10162
10164 10165 10169 10171 10182 10184 10195 10199 10218 10219 10223 10225 10236 10238 10244
10255 10257 10258 10261 10263 10275 10277 10278 10282 10284 10295 10297 10308 10312 10331
10332 10336 10338 10349 10351 10357 10368 10370 10371 10374 10376 10388 10390 10391 10395
10397 10408 10410 10421 10426 10445 10446 10451 10453 10464 10467 10473 10484 10487 10488
10491 10493 10505 10508 10509 10513 10515 10526 10529 10540 10544 10563 10564 10568 10570
10581 10583 10584 10587 10590 10601 10603 10604 10607 10609 10621 10623 10624 10628 10630
10641 10643 10655 10659 10678 10679 10683 10685 10696 10698 10704 10715 10717 10718 10721
10723 10735 10737 10738 10742 10744 10755 10757 10768 10772 10791 10792 10796 10798 10809
10811 10812 10815 10818 10829 10831 10832 10835 10837 10849 10851 10852 10856 10858 10869
10871 10882 10886 10905 10906 10910 10912 10923 10925 10931 10942 10944 10945 10948 10950
10962 10964 10965 10969 10971 10982 10984 10995 10999 11018 11019 11023 11025 11036 11038
11044 11055 11057 11058 11061 11063 11075 11077 11078 11082 11084 11095 11097 11108 11112
11131 11132 11136 11138 11149 11151 11152 11155 11158 11169 11171 11172 11175 11177 11189
11191 11192 11196 11198 11209 11211 11222 11227 11246 11247 11252 11254 11265 11268 11274
11285 11288 11289 11292 11294 11306 11309 11310 11314 11316 11327 11330 11342 11346 11365
11366 11370 11372 11383 11385 11391 11402 11404 11405 11408 11410 11422 11424 11425 11431
11442 11444 11455 11459 11478 11479 11483 11485 11496 11498 11504 11515 11517 11518 11521
11523 11535 11537 11538 11544 11555 11557 11568 11572 11591 11592 11596 11598 11609 11611
11612 11615 11618 11629 11631 11632 11635 11637 11649 11651 11652 11658 11669 11671 11682
11686 11705 11706 11710 11712 11723 11725 11726 11729 11732 11743 11745 11746 11749 11751
11763 11765 11766 11772 11783 11785 11796 11800 11819 11820 11824 11826 11837 11839 11845
11856 11858 11859 11862 11864 11876 11878 11879 11885 11896 11898 11909 11914 11933 11934
11939 11941 11952 11955 11961 11972 11975 11976 11979 11981 11993 11996 11997 12003 12014
12017 12029 12033 12057 12058 12062 12064 12075 12077 12078 12081 12084 12095 12097 12098
12101 12103 12115 12117 12118 12122 12124 12135 12137 12148 12152 12176 12177 12181 12183
12194 12196 12202 12213 12215 12216 12219 12221 12233 12235 12236 12240 12242 12253 12255
12266 12270 12294 12295 12299 12301 12312 12314 12315 12318 12321 12332 12334 12335 12338
12340 12352 12354 12355 12359 12361 12372 12374 12385 12389 12413 12414 12418 12420 12431
12433 12434 12437 12440 12451 12453 12454 12457 12459 12471 12473 12474 12478 12480 12491
12493 12504 12508 12532 12533 12537 12539 12550 12552 12559 12569 12571 12572 12575 12577
12589 12591 12592 12596 12598 12609 12611 12622 12627 12651 12652 12657 12659 12670 12673
```

```
                12679   12690   12693   12694   12697   12699   12711   12714   12715   12719   12721   12732   12735   12794   12800
                12801   12812   12824   12828   13131   13140   13141   13154   13183   13192   13193   13208   13248   13258   13270
                13286   13307   13317   13329   13354   13365   13450   13451   13452   13453   13454   13455   13459   13489   13490
                13505   13508   13509   13646   13649   13655   13690   13771   13779   13786   13843   13849   13850   13851   13852
                13853   13858   13876   13883   13884   14221   14222   14223   14224   14225   14228   14229   14230   14231
.IRP             3102    3182    3572    3684    3765    3842    3923    4000    4081    4166    4355    4526    4798    5190    5609
                 6031    6452    6890    7167    7341    7461    7581    7701    7797    7953    8189    8425    8727    9030    9263
                 9496    9729    9963   10076   10190   10303   10416   10535   10650   10763   10877   10990   11103   11217   11337
                11450   11563   11677   11791   11904   12024   12143   12261   12380   12499   12617   12744   12970   13019   13102
                13111   13459   13522   13562   13806   13832   14241   14247   14260   14261
.LIST               1    1251    1261    1271    1384    1391    1483    1485    1486    1487    1488    1489    1490    1491    1492
                 1493    1494    1495    1496    1497    3102    3123    3139    3157    3182    3208    3222    3226    3237    3243
                 3249    3391    3397    3403    3409    3424    3431    3439    3447    3455    3463    3471    3479    3487    3495
                 3503    3510    3516    3522    3526    3572    3578    3589    3593    3600    3604    3612    3621    3628    3635
                 3642    3650    3656    3684    3701    3765    3781    3842    3859    3923    3939    4000    4017    4081    4098
                 4166    4183    4355    4372    4526    4548    4798    4816    5190    5210    5609    5630    6031    6052    6452
                 6473    6890    6910    7167    7182    7341    7357    7461    7477    7581    7597    7701    7718    7797    7822
                 7953    7967    8189    8203    8425    8440    8727    8742    9030    9047    9263    9280    9496    9513    9729
                 9746    9963    9975   10076   10088   10190   10202   10303   10315   10416   10429   10535   10547   10650   10662
                10763   10775   10877   10889   10990   11002   11103   11115   11217   11230   11337   11349   11450   11462   11563
                11575   11677   11689   11791   11803   11904   11917   12024   12037   12143   12156   12261   12274   12380   12393
                12499   12512   12617   12631   12744   12752   12757   12767   12773   12779   12783   12800   12816   12846   12852
                12870   12874   12880   12885   12891   12897   13051   13057   13064   13068   13072   13079   13083   13383   13389
                13400   13406   13412   13416   13422   13428   13435   13454   13748   13883   13909   13917   13925   13933   13941
                13949   13957   13965   13973   13981   13989   13997   14005   14013   14021   14029   14037   14045   14053   14213
                14221   14222   14223   14224   14225   14226   14228   14229   14230   14231   14232
.MACRO              1    1261    1272    1446    3181    3572    3684    3765    3842    3923    4000    4081    4165    4190    4224
                 4354    4380    4414    4525    4555    4591    4626    4664    4797    4823    4963    4885    4886    4909    4931
                 4953    4975    4997    5019    5041    5063    5189    5217    5258    5282    5283    5307    5331    5355    5379
                 5403    5427    5451    5475    5608    5637    5680    5705    5706    5730    5754    5778    5802    5826    5850
                 5874    5898    6030    6059    6102    6126    6127    6151    6175    6199    6223    6247    6271    6295    6319
                 6451    6480    6525    6550    6551    6576    6601    6626    6651    6676    6701    6726    6751    6889    6917
                 6939    6984    6985    7016    7166    7199    7340    7367    7368    7388    7412    7435    7460    7487    7488
                 7508    7532    7555    7580    7607    7608    7628    7652    7675    7700    7736    7796    7837    7952    7978
                 7979    7999    8000    8049    8188    8214    8215    8235    8236    8285    8424    8452    8453    8474    8475
                 8587    8726    8754    8755    8776    8777    8889    9029    9071    9072    9123    9262    9304    9305    9356
                 9495    9537    9538    9589    9728    9770    9771    9822    9962    9995   10075   10108   10189   10222   10302
                10335   10415   10450   10534   10567   10649   10682   10762   10795   10876   10909   10989   11022   11102   11135
                11216   11251   11336   11369   11449   11482   11562   11595   11676   11709   11790   11823   11903   11938   12023
                12061   12142   12180   12260   12298   12379   12417   12498   12536   12616   12656   12743   13128   13176   14213
.MCALL           1251    1384    3123
.NLIST              1    1251    1261    1271    1384    1391    1483    1485    1486    1487    1488    1489    1490    1491    1492
                 1493    1494    1495    1496    1497    3102    3123    3139    3157    3182    3208    3222    3226    3237    3243
                 3249    3391    3397    3403    3409    3424    3431    3439    3447    3455    3463    3471    3479    3487    3495
                 3503    3510    3516    3522    3526    3572    3578    3589    3593    3600    3604    3612    3621    3628    3635
                 3642    3650    3656    3684    3701    3765    3781    3842    3859    3923    3939    4000    4017    4081    4098
                 4166    4183    4355    4372    4526    4548    4798    4816    5190    5210    5609    5630    6031    6052    6452
                 6473    6890    6910    7167    7182    7341    7357    7461    7477    7581    7597    7701    7718    7797    7822
                 7953    7967    8189    8203    8425    8440    8727    8742    9030    9047    9263    9280    9496    9513    9729
                 9746    9963    9975   10076   10088   10190   10202   10303   10315   10416   10429   10535   10547   10650   10662
                10763   10775   10877   10889   10990   11002   11103   11115   11217   11230   11337   11349   11450   11462   11563
                11575   11677   11689   11791   11803   11904   11917   12024   12037   12143   12156   12261   12274   12380   12393
                12499   12512   12617   12631   12744   12752   12757   12767   12773   12779   12783   12800   12816   12846   12852
                12870   12874   12880   12885   12891   12897   13051   13057   13064   13068   13072   13079   13083   13383   13389
                13400   13406   13412   13416   13422   13428   13435   13454   13748   13883   13909   13917   13925   13933   13941
                13949   13957   13965   13973   13981   13989   13997   14005   14013   14021   14029   14037   14045   14053   14213
```

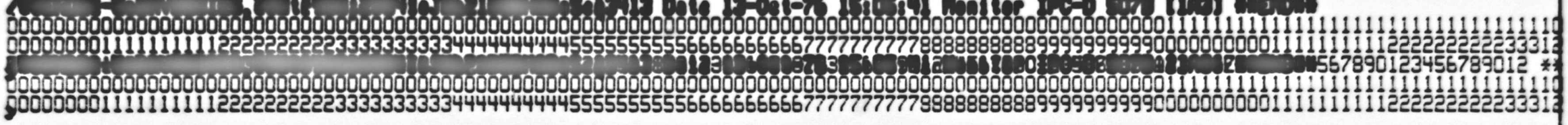
J10

```
MAINDEC-11-DERHAB-A     MACY11 27(732)  22-SEP-76  15:11  PAGE 338
DERHAB.SRC      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

        14221   14222   14223   14224   14225   14226   14228   14229   14230   14231   14232
.PAGE    1446    1504    2536    2798    2842    3096   12790   12913   12950   13022   13115   13375   13431   13443   13509
        13843   14190   14235   14280
.REM        1
.REPT    1391    1485    1491
.SBTTL   1261    1274    1385    1394    1409    1446    1504    2841    3096    3102    3182    3572    3684    3765    3842
         3923    4000    4081    4166    4355    4526    4798    5190    5609    6031    6452    6890    7167    7341    7461
         7581    7701    7797    7953    8189    8425    8727    9030    9263    9496    9729    9963   10076   10190   10303
        10416   10535   10650   10763   10877   10990   11103   11217   11337   11450   11563   11677   11791   11904   12024
        12143   12261   12380   12499   12617   12744   12790   12831   13116   13165   13219   13444   13509   13576   13646
        13790   13843   13885   14113   14190   14213   14235
.TITLE   1251
.WORD    1391    1392    1393    1454    1457    1458    1459    1460    1463    1464    1465    1466    1467    1468    1469
         1472    1473    1474    1483    1485    1486    1487    1488    1489    1490    1491    1492    1493    1494    1495
         1496    2914    3031    3032    3033    3064   12805   12808   12823   13643   13650   13651   13652   13839   13842
        14082   14087   14189   14220   14270   14272   15789   15790   15792   15794   15796   15798   15800   15802   15804
        15806   15808


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DERHAB.SEQ/SOL/CRF/PAGNUM/NL:TOC=DERHAB.SML,DERHAB.SRC
RUN-TIME:  106 158 18 SECONDS
RUN-TIME RATIO:  1222/283=4.3
CORE USED:  38K  (75 PAGES)
```

Spooler runtime 51 Seconds, 208 KCS, 1328 disk reads, 3 disk writes, 333 pages

[redacted] Monitor IPC-D 5078 [102] [redacted]