RH70

CONTROLLER DIAGNOSTIC

CERHADO

AH-8001D-MC

FICHE 1 OF 2

APR 1978

digital

MADE IN USA

RH70

CONTROLLER DIAGNOSTIC
CERHADO

AH-8001D-MC

FICHE 2 OF 2

APR 1978
digital
MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE:	AC-7999D-MC
PRODUCT NAME:	CERHADO RH70 CTRLR DIAG
PRODUCT DATE:	MARCH 1978
MAINTAINER:	DIAGNOSTIC ENGINEERING

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
****
CERHADO
****

```
                        2       COPYRIGHT (C) 1975, 1977
                                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
```

```
        *************************************************************
139     TRAP CATCHER
        *************************************************************

        142     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

        *************************************************************
146     STARTING ADDRESS(ES)
        *************************************************************

        152     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

        *************************************************************
160     MEMORY MANAGEMENT DEFINITIONS
        *************************************************************

        162     KT11 VECTOR ADDRESS

        166     KT11 STATUS REGISTER ADDRESSES

        173     KERNEL "I" PAGE DESCRIPTOR REGISTERS

        184     KERNEL "I" PAGE ADDRESS REGISTERS

        *************************************************************
199     COMMON TAGS
        *************************************************************

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

```
                                CERHADO


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

```
                                        CERHADO


                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.
```

```
                                        CERHADO


                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
```

```
                                CERHADO


                                        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"
```

```
                                CERHADO


                                        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
```

```
                                CERHADO

                                        "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)
```

```
                                        CERHADO


                                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" (BIT #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 #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
```

```
                                CERHADO

                        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)
```

```
                                CERHAD0

                                        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

                        7883            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
```

```
                                        CERHADO

                                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
```

```
                                        CERHADO
```

```
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
```

```
                                CERHADO

                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
```

```
                                        CERHADO


                                TEST 57 TEST DBL (RHCS3 BIT #10) TEST P

                                        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

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

                                        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

                                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 BO
                                        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

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

                                        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

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

                                        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
```

```
                                        CERHADO


                                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 RE\_RSE 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
```

```
                                CERHADO

                        11400   ;*******************************************************************
                        11401   ;END OF PASS ROUTINE
                        11402   ;*******************************************************************
                        11403   ;
                        11404   ;       INCREMENT THE PASS NUMBER ($PASS)
                        11405   ;       TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                        11406   ;       IF THERES A MONITOR GO TO IT
                        11407   ;       IF THERE ISN'T JUMP TO TST2
                        11408   ;
                        11409   ;*******************************************************************
                        11410   ;SUBROUTINES
                        11411   ;*******************************************************************
                        11412   ;
                        11413   ;*******************************************************************
                        11414   ;RS DATA TRANSFER SUBROUTINE
                        11415   ;*******************************************************************
                        11416   ;
                        11417   ;       THIS SUBROUTINE WRITES OR READS OR DOES A
                        11418   ;       WRITE CHECK ON CYLINDER 0 TRACK 0 SECTOR 0
                        11419   ;       IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY
                        11420   ;       FILLED
                        11421   ;       WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
                        11422   ;       MAIN PROGRAM IT MEANS THE TRANSFER IS COMPLETE
                        11423   ;
                        11424   ;       THE CALL IS BY A JSR R0,@COMND COMAND
                        11425   ;
                        11426   ;*******************************************************************
                        11427   ;RP04 DATA TRANSFER SUBROUTINE
                        11428   ;*******************************************************************
                        11429   ;
                        11430   ;       THIS SUBROUTINE WRITE/READ/WRITE CHECK ON THE RP04 CYLIN
                        11431   ;       TRACK 0, SECTOR 0.
                        11432   ;
                        11433   ;       IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
                        11434   ;       WHEN IT RETURNS FROM THIS SUBROUTINE TO THE MAIN
                        11435   ;       PROGRAM IT MEANS THE COMMAND IS COMPLETE
                        11436   ;
                        11437   ;       THE CALL IS BY A JSR R0,@COMND COMMAND.
```

```
                                CERHADO

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

```
                                CERHADO

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

```
                                CERHADO

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

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

                        ;*******************************************************************
                        ;ERROR MESSAGE TYPEOUT ROUTINE
                        ;*******************************************************************

                        ;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

                        ;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.

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

```
                                CERHADO

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

```
                                CERHADO

                                        ;*****************************************************************
                        12746           ;POWER DOWN AND UP ROUTINES
                                        ;*****************************************************************

                        12786           ;*****************************************************************
                                        ;*****************************************************************

                        %
```

```
                        .TITLE  CERHADO
                        ;*COPYRIGHT (C) 1975-1977
                        ;*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
000006                  R6=SP                   ;;STACK POINTER
000007                  R7=PC                   ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
```

```
                        ;
                        ;PRIORITY LEVEL DEFINITIONS
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  037540                              $ENDAD
            000052                          .=52
000052  000000                              000000
            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
```

E03

MEMORY MANAGEMENT DEFINITIONS SEQ 0030

```
  1416            177576                  SR2=    177576
  1417            172516                  SR3=    172516
  1418
  1419                                    ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
  1420
  1421            172300                  KIPDR0= 172300
  1422            172302                  KIPDR1= 172302
  1423            172304                  KIPDR2= 172304
  1424            172306                  KIPDR3= 172306
  1425            172310                  KIPDR4= 172310
  1426            172312                  KIPDR5= 172312
  1427            172314                  KIPDR6= 172314
  1428            172316                  KIPDR7= 172316
  1429
  1430                                    ;*KERNEL "I" PAGE ADDRESS REGISTERS
  1431
  1432            172340                  KIPAR0= 172340
  1433            172342                  KIPAR1= 172342
  1434            172344                  KIPAR2= 172344
  1435            172346                  KIPAR3= 172346
  1436            172350                  KIPAR4= 172350
  1437            172352                  KIPAR5= 172352
  1438            172354                  KIPAR6= 172354
  1439            172356                  KIPAR7= 172356
  1440
  1441                                    ;;**************************************************************
  1442            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
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $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
```

```
CERHADO MACY11 30(1046)  21-DEC-77  13:06  PAGE 33                                                    SEQ 0032
CERHAD.P11    21-DEC-77 12:48          COMMON TAGS

 1499  001224  000012                  $LF:    .ASCIZ  <12>                ;;LINE FEED
 1500                                  ;;*************************************************************
```

```
                        .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  046676                  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  065117                  DH1                     ;PC
                                                        ;RHDT
001232  066552                  DT1                     ;$ERRPC,$BDDAT,0
001234  066734                  DF1                     ;0

                        ;ITEM 2
001236  047050                  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  065117                  DH1                     ;PC
                                                        ;RHDT
001242  066552                  DT1                     ;$ERRPC,$BDDAT,0
001244  066734                  DF1                     ;0

                        ;ITEM 3
001246  047252                  EM3                     ;THE RH BASE ADDRESS WAS 772440
                                                        ;INDICATING A TM02, BUT THE
                                                        ;DRIVE TYPE DID NOT CONTAIN
                                                        ;142010 INDICATING NO
                                                        ;TM02
001250  065117                  DH1                     ;PC
                                                        ;RHDT
001252  066552                  DT1                     ;$ERRPC,$BDDAT,0
001254  066734                  DF1                     ;0,0

                        ;ITEM 4
```

```
1557   001256  047406                          EM4             ;THE RH BASE ADDRESS WAS 776300
1558                                                            ;INDICATING MORE THAN ONE
1559                                                            ;RH DEVICE. BUT THE DRIVE
1560                                                            ;TYPE DID NOT CONTAIN
1561                                                            ;0,1,2,3,4,20020,24020,20021,24021,
1562                                                            ;20022,24022, OR 142010 INDICATING NO
1563                                                            ;RH DEVICE IS PRESENT
1564   001260  065117                          DH1             ;PC
1565                                                            ;RHDT
1566   001262  066552                          DT1             ;$ERRPC,$BDDAT,0
1567   001264  066734                          DF1             ;0,0
1568
1569                                   ;ITEM 5
1570   001266  047705                          EM5             ;THE UNIBUS TIMED OUT
1571                                                            ;FOR EACH OF THE FOLLOWING
1572                                                            ;ADDRESSES, INDICATING
1573                                                            ;NO RH ON THE SYSTEM
1574                                                            ;
1575                                                            ;SO ABORT PROGRAM
1576   001270  065126                          DH5             ;PC
1577                                                            ;UNIBUS ADDRESSES ON WHICH
1578                                                            ;TIME OUT OCCURRED
1579   001272  066560                          DT5             ;$ERRPC,RSCS1,RPCS1,TMCS1,MIXCS1
1580   001274  066736                          DF5             ;0,0,0,0,0
1581
1582                                   ;ITEM 6
1583   001276  050060                          EM6             ;AFTER AN RH CLEAR (BIT #5
1584                                                            ;IN RHCS2) RHCS2 DOES NOT
1585                                                            ;HAVE ONLY IR AND UNIT
1586                                                            ;NUMBER
1587   001300  065202                          DH6             ;PC
1588                                                            ;TEST NO
1589                                                            ;RHCS2 GOOD
1590                                                            ;RHCS2 BAD
1591   001302  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1592   001304  066743                          DF6             ;0,0,0,0
1593
1594                                   ;ITEM 7
1595   001306  050176                          EM7             ;AFTER CLEARING THE RH AND
1596                                                            ;WRITING ONE WORD INTO
1597                                                            ;RHDB AND READING IT
1598                                                            ;BACK CAUSED RHDB TO
1599                                                            ;HAVE  WRONG VALUE GIVEN IN BAD RHDB
1600   001310  065245                          DH7             ;PC
1601                                                            ;TEST NO
1602                                                            ;RHDB GOOD
1603                                                            ;RHDB BAD
1604   001312  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1605   001314  066743                          DF6             ;0,0,0,0
1606
1607                                   ;ITEM 10
1608   001316  050370                          EM1C            ;AFTER CLEARING THE RH AND
1609                                                            ;WRITING ONE WORD INTO
1610                                                            ;RHDB AND READING IT
1611                                                            ;BACK, CAUSED RHCS2 TO
1612                                                            ;HAVE WRONG DATA
```

```
1613
1614  001320  065202                          DH6             ;PC
1615                                                          ;TEST NUMBER
1616                                                          ;RHCS2 GOOD
1617                                                          ;RHCS2 BAD
1618  001322  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1619  001324  066743                          DF6             ;0,0,0,0
1620
1621                                  ;ITEM 11
1622  001326  050563                          EM11            ;AFTER CLEARING THE RH AND
1623                                                          ;WRITING ONE WORD INTO
1624                                                          ;RHDB AND READING IT
1625                                                          ;BACK, CAUSED RHCS1 TO
1626                                                          ;HAVE WRONG DATA
1627                                                          ;GIVEN IN BAD RHCS1
1628  001330  065305                          DH11            ;PC
1629                                                          ;TEST NUMBER
1630                                                          ;RHCS1 GOOD
1631                                                          ;RHCS1 BAD
1632  001332  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1633  001334  066743                          DF6             ;0,0,0,0
1634
1635                                  ;ITEM 12
1636  001336  050756                          EM12            ;AFTER CLEARING THE RH AND
1637                                                          ;WRITING ONE WORD INTO
1638                                                          ;RHDB AND READING IT
1639                                                          ;BACK, CAUSED RHCS3 TO
1640                                                          ;HAVE WRONG DATA
1641                                                          ;GIVEN IN BAD RHCS3
1642  001340  065347                          DH12            ;PC
1643                                                          ;TEST NUMBER
1644                                                          ;RHCS3 GOOD
1645                                                          ;RHCS3 BAD
1646  001342  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1647  001344  066743                          DF6             ;0,0,0,0
1648
1649                                  ;ITEM 13
1650  001346  051151                          EM13            ;AFTER CLEARING THE RH AND
1651                                                          ;WRITING ONE WORD INTO
1652                                                          ;RHDB AND READING IT
1653                                                          ;BACK, CAUSED RHBA TO
1654                                                          ;HAVE WRONG DATA
1655                                                          ;GIVEN IN BAD RHBA
1656  001350  065411                          DH13            ;PC
1657                                                          ;TEST NUMBER
1658                                                          ;RHBA GOOD
1659                                                          ;RHBA BAD
1660  001352  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1661  001354  066743                          DF6             ;0,0,0,0
1662
1663                                  ;ITEM 14
1664  001356  051343                          EM14            ;AFTER CLEARING THE RH AND
1665                                                          ;WRITING ONE WORD INTO
1666                                                          ;RHDB AND READING IT
1667                                                          ;BACK, CAUSED RHBAE TO
1668                                                          ;HAVE WRONG DATA
```

```
                                                        ;GIVEN IN BAD RHBAE
001360  065451                      DH14              ;PC
                                                        ;TEST NUMBER
                                                        ;RHBAE GOOD
                                                        ;RHBAE BAD
001362  066574                      DT6               ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001364  066743                      DF6               ;0,0,0,0

                              ;ITEM 15
001366  051536                      EM15              ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHWC TO
                                                        ;HAVE WRONG DATA
                                                        ;GIVEN IN BAD RHWC
001370  065513                      DH15              ;PC
                                                        ;TEST NUMBER
                                                        ;RHWC GOOD
                                                        ;RHWC BAD
001372  066574                      DT6               ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001374  066743                      DF6               ;0,0,0,0

                              ;ITEM 16
001376  051730                      EM16              ;AFTER CLEARING THE RH AND
                                                        ;WRITING ONE WORD INTO
                                                        ;RHDB
                                                        ;CAUSED RHCS2 TO
                                                        ;HAVE WRONG DATA
                                                        ;GIVEN IN BAD RHCS2
001400  065202                      DH6               ;PC
                                                        ;TEST NUMBER
                                                        ;RHCS2 GOOD
                                                        ;RHCS2 BAD
001402  066574                      DT6               ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001404  066743                      DF6               ;0,0,0,0

                              ;ITEM 17
001406  052123                      EM17              ;AFTER CLEARING THE RH AND
                                                        ;WRITING TWO WORD INTO
                                                        ;RHDB AND READING IT
                                                        ;BACK, CAUSED RHDB TO
                                                        ;HAVE WRONG DATA
                                                        ;GIVEN IN BAD RHDB
001410  065245                      DH7               ;PC
                                                        ;TEST NUMBER
                                                        ;RHDB GOOD
                                                        ;RHDB BAD
001412  066574                      DT6               ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001414  066743                      DF6               ;0,0,0,0

                              ;ITEM 20
001416  052315                      EM20              ;AFTER CLEARING THE RH AND
                                                        ;WRITING TWO WORD INTO
                                                        ;RHDB AND READING ONE
                                                        ;BACK, CAUSED RHCS2 TO
                                                        ;HAVE WRONG DATA
```

```
1725
1726  001420  065202                        DH6             ;GIVEN IN BAD RHCS2
1727                                                        ;PC
1728                                                        ;TEST NUMBER
1729                                                        ;RHCS2 GOOD
1730  001422  066574                        DT6             ;RHCS2 BAD
1731  001424  066743                        DF6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1732                                                        ;0,0,0,0
1733                              ;ITEM 21
1734  001426  052511                        EM21            ;AFTER CLEARING THE RH AND
1735                                                        ;WRITING TWO WORD INTO
1736                                                        ;RHDB AND READING IT
1737                                                        ;BACK TWICE, CAUSED RHDB TO
1738                                                        ;HAVE WRONG DATA,
1739                                                        ;GIVEN IN BAD RHDB
1740  001430  065245                        DH7             ;PC
1741                                                        ;TEST NUMBER
1742                                                        ;RHDB GOOD
1743                                                        ;RHDB BAD
1744  001432  066574                        DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1745  001434  066743                        DF6             ;0,0,0,0
1746
1747                              ;ITEM 22
1748  001436  052711                        EM22            ;AFTER CLEARING THE RH AND
1749                                                        ;WRITING TWO WORD INTO
1750                                                        ;RHDB AND READING IT
1751                                                        ;BACK TWICE, CAUSED RHCS2 TO
1752                                                        ;HAVE WRONG DATA
1753                                                        ;GIVEN IN BAD RHCS2
1754  001440  065202                        DH6             ;PC
1755                                                        ;TEST NUMBER
1756                                                        ;RHCS2 GOOD
1757                                                        ;RHCS2 BAD
1758  001442  066574                        DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1759  001444  066743                        DF6             ;0,0,0,0
1760
1761                              ;ITEM 23
1762  001446  053112                        EM23            ;AFTER CLEARING THE RH AND
1763                                                        ;WRITING EIGHT WORD INTO
1764                                                        ;RHDB
1765                                                        ;CAUSED RHCS2 TO
1766                                                        ;HAVE WRONG DATA
1767                                                        ;GIVEN IN BAD RHCS2
1768  001450  065202                        DH6             ;PC
1769                                                        ;TEST NUMBER
1770                                                        ;RHCS2 GOOD
1771                                                        ;RHCS2 BAD
1772  001452  066574                        DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1773  001454  066743                        DF6             ;0,0,0,0
1774
1775                              ;ITEM 24
1776  001456  053307                        EM24            ;THE RH WAS CLEARED
1777                                                        ;AND A PATTERN OF 8 WORDS
1778                                                        ;WERE WRITTEN INTO RHDB
1779                                                        ;READING RHDB FOR THE
1780                                                        ;"N" TH. TIME GAVE WRONG
```

```
                                                                        ;VALUE IN RHDB
                                                                        ;N IS GIVEN IN "WORD NO"
001460  065553                          DH24                          ;PC
                                                                        ;TEST NO
                                                                        ;WORD NO
                                                                        ;RHDB GOOD
                                                                        ;RHDB BAD
001462  066606                          DT24                          ;$ERRPC,TSTNM,SILONM,$GDDAT,$BDDAT
001464  066747                          DF24                          ;0,0,0,0,0

                        ;ITEM 25
001466  053537                          EM25                          ;THE RH WAS CLEARED AND
                                                                        ;A PATTERN OF 8 WORDS WERE
                                                                        ;WRITTEN INTO RHDB
                                                                        ;AFTER READING ALL 8 WORDS
                                                                        ;FOLLWING REGISTER CONTA..IED WRONG
                                                                        ;VALUE
001470  065202                          DH6                           ;PC
                                                                        ;TEST NO
                                                                        ;RHCS2 GOOD
                                                                        ;RHCS2 BAD
001472  066574                          DT6                           ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001474  066743                          DF6                           ;0,0,0,0

                        ;ITEM 26
001476  053537                          EM25
001500  065305                          DH11
001502  066574                          DT6
001504  066743                          DF6                           ;0,0

                        ;ITEM 27
001506  053537                          EM25
001510  065347                          DH12                          ;PC
                                                                        ;TEST NO
                                                                        ;RHCS3 GOOD
                                                                        ;RHCS3 BAD
001512  066574                          DT6                           ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001514  066743                          DF6                           ;0,0,0,0

                        ;ITEM 30
001516  053537                          EM25
001520  065411                          DH13                          ;PC
                                                                        ;TEST NO
                                                                        ;RHBA GOOD
                                                                        ;RHBA BAD
001522  066574                          DT6                           ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001524  066743                          DF6                           ;0,0,0,0

                        ;ITEM 31
001526  053537                          EM25
001530  065451                          DH14                          ;PC
                                                                        ;TEST NO
                                                                        ;RHBAE GOOD
                                                                        ;RHBAE BAD
001532  066574                          DT6                           ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001534  066743                          DF6                           ;0,0,0,0
```

```
                                ;ITEM 32
001536  053537                          EM25
001540  065513                          DH15                    ;PC
                                                                ;TEST NO
                                                                ;RHWC GOOD
                                                                ;RHWC BAD
001542  066574                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001544  066743                          DF6                     ;0,0,0,0

                                ;ITEM 33
001546  053766                          EM33                    ;SETTING RH CLEAR BIT #5
                                                                ;IN RHCS2 CAUSED
                                                                ;ERROR REGISTER 1 TO HAVE
                                                                ;WRONG VALUE
001550  065623                          DH33                    ;PC
                                                                ;TEST NO
                                                                ;RHER1 GOOD
                                                                ;RHER1 BAD
001552  066574                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001554  066743                          DF6

                                ;ITEM 34
001556  054102                          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  065623                          DH33                    ;PC
                                                                ;TEST NO
                                                                ;RHER1 GOOD
                                                                ;RHER1 BAD
001562  066574                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001564  066743                          DF6                     ;0,0,0,0

                                ;ITEM 35
001566  054344                          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  065305                          DH11                    ;PC
                                                                ;TEST NO
                                                                ;RHCS1 GOOD
                                                                ;RHCS1 BAD
001572  066574                          DT6                     ;$ERRPC,TSTNM,$GDDAT,$BDDAT
001574  066743                          DF6                     ;0,0,0,0

                                ;ITEM 36
001576  054542                          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
```

```
1893                                    ;ON CHECKING RHCS1
1894                                    ;IT DID NOT CONTAIN SC,MCPE
1895                                    ;AND RDY
```

```
1896   001600  065305                          DH11            ;PC
1897                                                            ;TEST NUMBER
1898                                                            ;RHCS1 GOOD
1899                                                            ;RHCS1 BAD
1900   001602  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1901   001604  066743                          DF6             ;0,0,0,0
1902
1903                                   ;ITEM 37
1904   001606  055017                          EM37            ;RH WAS CLEARED
1905                                                            ;RHER1 WAS CHECKED
1906                                                            ;PAT (RHCS2-BIT #4) WAS SET
1907                                                            ;RHER1 WAS READ
1908                                                            ;"1" WAS WRITTEN IN "TRE" RHCS1
1909                                                            ;AN RH CLEAR WAS TO
1910                                                            ;GIVE WHAT IS IN "GOOD"
1911                                                            ;BUT GAVE WHAT IS IN "BAD
1912   001610  065305                          DH11            ;PC
1913                                                            ;TEST NO
1914                                                            ;RHCS1 GOOD
1915                                                            ;RHCS1 BAD
1916   001612  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1917   001614  066743                          DF6             ;0,0,0,0
1918
1919                                   ;ITEM 40
1920   001616  055017                          EM37
1921   001620  065202                          DH6
1922   001622  066574                          DT6
1923   001624  066743                          DF6
1924
1925                                   ;ITEM 41
1926   001626  055017                          EM37
1927   001630  065347                          DH12
1928   001632  066574                          DT6
1929   001634  066743                          DF6
1930
1931                                   ;ITEM 42
1932   001636  055017                          EM37
1933   001640  065411                          DH13
1934   001642  066574                          DT6
1935   001644  066743                          DF6
1936
1937                                   ;ITEM 43
1938   001646  055017                          EM37
1939   001650  065451                          DH14
1940   001652  066574                          DT6
1941   001654  066743                          DF6
1942
1943                                   ;ITEM 44
1944   001656  055017                          EM37
1945   001660  065513                          DH15
1946   001662  066574                          DT6
1947   001664  066743                          DF6
1948
1949                                   ;ITEM 45
1950   001666  055321                          EM45            ;AFTER AN RH CLEAR
1951                                                            ;"1" WAS WRITTEN IN THE DISK
```

```
1952
1953
1954
1955
1956
1957
1958
1959  001670  065665                          DH45                    ;ADDRESS REGISTER (IN TAPE
1960                                                                    ;DRIVES CALLED FRAME COUNT )
1961                                                                    ;PAT IN RHCS1 WAS SET
1962                                                                    ;ON READING RHDA (IN TAPE
1963  001672  066574                          DT6                     ;DRIVES RHFC ) IT DID NOT
1964  001674  066743                          DF6                     ;CONTAIN "1"
1965                                                                    ;RHDA=RHFC
1966                                ;ITEM 46                            ;PC
1967  001676  055557                          EM46                    ;TEST NO
1968                                                                    ;RHDA GOOD
1969                                                                    ;RHDA BAD
1970                                                                    ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1971                                                                    ;0,0,0,0
1972
1973
1974
1975
1976  001700  065305                          DH11                    ;AFTER AN RH CLEAR
1977                                                                    ;"1" WAS WRITTEN IN THE
1978                                                                    ;DISK ADDRESS REGISTER (IN
1979                                                                    ;TAPE
1980  001702  066574                          DT6                     ;PAT IN RHCS1 WAS SET
1981  001704  066743                          DF6                     ;ON READING RHDA (IN TAPE
1982                                                                    ;  ) FOLLOWING
1983                                ;ITEM 47                            ;REGISTER DID NOT CONTAIN
1984  001706  055557                          EM46                    ;WHAT IS IN "GOOD"
1985  001710  065202                          DH6                     ;PC
1986  001712  066574                          DT6                     ;TEST NO
1987  001714  066743                          DF6                     ;RHCS1 GOOD
1988                                                                    ;RHCS1 BAD
1989                                ;ITEM 50                            ;$ERRPC,TSTNM,$GDDAT,$BDDAT
1990  001716  055557                          EM46
1991  001720  065347                          DH12
1992  001722  066574                          DT6
1993  001724  066743                          DF6
1994
1995                                ;ITEM 51
1996  001726  055557                          EM46
1997  001730  065411                          DH13
1998  001732  066574                          DT6
1999  001734  066743                          DF6
2000
2001                                ;ITEM 52
2002  001736  055557                          EM46
2003  001740  065451                          DH14
2004  001742  066574                          DT6
2005  001744  066743                          DF6
2006
2007                                ;ITEM 53
```

```
 2008  001746  055557                        EM46
 2009  001750  065513                        DH15
 2010  001752  066574                        DT6
 2011  001754  066743                        DF6
 2012
 2013                                ;ITEM 54
 2014  001756  056060                        EM54          ;AN RH CLEAR (RHCS2 BIT #5)
 2015                                                      ;WAS GIVEN
 2016                                                      ;A16 (RHCS1 BIT #8) WAS SET
 2017                                                      ;ON READING THE FOLLOWING
 2018                                                      ;REGISTER IT DID NOT
 2019                                                      ;CONTAIN WHAT IS IN "GOOD"
 2020  001760  065305                        DH11
 2021  001762  066574                        DT6
 2022  001764  066743                        DF6
 2023
 2024                                ;ITEM 55
 2025  001766  056060                        EM54
 2026  001770  065451                        DH14
 2027  001772  066574                        DT6
 2028  001774  066743                        DF6
 2029
 2030                                ;ITEM 56
 2031  001776  056271                        EM56          ;AN RH CLEAR (RHCS2 BIT #5)
 2032                                                      ;WAS GIVEN
 2033                                                      ;A16 WAS WRITTEN IN RHCS1
 2034                                                      ;ALL ZEROS WERE WRITTEN IN RHBAE
 2035                                                      ;THE FOLLOWING REGISTER DID
 2036                                                      ;NOT CONTAIN WHAT IS IN "GOOD"
 2037  002000  065451                        DH14
 2038  002002  066574                        DT6
 2039  002004  066743                        DF6
 2040
 2041                                ;ITEM 57
 2042  002006  056271                        EM56
 2043  002010  065305                        DH11
 2044  002012  066574                        DT6
 2045  002014  066743                        DF6
 2046
 2047                                ;ITEM 60
 2048  002016  056527                        EM60          ;AN RH CLEAR (RHCS2 BIT #5)
 2049                                                      ;WAS GIVEN
 2050                                                      ;A17 (RHCS1 BIT #9) WAS SET
 2051                                                      ;ON READING THE FOLLOWING
 2052                                                      ;REGISTER IT DID NOT
 2053                                                      ;CONTAIN WHAT IS IN "GOOD"
 2054  002020  065305                        DH11
 2055  002022  066574                        DT6
 2056  002024  066743                        DF6
 2057
 2058                                ;ITEM 61
 2059  002026  056527                        EM60
 2060  002030  065451                        DH14
 2061  002032  066574                        DT6
 2062  002034  066743                        DF6
 2063
```

```
                                ;ITEM 62
002036  056740                          EM62            ;AN RH CLEAR (RHCS2 BIT #5)
                                                        ;WAS GIVEN
                                                        ;A17 WAS WRITTEN IN RHCS1
                                                        ;ALL ZEROS WERE WRITTEN IN RHBAE
                                                        ;THE FOLLOWING REGISTER DID
                                                        ;NOT CONTAIN WHAT IS IN "GOOD"
002040  065451                          DH14
002042  066574                          DT6
002044  066743                          DF6

                                ;ITEM 63
002046  056740                          EM62
002050  065305                          DH11
002052  066574                          DT6
002054  066743                          DF6

                                ;ITEM 64
002056  057175                          EM64            ;AN RH CLEAR (RHCS2 BIT #5)
                                                        ;WAS GIVEN
                                                        ;IE (RHCS1 BIT #6) WAS SET
                                                        ;ON READING THE FOLLOWING
                                                        ;REGISTER IT DID NOT
                                                        ;CONTAIN WHAT IS IN "GOOD"
002060  065305                          DH11
002062  066574                          DT6
002064  066743                          DF6

                                ;ITEM 65
002066  057175                          EM64
002070  065347                          DH12
002072  066574                          DT6
002074  066743                          DF6

                                ;ITEM 66
002076  057175                          EM64            ;AN RH CLEAR (RHCS2 BIT #5)
                                                        ;WAS GIVEN
                                                        ;A16 WAS WRITTEN IN RHCS1
                                                        ;ALL ZEROS WERE WRITTEN IN RHBAE
                                                        ;THE FOLLOWING REGISTER DID
                                                        ;NOT CONTAIN WHAT IS IN "GOOD"
002100  065347                          DH12
002102  066574                          DT6
002104  066743                          DF6

                                ;ITEM 67
002106  057175                          EM64
002110  065305                          DH11
002112  066574                          DT6
002114  066743                          DF6

                                ;ITEM 70
002116  057644                          EM70            ;RH CLEAR WAS GIVEN
                                                        ;TWO SUCCESSIVE "GO" (RHCS1 BIT #0)
                                                        ;WAS GIVEN WITHOUT GIVING
                                                        ;TIME FOR FIRST "GO" TO
```

```
2120
2121
2122
2123
2124   002120   065305                          DH11
2125   002122   066574                          DT6
2126   002124   066743                          DF6
2127
2128                                    ;ITEM 71
2129   002126   057644                          EM70
2130   002130   065202                          DH6
2131   002132   066574                          DT6
2132   002134   066743                          DF6
2133
2134                                    ;ITEM 72
2135   002136   057644                          EM70
2136   002140   065347                          DH12
2137   002142   066574                          DT6
2138   002144   066743                          DF6
2139
2140                                    ;ITEM 73
2141   002146   057644                          EM70
2142   002150   065411                          DH13
2143   002152   066574                          DT6
2144   002154   066743                          DF6
2145
2146                                    ;ITEM 74
2147   002156   057644                          EM70
2148   002160   065451                          DH14
2149   002162   066574                          DT6
2150   002164   066743                          DF6
2151
2152                                    ;ITEM 75
2153   002166   057644                          EM70
2154   002170   065513                          DH15
2155   002172   066574                          DT6
2156   002174   066743                          DF6
2157
2158                                    ;ITEM 76
2159   002176   060120                          EM76
2160
2161
2162
2163
2164
2165
2166
2167   002200   065305                          DH11
2168   002202   066574                          DT6
2169   002204   066743                          DF6
2170
2171                                    ;ITEM 77
2172   002206   060120                          EM76
2173   002210   065202                          DH6
2174   002212   066574                          DT6
2175   002214   066743                          DF6
```

```
                                        ;COMPLETE
                                        ;THE FOLLOWING REGISTER
                                        ;DID NOT CONTAIN WHAT IS
                                        ;IN "GOOD"
                                        ;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
```

```
2176
2177                                    ;ITEM 100
2178  002216  060120                           EM76
2179  002220  065347                           DH12
2180  002222  066574                           DT6
2181  002224  066743                           DF6
2182
2183                                    ;ITEM 101
2184  002226  060120                           EM76
2185  002230  065411                           DH13
2186  002232  066574                           DT6
2187  002234  066743                           DF6
2188
2189                                    ;ITEM 102
2190  002236  060120                           EM76
2191  002240  065451                           DH14
2192  002242  066574                           DT6
2193  002244  066743                           DF6
2194
2195                                    ;ITEM 103
2196  002246  060120                           EM76
2197  002250  065513                           DH15
2198  002252  066574                           DT6
2199  002254  066743                           DF6
2200
2201                                    ;ITEM 104
2202  002256  060413                           EM104                ;RH CLEAR WAS GIVEN AN
2203                                                                 ;IPCK BIT SHOWN IN "IPCK" WAS SET
2204                                                                 ;ZEROS WERE MOVED INTO RHDB
2205                                                                 ;ON READING RHDB
2206                                                                 ;THE FOLLOWING REGISTER
2207                                                                 ;DID NOT CONTAIN WHAT
2208                                                                 ;IS IN GOOD
2209  002260  065725                           DH104                ;PC,TEST NO,IPCK,RHCS3 GOOD,RHCS3 BAD
2210  002262  066622                           DT104                ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2211  002264  066754                           DF104                ;0,0,0,0,0
2212
2213                                    ;ITEM 105
2214  002266  060413                           EM104
2215  002270  065775                           DH105                ;PC,TST NO, IPCK,RHCS2 GOOD,RHCS2 BAD
2216  002272  066636                           DT105                ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2217  002274  066761                           DF105                ;0,0,0,0,0
2218
2219                                    ;ITEM 106
2220  002276  060651                           EM106                ;RH CLEAR WAS GIVEN AN
2221                                                                 ;IPCK BIT SHOWN IN "IPCK" WAS SET
2222                                                                 ;ZEROS WERE MOVED INTO
2223                                                                 ;RHDB FROM AN ODD WORD
2224                                                                 ;RHDB WAS READ
2225                                                                 ;AN RH CLEAR WAS GIVEN
2226                                                                 ;TO CLEAR ALL ERRORS
2227                                                                 ;THE FOLLLOWING REGISTER
2228                                                                 ;DID NOT CONTAIN WHAT
2229                                                                 ;IS IN GOOD
2230  002300  066045                           DH106                ;PC,TSTNM,IPCK,RHCS1 GOOD,RHCS1 BAD
2231  002302  066652                           DT106                ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
```

```
2232  002304  066766                          DF106                   ;0,0,0,0,0
2233
2234                                    ;ITEM 107
2235  002306  060651                          EM106
2236  002310  065775                          DH105                   ;PC,TSTNM,IPCK,RHCS2 GOOD,RHCS2 BAD
2237  002312  066636                          DT105
2238  002314  066761                          DF105
2239
2240                                    ;ITEM 110
2241  002316  060651                          EM106
2242  002320  065725                          DH104
2243  002322  066622                          DT104
2244  002324  066754                          DF104
2245
2246                                    ;ITEM 111
2247  002326  060651                          EM106
2248  002330  066115                          DH111                   ;PC,TSTNM,IPCK,RHBA GOOD,RHBA BAD
2249  002332  066622                          DT104                   ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2250  002334  066754                          DF104                   ;0,0,0,0,0
2251
2252                                    ;ITEM 112
2253  002336  060651                          EM106
2254  002340  066163                          DH112                   ;PC,TSTNM,IPCK,RHBAE GOOD,RHBAE BAD
2255  002342  066622                          DT104                   ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2256  002344  066754                          DF104                   ;0,0,0,0,0
2257
2258                                    ;ITEM 113
2259  002346  060651                          EM106
2260  002350  066233                          DH113                   ;PC,TSTNM,IPCK,RHWC GOOD,RHWC BAD
2261  002352  066622                          DT104                   ;$ERRPC,TSTNM,IP,$GDDAT,$BDDAT
2262  002354  066754                          DF104                   ;0,0,0,0,0
2263
2264                                    ;ITEM 114
2265  002356  061142                          EM114                   ;RH CLEAR WAS GIVEN AN
2266                                                                  ;A WRITE CHECK WAS DONE
2267                                                                  ;THE FOLLOWING REGISTER
2268                                                                  ;DID NOT CONTAIN WHAT IS
2269                                                                  ;IN "GOOD"
2270  002360  065202                          DH6
2271  002362  066574                          DT6
2272  002364  066743                          DF6
2273
2274                                    ;ITEM 115
2275  002366  061142                          EM114                   ;RH CLEAR WAS GIVEN AN
2276                                                                  ;A WRITE CHECK WAS DONE
2277                                                                  ;THE FOLLOWING REGISTER
2278                                                                  ;DID NOT CONTAIN WHAT IS
2279                                                                  ;IN "GOOD"
2280  002370  065347                          DH12
2281  002372  066574                          DT6
2282  002374  066743                          DF6
2283
2284                                    ;ITEM 116
2285  002376  061311                          EM116                   ;RH CLEAR WAS GIVEN AN
2286                                                                  ;A WRITE CHECK WAS DONE
2287                                                                  ;THEN ANOTHER RH CLEAR WAS
```

```
                                                                        ;GIVEN TO CLEAR ALL ERRORS
                                                                        ;THE FOLLOWING REGISTER DID
                                                                        ;NOT CONTAIN WHAT IS IN "GOOD"
002400  065305                                  DH11
002402  066574                                  DT6
002404  066743                                  DF6

                                        ;ITEM 117
002406  061311                                  EM116
002410  065202                                  DH6
002412  066574                                  DT6
002414  066743                                  DF6

                                        ;ITEM 120
002416  061311                                  EM116
002420  065347                                  DH12
002422  066574                                  DT6
002424  066743                                  DF6

                                        ;ITEM 121
002426  061311                                  EM116
002430  065411                                  DH13
002432  066574                                  DT6
002434  066743                                  DF6

                                        ;ITEM 122
002436  061311                                  EM116
002440  065451                                  DH14
002442  066574                                  DT6
002444  066743                                  DF6

                                        ;ITEM 123
002446  061311                                  EM116
002450  065513                                  DH15
002452  066574                                  DT6
002454  066743                                  DF6

                                        ;ITEM 124
002456  061521                                  EM124                 ;ON A SILO TEST TO
                                                                      ;TEST DBL RHCS3-BIT #10
                                                                      ;THE FOLLOWING REGISTER
                                                                      ;DID NOT CONTAIN WHAT
                                                                      ;IS IN "GOOD"
002460  065347                                  DH12
002462  066574                                  DT6
002464  066743                                  DF6

                                        ;ITEM 125
002466  061521                                  EM124
002470  065305                                  DH11
002472  066574                                  DT6
002474  066743                                  DF6

                                        ;ITEM 126
002476  061521                                  EM124
002500  065202                                  DH6
```

```
 2344  002502  066574                        DT6
 2345  002504  066743                        DF6
 2346
 2347                                ;ITEM 127
 2348  002506  061521                        EM124
 2349  002510  065411                        DH13
 2350  002512  066574                        DT6
 2351  002514  066743                        DF6
 2352
 2353                                ;ITEM 130
 2354  002516  061521                        EM124
 2355  002520  065451                        DH14
 2356  002522  066574                        DT6
 2357  002524  066743                        DF6
 2358
 2359                                ;ITEM 131
 2360  002526  061521                        EM124
 2361  002530  065513                        DH15
 2362  002532  066574                        DT6
 2363  002534  066743                        DF6
 2364                                ;ITEM132
 2365  002536  061664                        EM132          ;BEFORE DATA TRANSFER
 2366                                                       ;COMMAND WAS TO BE GIVEN
 2367                                                       ;THE RP DRIVE
 2368                                                       ;STATUS REGISTER DID NOT
 2369                                                       ;CONTAIN WHAT IS IN GOOD
 2370
 2371  002540  066301                        DH132          ;PC
 2372                                                       ;TEST NO.
 2373                                                       ;RHDS1 GOOD
 2374                                                       ;RHDS1 BAD
 2375  002542  066574                        DT6
 2376  002544  066743                        DF6
 2377                                ;ITEM133
 2378  002546  062042                        EM133          ;BEFORE DATA TRANSFER
 2379                                                       ;COMMAND WAS TO BE GIVEN
 2380                                                       ;THE RS DRIVE
 2381                                                       ;STATUS REGISTER DID NOT
 2382                                                       ;CONTAIN WHAT IS IN GOOD
 2383
 2384  002550  066301                        DH132          ;PC
 2385                                                       ;TEST NO.
 2386                                                       ;RHDS1 GOOD
 2387                                                       ;RHDS1 BAD
 2388  002552  066574                        DT6
 2389  002554  066743                        DF6
 2390                                ;ITEM 134
 2391  002556  062220                        EM134          ;BEFORE DATA TRANSFER COMMAND
 2392                                                       ;WAS TO BE GIVEN THE
 2393                                                       ;MAG TAPE DRIVE STATUS
 2394                                                       ;REGISTER DID NOT CONTAIN WHAT
 2395                                                       ;IS IN GOOD
 2396
 2397  002560  066301                        DH132
 2398  002562  066574                        DT6
 2399  002564  066743                        DF6
```

```
2400                                    ;ITEM135
2401  002566  062410                            EM135           ;WAS WATIING FOR A BIT TO SET
2402                                                            ;BIT IN QUESTION IS IN "BIT WAITED FOR"
2403                                                            ;REGISTER IN QUESTION IN "REG ADDR"
2404  002570  066343                            DH135           ;PC
2405                                                            ;TEST NO
2406                                                            ;PC OF WAT
2407                                                            ;BIT
2408                                                            ;REG ADDR
2409  002572  066666                            DT135           ;$ERRPC,TSTNM,WATIPC,WAITBT,WAITRE
2410  002574  066773                            DF135           ;0,0,0,0,0
2411                                    ;ITEM136
2412  002576  062610                            EM136           ;WAS WATIING FOR A BIT TO RESET
2413                                                            ;BIT IN QUESTION IS IN "BIT WAITED FOR"
2414                                                            ;REGISTER IN QUESTION IN "REG ADDR"
2415  002600  066343                            DH135           ;PC
2416                                                            ;TEST NO
2417                                                            ;PC OF WAT
2418                                                            ;BIT
2419                                                            ;REG ADDR
2420  002602  066666                            DT135           ;$ERRPC,TSTNM,WATIPC,WAITBT,WAITRE
2421  002604  066773                            DF135           ;0,0,0,0,0
2422                                    ;ITEM 137
2423  002606  063012                            EM137           ;ON WRITING AND READING
2424                                                            ;THE REGISTER # IN "REG. ADDR"
2425                                                            ;IT DID NOT CONTAIN
2426                                                            ;EXPECTED VALUE.
2427  002610  066423                            DH137           ;PC
2428                                                            ;TEST NO
2429                                                            ;REG ADDR
2430                                                            ;GOOD DATA
2431                                                            ;BAD DATA
2432  002612  066702                            DT137           ;$ERRPC,TSTNM,$BDADR,$GDDAT,$BDDAT
2433  002614  067000                            DF137           ;0,0,0,0,0
2434
2435
2436                                    ;ITEM 140
2437  002616  063140                            EM140
2438  002620  066045                            DH106
2439  002622  066652                            DT106
2440  002624  066766                            DF106
2441                                    ;ITEM 141
2442  002626  063372                            EM141           ;AFTER CLEARING THE RH AND
2443                                                            ;WRITING TWO WORD INTO
2444                                                            ;RHDB AND READING IT
2445                                                            ;BACK TWICE, CAUSED RHCS1 TO
2446                                                            ;HAVE WRONG DATA,
2447                                                            ;GIVEN IN BAD RHCS1
2448  002630  065305                            DH11            ;PC
2449                                                            ;TEST NUMBER
2450                                                            ;RHCS1 GOOD
2451                                                            ;RHCS1 BAD
2452  002632  066574                            DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2453  002634  066743                            DF6             ;0,0,0,0
2454
2455
```

```
2456                                    ;ITEM 142
2457  002636  063573                          EM142           ;AFTER CLEARING THE RH AND
2458                                                          ;WRITING TWO WORD INTO
2459                                                          ;RHDB AND READING IT
2460                                                          ;BACK TWICE, CAUSED RHCS3 TO
2461                                                          ;HAVE WRONG DATA,
2462                                                          ;GIVEN IN BAD RHCS3
2463  002640  065347                          DH12            ;PC
2464                                                          ;TEST NUMBER
2465                                                          ;RHCS3 GOOD
2466                                                          ;RHCS3 BAD
2467  002642  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2468  002644  066743                          DF6             ;0,0,0,0
2469
2470
2471                                    ;ITEM 143
2472  002646  063774                          EM143           ;AFTER CLEARING THE RH AND
2473                                                          ;WRITING TWO WORD INTO
2474                                                          ;RHDB AND READING IT
2475                                                          ;BACK TWICE, CAUSED RHBA TO
2476                                                          ;HAVE WRONG DATA,
2477                                                          ;GIVEN IN BAD RHBA
2478  002650  065411                          DH13            ;PC
2479                                                          ;TEST NUMBER
2480                                                          ;RHBA GOOD
2481                                                          ;RHBA BAD
2482  002652  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2483  002654  066743                          DF6             ;0,0,0,0
2484
2485
2486                                    ;ITEM 144
2487  002656  064174                          EM144           ;AFTER CLEARING THE RH AND
2488                                                          ;WRITING TWO WORD INTO
2489                                                          ;RHDB AND READING IT
2490                                                          ;BACK TWICE, CAUSED RHBAE TO
2491                                                          ;HAVE WRONG DATA,
2492                                                          ;GIVEN IN BAD RHBAE
2493  002660  065451                          DH14            ;PC
2494                                                          ;TEST NUMBER
2495                                                          ;RHBAE GOOD
2496                                                          ;RHBAE BAD
2497  002662  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2498  002664  066743                          DF6             ;0,0,0,0
2499
2500
2501                                    ;ITEM 145
2502  002666  064375                          EM145           ;AFTER CLEARING THE RH AND
2503                                                          ;WRITING TWO WORD INTO
2504                                                          ;RHDB AND READING IT
2505                                                          ;BACK TWICE, CAUSED RHWC TO
2506                                                          ;HAVE WRONG DATA,
2507                                                          ;GIVEN IN BAD RHWC
2508  002670  065513                          DH15            ;PC
2509                                                          ;TEST NUMBER
2510                                                          ;RHWC GOOD
2511                                                          ;RHWC BAD
```

```
2512  002672  066574                          DT6             ;$ERRPC,TSTNM,$GDDAT,$BDDAT
2513  002674  066743                          DF6             ;0,0,0,0
2514                                  ;ITEM146
2515  002676  064575                          EM146           ;A DEVICE BASE ADDRESS DID NOT
2516                                                          ;TIME OUT BUT CORRESPONDING
2517                                                          ;VECTOR ADDRESS DID TIME OUT
2518  002700  066475                          DH146           ;PC
2519                                                          ;BASE
2520                                                          ;VECTOR
2521  002702  066716                          DT146           ;$ERRPC,TESTDV,TESTVC,0
2522  002704  067005                          DF146           ;0,0,0
2523
2524                                  ;ITEM147
2525  002706  064767                          EM147           ;A DEVICE ADDRESS DID NOT
2526                                                          ;TIME OUT BUT NO UNITS HAD
2527                                                          ;APPROPRIATE DRIVE TYPE
2528  002710  066524                          DH147           ;PC
2529                                                          ;BASE ADDRESS
2530  002712  066726                          DT147           ;$ERRPC,TESTDV
2531  002714  067010                          DF147
2532
```

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


                                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)
```

ERROR POINTER TABLE


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

                        ;MAINTAINABILITY REGISTER (RHMR)(#03)

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

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

                        AT0=    1               ;DEVICE 0 (BIT #0)
                        AT1=    2               ;DEVICE 1 (BIT #1)
                        AT2=    4               ;DEVICE 2 (BIT #2)
                        AT3=    10              ;DEVICE 3 (BIT #3)
                        AT4=    20              ;DEVICE 4 (BIT #4)
                        AT5=    40              ;DEVICE 5 (BIT #5)
                        AT6=    100             ;DEVICE 6 (BIT #6)
                        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)

                        EXT1=   1               ;EXTENSION 1 (BIT #0)
                        EXT2=   2               ;EXTENSION 2 (BIT #1)
                        EXT4=   4               ;EXTENSION 3 (BIT #2)
                        EXT10=  10              ;EXTENSION 4 (BIT #3)
                        EXT20=  20              ;EXTENSION 5 (BIT #4)
                        EXT40=  40              ;EXTENSION 6 (BIT #5)
                        SC1=    100             ;SECTOR COUNT FIELD 0 (BIT #6)
                        SC2=    200             ;SECTOR COUNT FIELD 1 (BIT #7)
                        SC4=    400             ;SECTOR COUNT FIELD 2 (BIT #8)
                        SC10=   1000            ;SECTOR COUNT FIELD 3 (BIT #9)
                        SC20=   2000            ;SECTOR COUNT FIELD 4 (BIT #10)
                        TRK1=   4000            ;TRACK FIELD 1 (BIT #11)
                        TRK2=   10000           ;TRACK FIELD 2 (BIT #12)
                        TRK4=   20000           ;TRACK FIELD 3 (BIT #13)
                        TRK10=  40000           ;TRACK FIELD 4 (BIT #14)
                        TRK20=  100000          ;TRACK FIELD 5 (BIT #15)

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

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

```
2701            000100          TUF=    100             ;TRANSITIONS UNSAFE (BIT #6)
2702            000200          FEN=    200             ;FAILSAFE ENABLED (BIT #7)
2703            000400          WRU=    400             ;WRITE READY UNSAFE (BIT #8)
2704            001000          MHS=    1000            ;MULTIPLE HEAD SELECT (BIT #9)
2705            002000          NHS=    2000            ;NO HEAD SELECTION (BIT #10)
2706            004000          IXE=    4000            ;INDEX ERROR (BIT #11)
2707            010000          VU30=   10000           ;30VOLT UNSAFE (BIT #12)
2708            020000          PLU=    20000           ;PLO UNSAFE (BIT #13)
2709            100000          ACU=    100000          ;ACUNSAFE (BIT #15)
2710
2711                            ;OFFSET REGISTER (RHOF) (#11)
2712
2713            000001          OF25=   1               ;OFFSET 25 MICRO INCHES (BIT #0)
2714            000002          OF50=   2               ;OFFSET 50 MICRO INCHES (BIT #1)
2715            000004          OF100=  4               ;OFFSET 100 MICRO INCHES (BIT #2)
2716            000010          OF200=  10              ;OFFSET 200 MICRO INCHES .BIT #3)
2717            000020          OF400=  20              ;OFFSET 400 MICRO INCHES (BIT #4)
2718            000040          OF800=  40              ;OFFSET 800 MICRO INCHES (BIT #5)
2719
2720            000200          OFREV=  200             ;OFFSET NEGATIVE (REVERSE) (BIT #5)
2721            002000          HCI=    2000            ;HEADER COMPARE INHIBIT (BIT #10)
2722            004000          ECI=    4000            ;ERROR CORRECTION CODE INHIBIT (BIT #11)
2723            010000          FMT22=  10000           ;FORMAT BIT (BIT #12
2724
2725                            ;TAPE CONTROL REGISTER (RHTC)
2726
2727            000001          S1=     1               ;SLAVE NUMBER
2728            000002          S2=     2               ;SLAVE NUMBER
2729            000004          S4=     4               ;SLAVE NUMBER
2730            000010          EPAR=   10              ;EVEN PARITY
2731            000020          FMT1=   20              ;FORMAT
2732            000040          FMT2=   40              ;FORMAT
2733            000100          FMT4=   100             ;FORMAT
2734            000200          FMT8=   200             ;FORMAT
2735            000400          DEN1=   400             ;DENSIT
2736            001000          DEN2=   1000            ;DENSITY
2737            002000          DEN4=   2000            ;DENSITY
2738            001400          BPI8=   1400            ;800 BPI
2739            000300          NML=    300             ;NORMAL - 2 FRAMES PER WORD
2740
2741
2742
2743
2744
2745                            ;DRIVE TYPE REGISTER
2746
2747            002000          SLVPR=  2000            ;SLAVE PRESENT (BIT #10)
2748            010000          CH7=    10000           ;CHANNEL 7 (BIT #12)
2749
2750                            ;DESIRED CYLINDER ADDRESS (RHCA) (#12)
2751                            ;EACH BIT IS CALLED BY BIT NUMBER.
2752
2753
2754
2755
2756                            ;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/L..ITE (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



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

```
2795
2796
2797
2798            000000                  CS1=    0                       ;CONTROL STATUS 1
2799            000002                  WC=     2                       ;WORD COUNT
2800            000004                  BA=     4                       ;BUS ADDRESS
2801            000006                  FC=     6                       ;FRAME COUNT
2802            000006                  DA=     6                       ;DESIRED SECTOR/TRACK
2803            000010                  CS2=    10                      ;CONTROL STATUS 2
2804            000012                  DS=     12                      ;DRIVE STATUS
2805            000014                  ER1=    14                      ;ERROR 1
2806            000016                  AS=     16                      ;ATTENTION SUMMERY
2807            000020                  LA=     20                      ;LOOK AHEAD
2808            000022                  DB=     22                      ;DATA BUFFER
2809            000024                  MR=     24                      ;MAINTENANCE REGISTER
2810            000026                  DT=     26                      ;DRIVE TYPE
2811            000030                  SN=     30                      ;SERIAL NUMBER
2812            000032                  OF=     32                      ;OFFSET
2813            000032                  TC=     32                      ;TAPE CONTROL
2814            000034                  DC=     34                      ;DESIRED CYLINDER
2815            000036                  CC=     36                      ;CURRENT CYLINDER
2816            000040                  ER2=    40                      ;ERROR 2
2817            000042                  ER3=    42                      ;ERROR 3
2818            000044                  EC1=    44                      ;ECC 1
2819            000046                  EC2=    46                      ;ECC 2
2820
2821
2822                                    ;BUFFERS
2823            003000                  .=3000
2824   003000                           BUFEW:
2825   003000   000000                  BFEVEN: 0
2826   003002                           BUFOW:
2827   003002   000422                  BFODD:  .BLKW   274.             ;BUFFER
2828                                    ;STARTING ADDRESS OF REGISTERS
2829
2830   004046   172040                  RSCS1:  172040                  ;RS ALONG
2831   004050   176700                  RPCS1:  176700                  ;RP ALONE
2832   004052   172440                  TMCS1:  172440                  ;TM ALONE
2833   004054   176300                  MIXCS1: 176300                  ;MIXED SYSTEM
2834   004056   000000                  PRESENT:        0
2835
2836
2837
2838                                    .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
```

```
2891   004136   000002                    UNLOAD: 2                       ;UNLOAD (STAND BY)
2892   004140   000006                    RECALI: 6                       ;RECALIBRATE
2893   004142   000010                    DCLEAR: 10                      ;DRIVE CLEAR
2894   004144   000012                    RELEAS: 12                      ;RELEASE (DUAL-PORT OPERATION)
2895   004146   000030                    SERCH:  30                      ;SEARCH COMMAND
2896   004150   000050                    WRCHEK: 50                      ;WRITE CHECK DATA
2897   004152   000052                    WRCHDT: 52                      ;WRITE CHECK HEADER AND DATA
2898   004154   000060                    WRIDAT: 60                      ;WRITE DATA
2899   004156   000062                    WRIFOR: 62                      ;WRITE HEADER AND DATA (FORMAT)
2900   004160   000070                    READAT: 70                      ;READ DATA
2901   004162   000072                    REFOR:  72                      ;READ HEADER AND DATA
2902   004164   000004                    SEECOM: 4                       ;SEEK COMMAND
2903   004166   000014                    OFSETC: 14                      ;OFFSET COMMAND
2904   004170   000016                    RETCL:  16                      ;RETURNTO CENTERLINE
2905   004172   000022                    PKACK:  22                      ;PACK ACKNOWLEDGE
2906   004174   000020                    READIN: 20                      ;READ IN
2907   004176   000006                    REWIND: 6                       ;REWIND
2908   004200   000076                    REVRED: 76                      ;REVERSE READ
2909   004202   000030                    SPACFD: 30                      ;SPACE FORWARD
2910   004204   000066                    REVWRT: 66              ;REVERSE WRITE
2911   004206   000000                    ILLEGL: .WORD   0               ;COMPUTED ILLEGAL FUNCTION
2912
2913                                      ;DATA BUFFER FOR READ WRITE
2914   004210   000106                    WRFROM: .BLKW   70.             ;WRITE FROM THIS BUFFER
2915   004424   000106                    REINTO: .BLKW   70.             ;READ INTO THIS BUFFER
2916   004640   000000                    COMND:  0               ;STORE COMMAND FOR COMMAND ROUTINE
2917   004642   000000                    COMAND: 0               ;STORE COMMACD
2918   004644   000000                    NOGO:   0                       ;IF ZERO GO IS TO BE GIVEN IN COMMAND ROUTINE
2919                                                                      ;IF ONES GO IS NOT TO BE GIVEN IN COMMAND ROUTINE
2920                                                                      ;USED IN PGE TEST
2921   004646   000000                    WATO:   0                       ;IF ZERO WAIT FOR BIT TO SET
2922                                                                      ;IF ONES WAIT FOR BIT TO RESET
2923                                                                      ;USED IN WAIT TRAP
2924   004650   000000                    WRTBIT: 0                       ;BITS NOT WRITTEN INTO
2925   004652   000000                    SETBIT: 0                       ;BITS ALWAYS SET
2926   004654   000000                    CLRBIT: 0                       ;BITS ALWAYS CLEARED
2927
2928
2929
2930                                      ;RESERVED LOCATIONS
2931   004656   000000                    TSTNM:  0                       ;TEST NUMBER
2932   004660   000000                    SLAVE:  0                       ;KEEP SLAVE NO.
2933   004662   000000                    IP:     0               ;IPCK NUMBER FOR ERROR PRINTOUT
2934   004664   000000                    SILONM: 0               ;SILO WORD NUMBER FOR ERROR PRINT OUT
2935   004666   000000                    WAITPC: 0               ;WAIT TRAP PC
2936   004670   000000                    WAITRE: 0                       ;WAITING FOR REGISTER ADDRESS IN WAIT TRAP
2937   004672   000000                    WAITBT: 0                       ;WAITING FOR BIT IN WAIT TRAP
2938
2939                                      ;TABLE FOR ATTENTION BITS
2940                                      ;ATTENTION TABLE
2941   004674     001     002     004     ATABLE: .BYTE   1,2,4,10,20,40,100,200
2942   004677     010     020     040
2943   004702     100     200
2944
2945
2946
```

```
2947                                    ;RESERVED LOCATIONS FOR UNIT SELECT
2948  004704  172040                    BASEAD: 172040                  ;RS BASE ADDRESS
2949  004706  176700                    BASPAD: 176700                  ;RP BASE ADDRESS
2950  004710  172440                    BASTAD: 172440                  ;TU BASE ADDRESS
2951  004712  176300                    BASEMAD:        176300                  ;MIXED SYSTEM BASE ADDRESS
2952
2953
2954  004714  000000                    BASEVC: 0                       ;RS VECTOR
2955  004716  000254                    BASP:   254                             ;RP VECTOR
2956  004720  000001                    BAST:   1                       ;TU VECTOR
2957  004722  000002                    BASM:   2                       ;MIXED VECTOR
2958
2959                                    ;TABLE FOR GIVEN BASE ADDRESSES
2960  004724  000010                    BSGIVA: .BLKW   8.
2961
2962
2963                                    ;TABLE FOR GIVEN VECTOR
2964  004744  000010                    BSGIVV: .BLKW   8.
2965
2966
2967  004764  000004                    NMRHS:  4                       ;NUMBER OF RH
2968  004766  000000                    BASINX: 0                       ;INDEX FOR BASE
2969  004770  000000                    WORKBS: 0                       ;WORKING BASE
2970  004772  000000                    WORKNM: 0                       ;WORKING NUMBER FOR RH
2971  004774  000000                    WORKVC: 0                       ;WORKING VECTOR FOR RH
2972  004776  000000                    FORBAE: 0                       ;TOTAL NO. OF REG. FOR BAE CALCULATION
2973  005000  000000                    LOPCT:  0                       ;LOOP COUNT FOR 200 START
2974  005002  000000                    LOPCT1: 0                       ;LOOP COUNT FOR 210 START
2975  005004  000000                    USEVEC: 0                       ;USE VECTOR
2976
2977
2978
2979  005006  000000                    GIVE:   0                       ;IF ONES OPERATOR WILL GIVE ADDRESSS
2980  005010  000000                    UNIT:   0                               ;UNIT UNDER TEST
2981  005012  000000                    ST200:  0                       ;ALL ONES INDICATE STARTING FROM 200
2982                                    ;TESTING TABLE
2983  005014  000010                    TSRHNO: .BLKW   8.              ;RH NO TO BE TESTED IN ORDER OF TEST
2984
2985  005034  000010                    TSDEVICE:       .BLKW   8.      ;DEVICE TO BE TESTED IN ORDER OF TEST
2986                                                                    ;1=RS03,3/L,3/LA  2=RS04,4/L  4=RP04,5,6 SINGLE PORT
2987                                                                    ;10=RP04,5,6 DUAL PORT  20=TM02
2988
2989
2990  005054  000010                    TSUNIT: .BLKW   8.              ;UNIT NO. TO BE TESTED IN ORDER OF TEST
2991
2992
2993  005074  000010                    TSSLAV: .BLKW   8.              ;SLAVE NO TO BE TESTED IN ORDER OF TEST
2994
2995
2996  005114  000000                    TSTOTL: 0                       ;TOTAL NO. OF UNITS TO BE TESTED
2997  005116  000000                    TSINDX: 0                       ;INDEX TO ONE UNDER TEST
2998
2999
3000  005120  000010                    TSVEC:  .BLKW   8.              ;VECTOR ADDRESS IN ORDER OF TEST
3001
3002
```

```
3003  005140  000010                  TSBAE:  .BLKW   8.              ;RHBAE ADDRESS IN ORDER OF TEST
3004
3005
3006  005160  000010                  TSCS3:  .BLKW   8.              ;RHCS3 ADDRESS IN ORDER OF TEST
3007
3008
3009  005200  000010                  TSCOMD: .BLKW   8.              ;COMMAND ADDRESS IN ORDER OF TEST
3010
3011
3012  005220  000010                  TSBASE: .BLKW   8.              ;BASE ADDRESS IN ORDER OF TEST
3013
3014
3015
3016
3017
3018
3019  005240  000000                  ERFLG$: 0                       ;ERROR FLAG
3020  005242  000000                  FIRST:  0                       ;IF ZERO WILL TYPE HEADER
3021                                                                          ;IF ONES WILL NOT TYPE HEADER
3022
3023
3024
3025  005244  000000                  ATTENT: 0               ;ATTENTION BIT FOR PRESENT UNIT
3026  005246  000000                  TOTALAT:        0                       ;TATAL ATTENTION BITS
3027
3028  005250  000000                  TMP0:   .WORD   0               ;TEMP STORAGE
3029  005252  000000                  TMP1:   .WORD   0
3030  005254  000000                  TMP4:   .WORD   0               ;TEMP STORAGE
3031                                  ;PERMANENT TABLE
3032  005256  172040                  PERRS:  172040                  ;RS BASE ADDRESS
3033  005260  176700                  PERRP:  176700                  ;RP BASE ADDRESS
3034  005262  172440                  PERTU:  172440                  ;TU BASE ADDRESS
3035  005264  176300                  PERMX:  176300                  ;MIXED BASE ADDRESS
3036
3037  005266  000204                  PERRSV: 204                     ;RS VECTOR ADDRESS
3038  005270  000254                  PERRPV: 254                     ;RP VECTOR ADDRESS
3039  005272  000224                  PERTUV: 224                     ;TU VECTOR ADDRESS
3040  005274  000000                  PERMXV: 0                       ;MIXED VECTOR ADDRESS
3041
3042  005276  000004                  PERNUM: 4                       ;NUMBER OF RH
3043
3044                                  ;WORKING TABLE
3045
3046  005300  000004                  DEVPNT: .BLKW   4               ;BASE ADDRESS TO BE TESTED
3047  005310  000004                  VECPNT: .BLKW   4               ;VECTOR ADDRESS TO BE TESTED
3048  005320  000000                  LTRY:   0                       ;NO. OF UNITS TO BE TESTED
3049  005322  000000                  TFOUND: 0                       ;TWICE NUMBER OF RH FOUND
3050  005324  000000                  TESTDV: 0                       ;STORES BASE ADDRESS OF TEST DEVICE
3051  005326  000000                  TESTVC: 0                       ;STORES VECTOR OF TEST DEVICE
3052                                                                  ;THE ABOVE TWO IS BEFORE ANY
3053                                                                  ;DEVICE IS FOUND.
3054                                  ;TABLE FOR DRIVE TYPES
3055                                  ;0=RS03, 1=RS03/L, 2=RS04, 3=RS04/L, 4=RS03/LA
3056                                  ;20020=SINGLE PORT RP04, 24020=DUAL PORT RP04
3057                                  ;20021=SINGLE PORT RP05, 24021=DUAL PORT RP05
3058                                  ;20022=SINGLE PORT RP06, 24022=DUAL PORT RP06
```

```
                                        ;142010=TU16
                                        
005330  000000  000001  000002  TYPNT:  .WORD   0,1,2,3,4,20020,24020,20021,24021,20022,24022,142010
005336  000003  000004  020020
005344  024020  020021  024021
005352  020022  024022  142010
005360  000000                  POINTT: 0                       ;POITER TO ABOVE TABLE
005362  000014                  NTYPNT: 14                      ;NUMBER OF ABOVE TYPES

005364  000000                  TMPSLV: 0                       ;TEMPORARY SLAVE COUNTER

                                        ;TABLE OF DATA FOR DEVICES FOUND
005366  000004                  FDEVIC: .BLKW   4               ;DEVICE FOUND, 1=TU, 2=RP DUAL
                                                                ;3=RP SINGLE, 4=RS04, 5=RS03
005376  000004                  FUNITN: .BLKW   4               ;UNIT NUMBER FOUND
005406  000004                  FTYPE:  .BLKW   4               ;DDRIVE TYPE FOUND
005416  000004                  FVECTR: .BLKW   4               ;VECTOR FOUND
005426  000004                  FBASEA: .BLKW   4               ;BASE ADDRESS FOUND
005436  000004                  FRHNM:  .BLKW   4               ;RH NUMBER FOUND
005446  000004                  FSLAVE: .BLKW   4               ;TU16 SLAVE NUMBER FOUND
005456  000004                  FDRIVR: .BLKW   4               ;DRIVER ADDRESS FOUND
005466  000004                  FBAE:   .BLKW   4               ;BAE ADDRESS FOUND
005476  000004                  FCS3:   .BLKW   4               ;CS3 ADDRESS FOUND

005506  000000                  TSTUNT: 0                       ;TOTAL NUMBER OF RH FOUND AT
                                                                ;END OF SIZE
005510  000000                  WORUNT: 0                       ;SAME AS ABOVE BUT TO BE
                                                                ;DECREASED AT END PROGRAM

005512  000000                  VECTOR: 0                       ;VECTOR UNDER TEST
005514  177740                  LERADD: 177740                  ;LOW ERROR ADDRESS REG.
005516  177742                  HERADD: 177742                          ;HIGH ERROR ADDRESS REG
005520  177744                  MEMERR: 177744                          ;MEMORY SYSTEM ADDRESS REG
        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  043104  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
005564  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
005572  012737  044730  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
005600  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
005606  012737  046434  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
005614  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
005622  012737  046514  000024          MOV     #$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  043042                  MOV     #RPVECT,(R0)+           ;THIS IS FOR UNTIMELY INTERRUPTS
006002  012710  000340                  MOV     #340,(R0)               ;RP04 INTERRUPT SERVICE ROUTINE
                                                                        ;PRIORITY = 7
006006  004737  044050          JSR     PC,@#$TKINT     ;INITILIZE THE TK
006012  005037  045064          CLR     PRITEM                  ;CLEAR FOR ERROR MESSAGE PRINTOUT
006016  005737  005242          TST     @#FIRST                 ;IS THIS FIRST TIME ROUND
006022  001001                  BNE     1$                      ;BRANCH IF NOT
006024  000402                  BR      2$
```

```
3149   006026  000137  006102           1$:     JMP     @#SND1
3150   006032  023737  000042  000046   2$:     CMP     @#42,@#46       ;ARE WE IN QV OR AUTO ACCEPT MODE?
3151   006040  001420                           BEQ     SND1            ;IF YES, SKIP HEADER
3152   006042  104401  006050                   TYPE    ,68$            ;;TYPE ASCIZ STRING
3153   006046  000415                           BR      67$             ;;GET OVER THE ASCIZ
3154                                    ;;68$:  .ASCIZ  <15><12>/CERHADO RH70 CTRLR DIAG/
3155   006102                           67$:
3156   006102  012737  177777  005242   SND1:   MOV     #-1,@#FIRST             ;NEXT TIME DO NOT GIVE HEADER
3157   006110  012737  037732  000114           MOV     #PARITY,@#PARE70        ;PARITY VECTOR
3158   006116  012737  000340  000116           MOV     #340,@#PARE70+2         ;PRIORITY 7
3159   006124  012737  037574  000004           MOV     #TIEOUT,@#ERRVEC        ;TIMEOUT VECTOR
3160   006132  012737  000340  000006           MOV     #340,@#ERRVEC+2         ;PRIORITY 7
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180                            ;;*****************************************************************
3181                            ;*TEST 1         SIZE FOR RH DEVICES
3182                            ;*      THIS TEST DETERMINS WHICH RH DEVICE IS ON THE SYSTEM.
3183                            ;*      IF THE SYSTEM HAS MORE THAN ONE RH DEVICE THEN ONLY ONE OF
3184                            ;*      THE DEVICES WILL BE USED IN THE FOLLOWING TESTS.
3185                            ;*      THE ONE THAT WILL BE USED IS THE FIRST ONE THAT IS PRESENT
3186                            ;*      IN THE FOLLOWING LIST
3187                            ;*      RS
3188                            ;*      RP
3189                            ;*      TM
3190                            ;*
3191                            ;*      THE WAY THE TEST WORKS IS AS FOLLOWS:-
3192                            ;*      A REFRENCE IS MADE TO THE CONTROL AND STATUS REGISTER 1
3193                            ;*      FOR THE RS BY A TST INSTRUCTION. IF IT RESPONDS THEN IT IS
3194                            ;*      ASSUMED PRESENT AND THE LOCATIONS FOR THE I/O REGISTERS ARE
3195                            ;*      FILLED WITH THE APPROPIATE ADDRESSES.
3196                            ;*      THEN THE DRIVE TYPE REGISTER IS CHECKED TO HAVE GOOD
3197                            ;*      VALUES.
3198                            ;*      IF THE TST INSTRUCTION TRAPS TO A NON EXISTANT VECTOR
3199                            ;*      THEN A SIMILAR ATTEMPT IS MADE TO A RP CONTROL AND STATUS
3200                            ;*      REGISTER 1.
3201                            ;*      THEN A TM IS TRYED.
3202                            ;*      THEN A MIXED SYSTEM WITH MORE THAN ONE RH DEVICS IS TRYED.
3203
3204                            ;;*****************************************************************
```

```
3205  006140  000004                    TST1:   SCOPE
3206  006142  012737  000001  001212            MOV     #1,$TIMES       ;;DO 1 ITERATION
3207  006150  005737  005012                    TST     ST200           ;IS IT A 200 START
3208  006154  001413                            BEQ     2$              ;BRANCH IF NOT 200 START
3209  006156  012700  005256                    MOV     #PERRS,R0       ;POINTER TO MOVE 9 WMRDS FROM
3210  006162  012701  000011                    MOV     #9.,R1          ;NUMBER TO MOVE
3211  006166  012702  005300                    MOV     #DEVPNT,R2      ;POINTER TO MOVE 9 TORDS TO
3212  006172  012022                    1$:     MOV     (R0)+,(R2)+     ;FILL WORK TABLE
3213  006174  005301                            DEC     R1
3214  006176  001375                            BNE     1$              ;BRANCH IF 9 NOT DONE
3215  006200  000137  006500                    JMP     SETSIZ
3216  006204                            2$:
3217  006204  104401  006212                    TYPE    ,65$            ;;TYPE ASCIZ STRING
3218  006210  000417                            BR      64$             ;;GET OVER THE ASCIZ
3219                                    ;;65$:  .ASCIZ  <15><12>/HOW MANY RH TO BE TESTED   /
3220  006250                            64$:
3221  006250  104401  006256                    TYPE    ,67$            ;;TYPE ASCIZ STRING
3222  006254  000402                            BR      66$             ;;GET OVER THE ASCIZ
3223                                    ;;67$:  .ASCIZ  <15><12>/ /
3224  006262                            66$:
3225  006262  104410                            RDOCT
3226  006264  011637  005320                    MOV     (SP),LTRY       ;GET NUMBER OF RH TO BE TESTED
3227  006270  012637  005250                    MOV     (SP)+,TMP0      ;WORKING NUMBER OF RH
3228  006274  012737  000001  005252            MOV     #1,TMP1         ;RH NUMBER TO BE TYPED
3229  006302  005000                            CLR     R0              ;INDEX TO WORKING TABLE
3230
3231  006304                            3$:
3232  006304  104401  006312                    TYPE    ,69$            ;;TYPE ASCIZ STRING
3233  006310  000417                            BR      68$             ;;GET OVER THE ASCIZ
3234                                    ;;69$:  .ASCIZ  <15><12>/TYPE BASE ADDRESS OF RH NO/
3235  006350                            68$:
3236  006350  013746  005252                    MOV     TMP1,-(SP)
3237  006354  104405                            TYPDS
3238  006356  104401  006364                    TYPE    ,71$            ;;TYPE ASCIZ STRING
3239  006362  000402                            BR      70$             ;;GET OVER THE ASCIZ
3240                                    ;;71$:  .ASCIZ  <15><12>/ /
3241  006370                            70$:
3242  006370  104410                            RDOCT
3243  006372  012660  005300                    MOV     (SP)+,DEVPNT(R0)  ;FILL WORKING TABLE WITH BASE ADDRESS
3244  006376  104401  006404                    TYPE    ,73$            ;;TYPE ASCIZ STRING
3245  006402  000420                            BR      72$             ;;GET OVER THE ASCIZ
3246                                    ;;73$:  .ASCIZ  <15><12>/TYPE VECTOR ADDRESS OF RH NO/
3247  006444                            72$:
3248  006444  013746  005252                    MOV     TMP1,-(SP)
3249  006450  104405                            TYPDS
3250  006452  104401  000200                    TYPE    ,CRLF
3251  006456  104410                            RDOCT
3252  006460  012660  005310                    MOV     (SP)+,VECPNT(R0)  ;FILL WORKING TABLE WITH VECTOR ADDRESS
3253  006464  005720                            TST     (R0)+           ;ADD 2 TO R0
3254  006466  005237  005252                    INC     TMP1            ;GET NEXT RH NUMBER
3255  006472  005337  005250                    DEC     TMP0            ;COUNT DOWN RH TO BE TESTED
3256  006476  001302                            BNE     3$              ;BRANCH IF ALL NOT DONE
3257
3258                                    ;R0 WILL HAVE DEVICE POINTER, R1 WILL HAVE VECTOR POINTER
3259  006500  012700  005300           SETSIZ: MOV     #DEVPNT,R0      ;DEVICE POINTER
3260  006504  012701  005310                    MOV     #VECPNT,R1      ;VECTOR PMINTER
```

```
3261  006510  005037  005322                  CLR     TFOUND          ;WILL BE USED AS POINTER
3262  006514                          SS2:                            ;TO FOUND TABLE.
3263  006514  012737  010506  000004  1$:     MOV     #SS4,ERRVEC     ;TIME OUT VECTOR TO REDUCE LTRY
3264  006522  012037  005324                  MOV     (R0)+,TESTDV    ;ADDRESS OF DEVICE TO BE TESTED
3265  006526  012137  005326                  MOV     (R1)+,TESTVC    ;VECTOR TO BE TESTED
3266  006532  005777  176566                  TST     @TESTDV         ;TRY TIME OUT ON DEVICE BASE
3267                                          ;IF NO TIME OUT OCCURS THEN DEVICE IN TESTDV IS PRESENT
3268                                          ;NOW FILL ALL REGISTER ADDRESS TABLE
3269  006536  012702  000026                  MOV     #22.,R2         ;COUNT 22
3270  006542  012703  004060                  MOV     #RHCS1,R3       ;GET BASE LOCATION
3271  006546  013704  005324                  MOV     TESTDV,R4       ;GET BASE ADDRESS
3272  006552  010423                  2$:     MOV     R4,(R3)+        ;FILL PROPER ADDRESS
3273  006554  062704  000002                  ADD     #2,R4           ;INCREMENT ADDRESS BY 2
3274  006560  005302                          DEC     R2              ;COUNT DOWN
3275  006562  001373                          BNE     2$              ;BRANCH IF 22 NOT DONE.
3276  006564  005077  175300                  CLR     @RHCS2          ;SET UNIT NUMBER
3277  006570                          SS6:
3278  006570  012737  005330  005360  3$:     MOV     #TYPNT,POINTT   ;POINTER TO DRIVE TYPE
3279  006576  012737  000014  005362          MOV     #14,NTYPNT      ;NUMBER OF DRIVE TYPES
3280
3281  006604                          SS7:
3282  006604  022737  000001  005362  4$:     CMP     #1,NTYPNT       ;IS IT FOR TU
3283  006612  001424                          BEQ     5$              ;IF FOR TU BRANCH
3284  006614  017737  175266  001126          MOV     @RHDT,$BDDAT    ;READ DRIVE TYPE
3285  006622  012702  000014                  MOV     #14,R2  ;GET NUMBER FOR DT NOT DONE
3286  006626  163702  005362                  SUB     NTYPNT,R2               ;GET DT TO DO
3287  006632  006302                          ASL     R2              ;MULTIPLY R2 BY 2
3288  006634  026237  005330  001126          CMP     TYPNT(R2),$BDDAT  ;IS DRIVE TYPE FOUND
3289  006642  001402                          BEQ     16$             ;DRIVE TYPE FOUND SO BRANCH
3290  006644  000137  010532                  JMP     SS8
3291  006650  032777  010000  175212  16$:    BIT     #NED,@RHCS2     ;IS IT NON EXISTANT DRIVE
3292  006656  001437                          BEQ     8$              ;NED NOT SET, DRIVE TYPE FOUND SO BRANCH
3293                                          ;A DEVICE HAS NOT BEEN FOUND
3294  006660  000137  010524                  JMP     SS5             ;A DEVICE HAS NOT BEEN FOUND SO BRANCH
3295
3296                                          ;TRYING THE TU16 DRIVE TYPE
3297  006664  005037  005364          5$:     CLR     TMPSLV          ;SLAVE 0
3298  006670  013777  005364  175214  6$:     MOV     TMPSLV,@RHTC    ;MOVE SLAVE NUMBER IN TAPE CONTROL
3299  006676  017737  175204  001126          MOV     @RHDT,$BDDAT    ;READ DRIVE TYPE
3300  006704  032737  002000  001126          BIT     #SLVPR,$BDDAT   ;IS SLAVE PRESENT
3301  006712  001014                          BNE     7$
3302  006714  022737  000007  005364          CMP     #7,TMPSLV       ;IS 7 DONE
3303  006722  001002                          BNE     17$             ;BRANCH IF 7 DONE AND NONE FOUND
3304  006724  000137  010532                  JMP     SS8
3305  006730  005237  005364          17$:    INC     TMPSLV          ;GET NEXT SLAVE
3306  006734  012777  040000  175116          MOV     #TRE,@RHCS1     ;CLEAR CONTROLLER ERRORS
3307  006742  000752                          BR      6$              ;TRY NEXT SLAVE
3308                                          ;A DEVICE HAS BEEN FOUND SAVE SLAVE
3309  006744  013702  005322          7$:     MOV     TFOUND,R2       ;GET NO OF RH FOUND X2 FOR INDEX
3310  006750  013762  005364  005446          MOV     TMPSLV,FSLAVE(R2)
3311                                          ;A DEVICE HAS BEEN FOUND TEST VECTOR
3312  006756  012737  006772  000004  8$:     MOV     #9$,ERRVEC      ;TIME OUT VECTOR TO REPORT ERROR
3313  006764  005777  176336                  TST     @TESTVC         ;TRY TIME OUT ON VECTOR
3314  006770  000403                          BR      10$             ;A VECTOR EXISTS SO CONTINUE
3315  006772  104146                  9$:     ERROR   146             ;A DEVICE BASE ADDRESS DID NOT
3316                                                                  ;TIME OUT BUT ITS CORRESPONDING
```

```
3317                                                                    ;VECTOR ADDRESS TIMED OUT
3318                                                                    ;HIT CONTINUE TO REPEAT THIS
3319                                                                    ;TEST
3320   006774  000000                                           HALT
3321   006776  000767                                           BR      8$                      ;REPEAT THIS TEST
3322                                                            ;A DEVICE HAS BEEN FOUND AND VECTOR RESPONDS SO STORE RESULTS
3323   007000  013702  005322                          10$:     MOV     TFOUND,R2               ;GET NO OF RH FOUND X2 FOR INDEX
3324   007004  013762  005362  005366                          MOV     NTYPNT,FDEVIC(R2)       ;DEVICE 1=TU; 2,3,4,5,6,7=RP;
3325                                                                                            ;10,11,12,13,14=RS
3326   007012  017762  175052  005376                          MOV     @RHCS2,FUNITN(R2)       ;GET RHCS2
3327   007020  042762  177770  005376                          BIC     #177770,FUNITN(R2)      ;KEEP UNIT NUMBER
3328   007026  017762  175054  005406                          MOV     @RHDT,FTYPE(R2)         ;DRIVE BYTE
3329   007034  013762  005326  005416                          MOV     TESTVC,FVECTR(R2)       ;VECTOR
3330   007042  013762  005324  005426                          MOV     TESTDV,FBASEA(R2)       ;BASE ADDRESS
3331   007050  013762  005322  005436                          MOV     TFOUND,FRHNM(R2)        ;RH NUMBER X2 MINUS ONE
3332   007056  006262  005436                                  ASR     FRHNM(R2)               ;RH NUMBER MINUS ONE
3333   007062  005262  005436                                  INC     FRHNM(R2)               ;RH NUMBER
3334   007066  022762  000001  005366                          CMP     #1,FDEVIC(R2)           ;IS IT TU
3335   007074  001016                                          BNE     11$                     ;BRANCH IF NOT TU
3336   007076  012762  041674  005456                          MOV     #CMNDTM,FDRIVR(R2)      ;DRIVER ADDRESS FOR TU
3337   007104  013746  005324                                  MOV     TESTDV,-(SP)            ;BASE ADDRESS
3338   007110  062716  000034                                  ADD     #<2*14.>,(SP)           ;15TH ADDRESS IS BAE FOR TU
3339   007114  011662  005466                                  MOV     (SP),FBAE(R2)           ;SAVE BAE FOR TU
3340   007120  062716  000002                                  ADD     #2,(SP)                 ;16TH ADDRESS IS CS3 FOR TU
3341   007124  012662  005476                                  MOV     (SP)+,FCS3(R2)          ;SAVE CS3 FOR TU
3342
3343   007130  000511                                          BR      SS1
3344   007132  022762  000002  005366  11$:     CMP     #2,FDEVIC(R2)           ;IS IT RP DUAL PORT
3345   007140  001424                                          BEQ     12$                     ;BRANCH IF RP FOUND
3346   007142  022762  000003  005366                          CMP     #3,FDEVIC(R2)           ;IS IT RP
3347   007150  001420                                          BEQ     12$                     ;BRANCH IF RP FOUND
3348   007152  022762  000004  005366                          CMP     #4,FDEVIC(R2)           ;IS IT RP DUAL PORT
3349   007160  001414                                          BEQ     12$                     ;BRANCH IF RP FOUND
3350   007162  022762  000005  005366                          CMP     #5,FDEVIC(R2)           ;IS IT RP
3351   007170  001410                                          BEQ     12$                     ;BRANCH IF RP FOUND
3352   007172  022762  000006  005366                          CMP     #6,FDEVIC(R2)           ;IS IT RP DUAL PORT
3353   007200  001404                                          BEQ     12$                     ;BRANCH IF RP FOUND
3354   007202  022762  000007  005366                          CMP     #7,FDEVIC(R2)           ;IS IT RP
3355   007210  001016                                          BNE     13$                     ;BRANCH IF NO RP
3356   007212  012762  041556  005456  12$:     MOV     #CMNDRP,FDRIVR(R2)      ;DRIVER ADDRESS FOR RP
3357   007220  013746  005324                                  MOV     TESTDV,-(SP)            ;BASE ADDRESS
3358   007224  062716  000050                                  ADD     #<2*20.>,(SP)           ;21ST ADDRESS IS BAE FOR RP
3359   007230  011662  005466                                  MOV     (SP),FBAE(R2)           ;SAVE BAE FOR RP
3360   007234  062716  000002                                  ADD     #2,(SP)                 ;22ND ADDRESS IS CS3 FOR RP
3361   007240  012662  005476                                  MOV     (SP)+,FCS3(R2)          ;SAVE CS3 FOR RP
3362
3363   007244  000443                                          BR      SS1
3364   007246  022762  000010  005366  13$:     CMP     #10,FDEVIC(R2)          ;IS IT RS03/LA
3365   007254  001420                                          BEQ     14$                     ;BRANCH IF RS03/LA
3366   007256  022762  000011  005366                          CMP     #11,FDEVIC(R2)          ;IS IT RS04/L
3367   007264  001414                                          BEQ     14$                     ;BRANCH IF RS04/L
3368   007266  022762  000012  005366                          CMP     #12,FDEVIC(R2)          ;IS IT RS04
3369   007274  001410                                          BEQ     14$                     ;BRANCH IF RS04
3370   007276  022762  000013  005366                          CMP     #13,FDEVIC(R2)          ;IS IT RS03/L
3371   007304  001404                                          BEQ     14$                     ;BRANCH IF RS03/L
3372   007306  022762  000014  005366                          CMP     #14,FDEVIC(R2)          ;IS IT RS03
```

```
3373  007314  001016                          BNE     15$                     ;BRANCH IF NOT RS03
3374  007316  012762  041466  005456  14$:    MOV     #CMNDRS,FDRIVR(R2)      ;DRIVER ADDRESS FOR RS
3375  007324  013746  005324                  MOV     TESTDV,-(SP)            ;BASE ADDRESS
3376  007330  062716  000030                  ADD     #(2*12.),(SP)           ;13TH ADDRESS IS BAE FOR RS
3377  007334  011662  005466                  MOV     (SP),FBAE(R2)           ;SAVE BAE FOR RS
3378  007340  062716  000002                  ADD     #2,(SP)                 ;14TH ADDRESS IS CS3 FOR RS
3379  007344  012662  005478                  MOV     (SP)+,FCS3(R2)          ;SAVE CS3 FOR RS
3380  007350  000401                          BR      SS1                     ;CONTINUE
3381  007352  000000                  15$:    HALT                            ;PROGRAM ERROR
3382
3383                                          ;NOW TYPE RESULT
3384
3385  007354                          SS1:
3386  007354  104401  007362                  TYPE    ,65$                    ;;TYPE ASCIZ STRING
3387  007360  000406                          BR      64$                     ;;GET OVER THE ASCIZ
3388                                  ;;65$:  .ASCIZ  <15><12>/ON RH NO/
3389  007376                          64$:
3390  007376  016246  005436                  MOV     FRHNM(R2),-(SP) ;TYPE RH NUMBER
3391  007402  104405                          TYPDS
3392  007404  104401  007412                  TYPE    ,67$                    ;;TYPE ASCIZ STRING
3393  007410  000411                          BR      66$                     ;;GET OVER THE ASCIZ
3394                                  ;;67$:  .ASCIZ  /            BASE /
3395  007434                          66$:
3396  007434  016246  005426                  MOV     FBASEA(R2),-(SP)  ;TYPE BASE ADDRESS
3397  007440  104402                          TYPOC
3398  007442  104401  007450                  TYPE    ,69$                    ;;TYPE ASCIZ STRING
3399  007446  000404                          BR      68$                     ;;GET OVER THE ASCIZ
3400                                  ;;69$:  .ASCIZ  / FOUND/
3401  007460                          68$:
3402  007460  022762  000001  005366          CMP     #1,FDEVIC(R2)           ;IS IT TU
3403  007466  001020                          BNE     1$                      ;BRANCH IF NOT TU
3404  007470  104401  007476                  TYPE    ,71$                    ;;TYPE ASCIZ STRING
3405  007474  000410                          BR      70$                     ;;GET OVER THE ASCIZ
3406                                  ;;71$:  .ASCIZ  / TU16 AT SLAVE/
3407  007516                          70$:
3408  007516  016246  005446                  MOV     FSLAVE(R2),-(SP)  ;TYPE SLAVE NUMBER
3409  007522  104405                          TYPDS
3410  007524  000137  010324                  JMP     6$
3411                                                                          ;THE FOLLOWING TYPES OUT
3412                                                                          ;THE DEVICE AND REDEFINES
3413                                                                          ;FDEVIC: 1=TU, 2=RP DUAL PORT
3414                                                                          ;3=RP SINGLE PORT
3415                                                                          ;4=RS04/L,RS04
3416                                                                          ;5=RS03/LA,RS03/L,RS03
3417  007530  022762  000002  005366  1$:      CMP     #2,FDEVIC(R2)           ;IS IT RP06 DUAL PORT
3418  007536  001016                          BNE     2$                      ;BRANCH IF NOT RP06 DUAL PORT
3419  007540  104401  007546                  TYPE    ,73$                    ;;TYPE ASCIZ STRING
3420  007544  000411                          BR      72$                     ;;GET OVER THE ASCIZ
3421                                  ;;73$:  .ASCIZ  / RP06 DUAL PORT /
3422  007570                          72$:
3423  007570  000137  010324                  JMP     6$
3424  007574  022762  000003  005366  2$:      CMP     #3,FDEVIC(R2)           ;IS IT RP06 SINGLE PORT
3425  007602  001016                          BNE     3$                      ;BRANCH IF NOT RP06 SINGLE PORT
3426  007604  104401  007612                  TYPE    ,75$                    ;;TYPE ASCIZ STRING
3427  007610  000411                          BR      74$                     ;;GET OVER THE ASCIZ
3428                                  ;;75$:  .ASCIZ  / RP06 SINGLE PORT/
```

```
                                        74$:
007634                                          JMP     6$
007634  000137  010324                          CMP     #4,FDEVIC(R2)   ;IS IT RP05 DUAL PORT
007640  022762  000004  005366        3$:       BNE     4$              ;BRANCH IF NOT RP05 DUAL PORT
007646  001020                                  MOV     #2,FDEVIC(R2)
007650  012762  000002  005366                  TYPE    ,77$            ;;TYPE ASCIZ STRING
007656  104401  007664                          BR      76$             ;;GET OVER THE ASCIZ
007662  000410                          ;;77$:  .ASCIZ  / RP05 DUAL PORT/
                                        76$:
007704                                          JMP     6$
007704  000137  010324                          CMP     #5,FDEVIC(R2)   ;IS IT RP05 SINGLE PORT
007710  022762  000005  005366        4$:       BNE     8$              ;BRANCH IF NOT RP05 SINGLE PORT
007716  001020                                  MOV     #3,FDEVIC(R2)
007720  012762  000003  005366                  TYPE    ,79$            ;;TYPE ASCIZ STRING
007726  104401  007734                          BR      78$             ;;GET OVER THE ASCIZ
007732  000411                          ;;79$:  .ASCIZ  / RP05 SINGLE PORT/
                                        78$:
007756                                          BR      6$
007756  000562                          CMP     #6,FDEVIC(R2)   ;IS IT RP04 DUAL PORT
007760  022762  000006  005366        8$:       BNE     9$              ;BRANCH IF NOT RP04 DUAL PORT
007766  001017                                  MOV     #2,FDEVIC(R2)
007770  012762  000002  005366                  TYPE    ,81$            ;;TYPE ASCIZ STRING
007776  104401  010004                          BR      80$             ;;GET OVER THE ASCIZ
010002  000410                          ;;81$:  .ASCIZ  / RP04 DUAL PORT/
                                        80$:
010024                                          BR      6$
010024  000537                          CMP     #7,FDEVIC(R2)   ;IS IT RP04 SINGLE PORT
010026  022762  000007  005366        9$:       BNE     10$             ;BRANCH IF NOT RP04 SINGLE PORT
010034  001020                                  MOV     #3,FDEVIC(R2)
010036  012762  000003  005366                  TYPE    ,83$            ;;TYPE ASCIZ STRING
010044  104401  010052                          BR      82$             ;;GET OVER THE ASCIZ
010050  000411                          ;;83$:  .ASCIZ  / RP04 SINGLE PORT/
                                        82$:
010074                                          BR      6$
010074  000513                          CMP     #10,FDEVIC(R2)  ;IS IT RS03-LA
010076  022762  000010  005366        10$:      BNE     11$             ;BRANCH IF NOT RS03-LA
010104  001014                                  MOV     #5,FDEVIC(R2)
010106  012762  000005  005366                  TYPE    ,85$            ;;TYPE ASCIZ STRING
010114  104401  010122                          BR      84$             ;;GET OVER THE ASCIZ
010120  000405                          ;;85$:  .ASCIZ  / RS03-LA/
                                        84$:
010134                                          BR      6$
010134  000473                          CMP     #11,FDEVIC(R2)  ;IS IT RS04-L
010136  022762  000011  005366        11$:      BNE     12$             ;BRANCH IF NOT RS04-L
010144  001013                                  MOV     #4,FDEVIC(R2)
010146  012762  000004  005366                  TYPE    ,87$            ;;TYPE ASCIZ STRING
010154  104401  010162                          BR      86$             ;;GET OVER THE ASCIZ
010160  000404                          ;;87$:  .ASCIZ  / RS04-L/
                                        86$:
010172                                          BR      6$
010172  000454                          CMP     #12,FDEVIC(R2)  ;IS IT RS04
010174  022762  000012  005366        12$:      BNE     13$             ;BRANCH IF NOT RS04
010202  001012                                  MOV     #4,FDEVIC(R2)
010204  012762  000004  005366                  TYPE    ,89$            ;;TYPE ASCIZ STRING
010212  104401  010220                          BR      88$             ;;GET OVER THE ASCIZ
010216  000403                          ;;89$:  .ASCIZ  / RS04/
```

```
3485  010226                                  88$:
3486  010226  000436                          BR      6$
3487  010230  022762  000013  005366  13$:    CMP     #13,FDEVIC(R2)  ;IS IT RS03-L
3488  010236  001013                          BNE     14$             ;BRANCH IF NOT RS03-L
3489  010240  012762  000005  005366          MOV     #5,FDEVIC(R2)
3490  010246  104401  010254                  TYPE    ,91$            ;;TYPE ASCIZ STRING
3491  010252  000404                          BR      90$             ;;GET OVER THE ASCIZ
3492                                  ;;91$:  .ASCIZ  / RS03-L/
3493  010264                                  90$:
3494  010264  000417                          BR      6$
3495  010266  022762  000014  005366  14$:    CMP     #14,FDEVIC(R2)  ;IS IT RS03
3496  010274  001012                          BNE     5$              ;BRANCH IF NOT RS03
3497  010276  012762  000005  005366          MOV     #5,FDEVIC(R2)
3498  010304  104401  010312                  TYPE    ,93$            ;;TYPE ASCIZ STRING
3499  010310  000403                          BR      92$             ;;GET OVER THE ASCIZ
3500                                  ;;93$:  .ASCIZ  / RS03/
3501  010320                                  92$:
3502  010320  000401                          BR      6$
3503  010322  000000                  5$:     HALT                    ;PROGRAM ERROR
3504  010324                                  6$:
3505  010324  104401  010332                  TYPE    ,95$            ;;TYPE ASCIZ STRING
3506  010330  000411                          BR      94$             ;;GET OVER THE ASCIZ
3507                                  ;;95$:  .ASCIZ  <15><12>/AT UNIT NUMBER/
3508  010354                                  94$:
3509  010354  016246  005376                  MOV     FUNITN(R2),-(SP)  ;TYPE UNIT NUMBER
3510  010360  104405                          TYPDS
3511  010362  104401  010370                  TYPE    ,97$            ;;TYPE ASCIZ STRING
3512  010366  000411                          BR      96$             ;;GET OVER THE ASCIZ
3513                                  ;;97$:  .ASCIZ  /          VECTOR /
3514  010412                                  96$:
3515  010412  016246  005416                  MOV     FVECTR(R2),-(SP)  ;TYPE VECTOR
3516  010416  104402                          TYPOC
3517  010420  104401  010426                  TYPE    ,99$            ;;TYPE ASCIZ STRING
3518  010424  000402                          BR      98$             ;;GET OVER THE ASCIZ
3519                                  ;;99$:  .ASCIZ  <15><12>/ /
3520  010432                                  98$:
3521  010432  104401  010440                  TYPE    ,101$           ;;TYPE ASCIZ STRING
3522  010436  000402                          BR      100$            ;;GET OVER THE ASCIZ
3523                                  ;;101$: .ASCIZ  <15><12>/ /
3524  010444                                  100$:
3525
3526
3527  010444  062737  000002  005322          ADD     #2,TFOUND       ;GET READY FOR NEXT DEVICE
3528
3529                                          ;ALL UNITS DONE? IF LTRY BECOMES ZERO YES ALL DONE
3530  010452  005337  005320                  DEC     LTRY            ;REDUCE DEVICES LEFT TO TRY
3531  010456  001402                          BEQ     7$              ;BRANCH IF ALL COMPLETE
3532  010460  000137  006514                  JMP     SS2             ;ALL NOT DONE
3533  010464                                  7$:
3534  010464  013746  005322          SS3:    MOV     TFOUND,-(SP)    ;GET TWICE NUMBER OF RH FOUND
3535  010470  006216                          ASR     (SP)            ;DIVIDE BY 2
3536  010472  011637  005506                  MOV     (SP),TSTUNT
3537  010476  012637  005510                  MOV     (SP)+,WORUNT
3538  010502  000137  010610                  JMP     TST2    ;JUMP TO NEXT TEST
3539
3540                                          ;A RH TIMED OUT SO TRY NEXT UNIT OR END TRYING
```

```
3541  010506  005337  005320          SS4:    DEC     LTRY            ;RH TIMES OUT SO DECREASE RH TYPED
3542  010512  001402                          BEQ     1$              ;BRANCH IF ALL DONE
3543  010514  000137  006514                  JMP     SS2             ;ALL NOT DONE SO JUMP.
3544  010520  000137  010464          1$:     JMP     SS3             ;ALL DONE SO STORE NUMBER OF RH FOUND
3545
3546                                          ;A DRIVE TYPE DID NOT MATCH SO TRY NEXT
3547  010524  012777  040000  173326  SS5:    MOV     #TRE,@RHCS1     ;CLEAR NED ERROR
3548  010532  005337  005362  SS8:    DEC     NTYPNT          ;DECREASE NUMBER OF DRIVE TYPES TRIED
3549  010536  001402                          BEQ     1$              ;BRANCH IF ALL DONE
3550  010540  000137  006604                  JMP     SS7             ;ALL NOT DONE
3551  010544  005277  173320          1$:     INC     @RHCS2          ;GET NEXT UNIT NO.
3552  010550  012777  040000  173302          MOV     #TRE,@RHCS1     ;CLEAR NED ERROR
3553  010556  017746  173306                  MOV     @RHCS2,-(SP)    ;GET RHCS2
3554  010562  042716  177770                  BIC     #177770,(SP)    ;GET UNIT NO
3555  010566  022726  000010                  CMP     #8.,(SP)+       ;ARE 7 DONE
3556  010572  001402                          BEQ     2$              ;BRANCH IF 7 DONE
3557  010574  000137  006570                  JMP     SS6             ;7 NOT DONE SO TRY NEXT UNIT NO
3558
3559  010600  104147                  2$:     ERROR   147             ;DEVICE ADDRESS DID NOT TIME
3560  010602  000000                          HALT                    ;OUT BUT NO UNITS HAD
3561                                                                  ;APPROPRIATE DRIVE TYPE.
3562  010604  000137  010506                  JMP     SS4
3563
3564
3565
3566
3567
3568
3569
3570                                  ;;**********************************************************************
3571                                  ;*TEST 2         UNIT UNDER TEST
3572                                  ;*      THIS TYPES THE UNIT TO BE TESTED
3573                                  ;*      AND IS THE FIRST TEST AFTER THE END OF THE PROGRAM
3574                                  ;;**********************************************************************
3575  010610  000004                  TST2:   SCOPE
3576  010612  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
3577  010620  012737  037574  000004          MOV     #TIEOUT,@#ERRVEC        ;TIMEOUT VECTOR
3578  010626  012737  000340  000006          MOV     #340,@#ERRVEC+2         ;PRIORITY 7
3579
3580  010634  013746  005510                  MOV     WORUNT,-(SP)    ;GET WORKING RH NUMBER LEFT
3581  010640  013702  005506                  MOV     TSTUNT,R2       ;GET TOTAL NO OF RH FOUND
3582  010644  162602                          SUB     (SP)+,R2        ;NO OF RH TO BE TESTED
3583  010646  006302                          ASL     R2              ;INDEX FOR RH TO BE TESTED.
3584  010650  104401  010656                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3585  010654  000402                          BR      64$             ;;GET OVER THE ASCIZ
3586
3587  010662                          ;;65$:  .ASCIZ  <15><12>/ /
                                        64$:
3588  010662  104401  010670                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3589  010666  000410                          BR      66$             ;;GET OVER THE ASCIZ
3590
3591  010710                          ;;67$:  .ASCIZ  <15><12>/TESTING RH NO/
                                        66$:
3592  010710  016246  005436                  MOV     FRHNM(R2),-(SP) ;TYPE RH NO.
3593  010714  104405                          TYPDS
3594
3595  010716  104401  010724                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3596  010722  000410                          BR      68$             ;;GET OVER THE ASCIZ
```

```
                                        ;;69$:  .ASCIZ  /        USING /
                                        68$:
                        010744  104401  010752          TYPE    ,71$            ;;TYPE ASCIZ STRING
                        010750  000404                  BR      70$             ;;GET OVER THE ASCIZ
                                        ;;71$:  .ASCIZ  / BASE /
                                        70$:
                        010762  016246  005426          MOV     FBASEA(R2),-(SP)  ;TYPE BASE ADDRESS
                                                        TYPOC
                        010770  022762  000001  005366  CMP     #1,FDEVIC(R2)   ;IS IT TU
                        010776  001017                  BNE     1$              ;BRANCH IF NOT TU
                        011000  104401  011006          TYPE    ,73$            ;;TYPE ASCIZ STRING
                        011004  000410                  BR      72$             ;;GET OVER THE ASCIZ
                                        ;;73$:  .ASCIZ  / TU16 AT SLAVE/
                                        72$:
                        011026  016246  005446          MOV     FSLAVE(R2),-(SP)  ;TYPE SLAVE NUMBER
                                                        TYPDS
                        011034  000467                  BR      6$
                        011036  022762  000002  005366  1$:     CMP     #2,FDEVIC(R2)   ;IS IT RP DUAL PORT
                        011044  001014                  BNE     2$              ;BRANCH IF NOT RP DUAL PORT
                        011046  104401  011054          TYPE    ,75$            ;;TYPE ASCIZ STRING
                        011052  000410                  BR      74$             ;;GET OVER THE ASCIZ
                                        ;;75$:  .ASCIZ  / RP DUAL PORT /
                                        74$:
                        011074  000447                  BR      6$
                        011076  022762  000003  005366  2$:     CMP     #3,FDEVIC(R2)   ;IS IT RO SINGLE PORT
                        011104  001014                  BNE     3$              ;BRANCH IF NOT RP
                        011106  104401  011114          TYPE    ,77$            ;;TYPE ASCIZ STRING
                        011112  000410                  BR      76$             ;;GET OVER THE ASCIZ
                                        ;;77$:  .ASCIZ  / RP SINGLE PORT/
                                        76$:
                        011134  000427                  BR      6$
                        011136  022762  000004  005366  3$:     CMP     #4,FDEVIC(R2)   ;IS IT RS04
                        011144  001007                  BNE     4$              ;BRANCH IF NOT RS04
                        011146  104401  011154          TYPE    ,79$            ;;TYPE ASCIZ STRING
                        011152  000403                  BR      78$             ;;GET OVER THE ASCIZ
                                        ;;79$:  .ASCIZ  / RS04/
                                        78$:
                        011162  000414                  BR      6$
                        011164  022762  000005  005366  4$:     CMP     #5,FDEVIC(R2)   ;IS IT RS03
                        011172  001007                  BNE     5$              ;BRANCH IF NOT RS03
                        011174  104401  011202          TYPE    ,81$            ;;TYPE ASCIZ STRING
                        011200  000403                  BR      80$             ;;GET OVER THE ASCIZ
                                        ;;81$:  .ASCIZ  / RS03/
                                        80$:
                        011210  000401                  BR      6$
                        011212  000000          5$:     HALT                    ;PROGRAM ERROR

                        011214                  6$:
                        011214  104401  011222          TYPE    ,83$            ;;TYPE ASCIZ STRING
                        011220  000411                  BR      82$             ;;GET OVER THE ASCIZ
                                        ;;83$:  .ASCIZ  <15><12>/AT UNIT NUMBER/
                                        82$:
                        011244  016246  005376          MOV     FUNITN(R2),-(SP)  ;TYPE UNIT NUMBER
                        011250  104405                  TYPDS
                        011252  104401  011260          TYPE    ,85$            ;;TYPE ASCIZ STRING
                        011256  000411                  BR      84$             ;;GET OVER THE ASCIZ
```

```
3653                                            ;;85$:  .ASCIZ  /          VECTOR /
3654   011302                                   84$:
3655   011302  016246  005416                            MOV     FVECTR(R2),-(SP)  ;TYPE VECTOR
3656   011306  104402                                    TYPOC
3657   011310  012703  000024                            MOV     #20.,R3           ;COUNT 20
3658   011314  012704  004060                            MOV     #RHCS1,R4         ;GET BASE LOCATION
3659   011320  016205  005426                            MOV     FBASEA(R2),R5     ;GET BASE ADDRESS
3660   011324  010524                          7$:       MOV     R5,(R4)+          ;FILL PROPER ADDRESS
3661   011326  062705  000002                            ADD     #2,R5             ;INCREMENT BY 2
3662   011332  005303                                    DEC     R3                ;COUNT DOWN
3663   011334  001373                                    BNE     7$                ;BRANCH IF 20 NOT DONE
3664
3665   011336  016237  005376  005010                    MOV     FUNITN(R2),UNIT   ;UNIT NUMBER
3666   011344  016237  005416  005512                    MOV     FVECTR(R2),VECTOR  ;VECTOR
3667   011352  016237  005446  004660                    MOV     FSLAVE(R2),SLAVE  ;SLAVE NO
3668   011360  016237  005456  004640                    MOV     FDRIVR(R2),COMND  ;DRIVER ADDRESS
3669   011366  016237  005466  004130                    MOV     FBAE(R2),RHBAE    ;BAE ADDRESS
3670   011374  016237  005476  004132                    MOV     FCS3(R2),RHCS3    ;CS3 ADDRESS
3671
3672   011402  005037  045064                            CLR     PRITEM
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682                                            ;;*****************************************************************
3683                                            ;*TEST 3         BIT BANG RHCS1
3684
3685                                            ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
3686                                            ;*      IN RHCS1 REGISTER.
3687                                            ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
3688                                            ;*      AND WALKING 0'S (-2,-3,-5 ETC)
3689                                            ;*      IN THIS TEST SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO BITS WILL NOT BE WRITTEN INTO
3690                                            ;*      AND RDY!DVA BITS WILL ALWAYS BE SET
3691                                            ;*      AND BIT10 BITS WILL ALWAYS BE CLEARED
3692                                            ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
3693                                            ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
3694                                            ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
3695                                            ;*      SIGNAL.
3696
3697                                            ;;*****************************************************************
3698   011406  000004                           TST3:   SCOPE
3699   011410  012706  001000                            MOV     #STACK,SP         ;RESET STACK
3700   011414  012737  000003  004656                    MOV     #3,@#TSTNM        ;SAVE TEST NUMBER
3701
3702   011422  004737  040306                            JSR     PC,@#CLDISK       ;GIVE RH INITIALIZE
3703                                                                              ;SETUP UNIT NUBER
3704                                                                              ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
3705
3706   011426  012737  176201  004650                    MOV     #SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO,WRTBIT      ;SC!TRE!MCPE!DVA!BIT12!B
3707   011434  012737  004200  004652                    MOV     #RDY!DVA,SETBIT   ;RDY!DVA ARE BITS ALWAYS SET
3708   011442  012737  002000  004654                    MOV #BIT10,CLRBIT         ;BIT10 ARE BITS ALWAYS CLEARED
```

```
3709
3710                                                    ;FLOAT 1'S THRU THE RHCS1 REGISTER
3711  011450  012737  011474  001110                    MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
3712  011456  012705  000001                          MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
3713  011462  010537  001124              1$:         MOV     R5,$GDDAT       ;START WITH DATA
3714  011466  042737  176201  001124                  BIC     #SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO,$GDDAT      ;CLEAR BITS NOT WRITTEN
3715  011474  032737  041400  001140    2$:         BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3716  011502  001403                                BEQ     3$              ;BRANCH IF ANY ONE SET
3717  011504  052777  000040  172356                BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
3718                                                    ;SYNC IF IN DEBUG:
3719  011512  013777  001124  172340    3$:         MOV     $GDDAT,@RHCS1   ;WRITE RHCS1 REGISTER
3720  011520  017737  172334  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1 REGISTER
3721  011526  043737  004654  001124                  BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
3722  011534  053737  004652  001124                  BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
3723  011542  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;TEST
3724  011550  001404                                BEQ     4$              ;BRANCH IF GOOD
3725  011552  013737  004060  001122                  MOV     RHCS1,$BDADR    ;GET REGISTER ADDRESS
3726  011560  104137                                ERROR   137             ;ONE WAS BEING FLOATED
3727                                                    ;THRU RHCS1 REGISTER
3728                                                    ;ON READING RHCS1 BACK
3729                                                    ;IT DID NOT CONTAIN WHAT
3730                                                    ;WAS EXPECTED.
3731  011562  000241                    4$:         CLC                     ;CLEAR CARRY
3732  011564  006305                                ASL     R5              ;GET 1 ONE LEFT
3733  011566  103335                                BCC     1$              ;BRANCH IF 16 NOT DONE
3734
3735                                                    ;FLOAT A ZERO THRU RHCS1 REGISTER.
3736
3737  011570  012737  011614  001110                    MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
3738  011576  012705  177776                          MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
3739  011602  010537  001124              5$:         MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
3740  011606  042737  176201  001124                  BIC     #SC!TRE!MCPE!DVA!BIT12!BIT10!RDY!GO,$GDDAT      ;CLEAR BITS NOT WRITTEN
3741  011614  032737  041400  001140    6$:         BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3742  011622  001403                                BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
3743  011624  012777  000040  172236                MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
3744                                                    ;SYNC IF IN DEBUG
3745  011632  013777  001124  172220    7$:         MOV     $GDDAT,@RHCS1   ;WRITE INTO RH'XA.
3746  011640  017737  172214  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1
3747  011646  053737  004652  001124                  BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
3748  011654  043737  004654  001124                  BIC     CLRBIT,$GDDAT   ;CLEAR BITS ALWAYS 0
3749  011662  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;TEST
3750  011670  001404                                BEQ     8$              ;BRANCH IF GOOD
3751  011672  013737  004060  001122                  MOV     RHCS1,$BDADR    ;GET REGISTER ADDRESS
3752  011700  104137                                ERROR   137             ;ZERO WAS BEING FLOATED
3753                                                    ;THRU RHCS1 REGISTER
3754                                                    ;ON READING IT BACK IT
3755                                                    ;DID NOT CONTAIN WHAT
3756                                                    ;WAS EXPECTED
3757
3758  011702  000261                    8$:         SEC
3759  011704  006105                                ROL     R5
3760  011706  103735                                BCS     5$
3761
3762
3763                                                ;;**********************************************************************
3764                                                ;*TEST 4         BIT BANG RHCS2
```

```
                                        ;*      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.
                                        ;;*****************************************************************
011710  000004                        TST4:   SCOPE
011712  012706  001000                        MOV     #STACK,SP       ;RESET STACK
011716  012737  000004  004656                MOV     #4,@#TSTNM      ;SAVE TEST NUMBER
011724  004737  040306                        JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
011730  012737  177740  004650                MOV     #177740,WRTBIT  ;177740 ARE BITS NOT WRITTEN INTO
011736  012737  000100  004652                MOV     #IR,SETBIT      ;IR ARE BITS ALWAYS SET
                                        ;FLOAT 1'S THRU THE RHCS2 REGISTER
011744  012737  011770  001110                MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
011752  012705  000001                        MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
011756  010537  001124          1$:   2$:     MOV     R5,$GDDAT       ;START WITH DATA
011762  042737  177740  001124                BIC     #177740,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
011770  032737  041400  001140                BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
011776  001403                                BEQ     3$              ;BRANCH IF ANY ONE SET
012000  052777  000040  172062                BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                                        ;SYNC IF IN DEBUG:
012006  013777  001124  172054          3$:   MOV     $GDDAT,@RHCS2   ;WRITE RHCS2 REGISTER
012014  017737  172050  001126                MOV     @RHCS2,$BDDAT   ;READ RHCS2 REGISTER
012022  053737  004652  001124                BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
012030  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;TEST
012036  001404                                BEQ     4$              ;BRANCH IF GOOD
012040  013737  004070  001122                MOV     RHCS2,$BDADR    ;GET REGISTER ADDRESS
012046  104137                                ERROR   137             ;ONE WAS BEING FLOATED
                                                                        ;THRU RHCS2 REGISTER
                                                                        ;ON READING RHCS2 BACK
                                                                        ;IT DID NOT CONTAIN WHAT
                                                                        ;WAS EXPECTED.
012050  000241                          4$:   CLC                     ;CLEAR CARRY
012052  006305                                ASL     R5              ;GET 1 ONE LEFT
012054  103340                                BCC     1$              ;BRANCH IF 16 NOT DONE
                                        ;FLOAT A ZERO THRU RHCS2 REGISTER.
012056  012737  012102  001110                MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
012064  012705  177776                        MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
012070  010537  001124          5$:   6$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
012074  042737  177740  001124                BIC     #177740,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
012102  032737  041400  001140                BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012110  001403                                BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
```

```
3821   012112  012777  000040  171750          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
3822                                                                   ;SYNC IF IN DEBUG
3823   012120  013777  001124  171742  7$:     MOV     $GDDAT,@RHCS2   ;WRITE INTO RH'XA.
3824   012126  017737  171736  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2
3825   012134  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
3826   012142  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
3827   012150  001404                          BEQ     8$              ;BRANCH IF GOOD
3828   012152  013737  004070  001122          MOV     RHCS2,$BDADR    ;GET REGISTER ADDRESS
3829   012160  104137                  $       ERROR   137             ;ZERO WAS BEING FLOATED
3830                                                                   ;THRU RHCS2 REGISTER
3831                                                                   ;ON READING IT BACK IT
3832                                                                   ;DID NOT CONTAIN WHAT
3833                                                                   ;WAS EXPECTED
3834
3835   012162  000261                  8$:     SEC
3836   012164  006105                          ROL     R5
3837   012166  103740                          BCS     5$
3838
3839
3840                                   ;;*****************************************************************
3841                                   ;*TEST 5         BIT BANG RHCS3
3842
3843                                   ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
3844                                   ;*      IN RHCS3 REGISTER.
3845                                   ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
3846                                   ;*      AND WALKING 0'S (-2,-3,-5 ETC)
3847                                   ;*      IN THIS TEST 177660 BITS WILL NOT BE WRITTEN INTO
3848                                   ;*      AND 0 BITS WILL ALWAYS BE SET
3849                                   ;*      AND BIT9!BIT8!BIT7!BIT5!BIT4 BITS WILL ALWAYS BE CLEARED
3850                                   ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
3851                                   ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
3852                                   ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
3853                                   ;*      SIGNAL.
3854
3855                                   ;;*****************************************************************
3856   012170  000004                  TST5:   SCOPE
3857   012172  012706  001000                  MOV     #STACK,SP       ;RESET STACK
3858   012176  012737  000005  004656          MOV     #5,@#TSTNM      ;SAVE TEST NUMBER
3859
3860   012204  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
3861                                                                   ;SETUP UNIT NUBER
3862                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
3863
3864   012210  012737  177660  004650          MOV     #177660,WRTBIT  ;177660 ARE BITS NOT WRITTEN INTO
3865   012216  012737  000000  004652          MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
3866   012224  012737  001660  004654          MOV #BIT9!BIT8!BIT7!BIT5!BIT4,CLRBIT    ;BIT9!BIT8!BIT7!BIT5!BIT4 ARE BITS ALWAY
3867
3868                                           ;FLOAT 1'S THRU THE RHCS3 REGISTER
3869   012232  012737  012256  001110          MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
3870   012240  012705  000001                  MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
3871   012244  010537  001124          1$:     MOV     R5,$GDDAT       ;START WITH DATA
3872   012250  042737  177660  001124          BIC     #177660,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
3873   012256  032737  041400  001140  2$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3874   012264  001403                          BEQ     3$              ;BRANCH IF ANY ONE SET
3875   012266  052777  000040  171574          BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
3876                                                                   ;SYNC IF IN DEBUG:
```

```
3877  012274  013777  001124  171630  3$:     MOV     $GDDAT,@RHCS3   ;WRITE RHCS3 REGISTER
3878  012302  017737  171624  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 REGISTER
3879  012310  043737  004654  001124          BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
3880  012316  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
3881  012324  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
3882  012332  001404                          BEQ     4$              ;BRANCH IF GOOD
3883  012334  013737  004132  001122          MOV     RHCS3,$BDADR    ;GET REGISTER ADDRESS
3884  012342  104137                          ERROR   137             ;ONE WAS BEING FLOATED
3885                                                                  ;THRU RHCS3 REGISTER
3886                                                                  ;ON READING RHCS3 BACK
3887                                                                  ;IT DID NOT CONTAIN WHAT
3888                                                                  ;WAS EXPECTED.
3889  012344  000241                  4$:     CLC                     ;CLEAR CARRY
3890  012346  006305                          ASL     R5              ;GET 1 ONE LEFT
3891  012350  103335                          BCC     1$              ;BRANCH IF 16 NOT DONE
3892
3893                                          ;FLOAT A ZERO THRU RHCS3 REGISTER.
3894
3895  012352  012737  012376  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
3896  012360  012705  177776                  MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
3897  012364  010537  001124          5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
3898  012370  042737  177660  001124          BIC     #177660,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
3899  012376  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
3900  012404  001403                          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
3901  012406  012777  000040  171454          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
3902                                                                  ;SYNC IF IN DEBUG
3903  012414  013777  001124  171510  7$:     MOV     $GDDAT,@RHCS3   ;WRITE INTO RH'XA.
3904  012422  017737  171504  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3
3905  012430  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
3906  012436  043737  004654  001124          BIC CLRBIT,$GDDAT       ;CLEAR BITS ALWAYS 0
3907  012444  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
3908  012452  001404                          BEQ     8$              ;BRANCH IF GOOD
3909  012454  013737  004132  001122          MOV     RHCS3,$BDADR    ;GET REGISTER ADDRESS
3910  012462  104137                          ERROR   137             ;ZERO WAS BEING FLOATED
3911                                                                  ;THRU RHCS3 REGISTER
3912                                                                  ;ON READING IT BACK IT
3913                                                                  ;DID NOT CONTAIN WHAT
3914                                                                  ;WAS EXPECTED
3915
3916  012464  000261                  8$:     SEC
3917  012466  006105                          ROL     R5
3918  012470  103735                          BCS     5$
3919
3920
3921                                          ;;*****************************************************************
3922                                          ;*TEST 6         BIT BANG RHWC
3923
3924                                          ;*      TEST LOADING AND READING OF ALL POSSIBLE BITS
3925                                          ;*      IN RHWC REGISTER.
3926                                          ;*      USE A PATTERN OF WALKING 1'S (1,2,4,10 ETC)
3927                                          ;*      AND WALKING 0'S (-2,-3,-5 ETC)
3928                                          ;*      IN THIS TEST 0 BITS WILL NOT BE WRITTEN INTO
3929                                          ;*      AND 0 BITS WILL ALWAYS BE SET
3930                                          ;*      IF SWITCH 14 OR 9 OR 8 ARE SET FOR DEBUGGING
3931                                          ;*      THEN A RH CLEAR (RHCS2 BIT #5) WILL BE
3932                                          ;*      GIVEN TO AID SCOPE SYNCS ON THE CLEAR
```

```
                                        ;*      SIGNAL.
                                        ;
                                        ;;*************************************************************
012472  000004                          TST6:   SCOPE
012474  012706  001000                          MOV     #STACK,SP       ;RESET STACK
012500  012737  000006  004656                  MOV     #6,@#TSTNM      ;SAVE TEST NUMBER
                                        ;
012506  004737  040306                          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                        ;
012512  012737  000000  004650                  MOV     #0,WRTBIT       ;0 ARE BITS NOT WRITTEN INTO
012520  012737  000000  004652                  MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
                                        ;
                                                ;FLOAT 1'S THRU THE RHWC REGISTER
012526  012737  012552  001110                  MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
012534  012705  000001                          MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
012540  010537  001124          1$:             MOV     R5,$GDDAT       ;START WITH DATA
012544  042737  000000  001124                  BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
012552  032737  041400  001140  2$:             BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012560  001403                                  BEQ     3$              ;BRANCH IF ANY ONE SET
012562  052777  000040  171300                  BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                                        ;SYNC IF IN DEBUG:
012570  013777  001124  171264  3$:             MOV     $GDDAT,@RHWC    ;WRITE RHWC REGISTER
012576  017737  171260  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC REGISTER
012604  053737  004652  001124                  BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
012612  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;TEST
012620  001404                                  BEQ     4$              ;BRANCH IF GOOD
012622  013737  004062  001122                  MOV     RHWC,$BDADR     ;GET REGISTER ADDRESS
012630  104137                                  ERROR   137             ;ONE WAS BEING FLOATED
                                                                        ;THRU RHWC REGISTER
                                                                        ;ON READING RHWC BACK
                                                                        ;IT DID NOT CONTAIN WHAT
                                                                        ;WAS EXPECTED.
012632  000241          4$:             CLC                             ;CLEAR CARRY
012634  006305                          ASL     R5                      ;GET 1 ONE LEFT
012636  103340                          BCC     1$                      ;BRANCH IF 16 NOT DONE
                                        ;
                                        ;FLOAT A ZERO THRU RHWC REGISTER.
                                        ;
012640  012737  012664  001110                  MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
012646  012705  177776                          MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
012652  010537  001124          5$:             MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
012656  042737  000000  001124                  BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
012664  032737  041400  001140  6$:             BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
012672  001403                                  BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
012674  012777  000040  171166                  MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
                                                                        ;SYNC IF IN DEBUG
012702  013777  001124  171152  7$:             MOV     $GDDAT,@RHWC    ;WRITE INTO RH'XA.
012710  017737  171146  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC
012716  053737  004652  001124                  BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
012724  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;TEST
012732  001404                                  BEQ     8$              ;BRANCH IF GOOD
012734  013737  004062  001122                  MOV     RHWC,$BDADR     ;GET REGISTER ADDRESS
012742  104137                                  ERROR   137             ;ZERO WAS BEING FLOATED
                                                                        ;THRU RHWC REGISTER
```

```
                                                        ;ON READING IT BACK IT
                                                        ;DID NOT CONTAIN WHAT
                                                        ;WAS EXPECTED

012744  000261                  8$:     SEC
012746  006105                          ROL     R5
012750  103740                          BCS     5$

                                ;;**********************************************************************
                                ;*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.

                                ;;**********************************************************************
012752  000004                  TST7:   SCOPE
012754  012706  001000                  MOV     #STACK,SP       ;RESET STACK
012760  012737  000007  004656          MOV     #7,@#TSTNM      ;SAVE TEST NUMBER

012766  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

012772  012737  000000  004650          MOV     #0,WRTBIT       ;0 ARE BITS NOT WRITTEN INTO
013000  012737  000000  004652          MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
013006  012737  000001  004654          MOV #BIT00,CLRBIT       ;BIT00 ARE BITS ALWAYS CLEARED

                                        ;FLOAT 1'S THRU THE RHBA REGISTER
013014  012737  013040  001110          MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
013022  012705  000001                  MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
013026  010537  001124          1$:     MOV     R5,$GDDAT       ;START WITH DATA
013032  042737  000000  001124          BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
013040  032737  041400  001140  2$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
013046  001403                          BEQ     3$              ;BRANCH IF ANY ONE SET
013050  052777  000040  171012          BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                                ;SYNC IF IN DEBUG:
013056  013777  001124  171000  3$:     MOV     $GDDAT,@RHBA    ;WRITE RHBA REGISTER
013064  017737  170774  001126          MOV     @RHBA,$BDDAT    ;READ RHBA REGISTER
013072  043737  004654  001124          BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
013100  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
013106  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
013114  001404                          BEQ     4$              ;BRANCH IF GOOD
013116  013737  004064  001122          MOV     RHBA,$BDADR     ;GET REGISTER ADDRESS
013124  104137                          ERROR   137             ;ONE WAS BEING FLOATED
                                                                ;THRU RHBA REGISTER
                                                                ;ON READING RHBA BACK
```

```
                                                        ;IT DID NOT CONTAIN WHAT
                                                        ;WAS EXPECTED.
013126  000241                  4$:     CLC             ;CLEAR CARRY
013130  006305                          ASL     R5      ;GET 1 ONE LEFT
013132  103335                          BCC     1$      ;BRANCH IF 16 NOT DONE

                                        ;FLOAT A ZERO THRU RHBA REGISTER.

013134  012737  013160  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
013142  012705  177776                  MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
013146  010537  001124          5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
013152  042737  000000  001124          BIC     #0,$GDDAT       ;CLEAR BITS NOT WRITTEN INTO
013160  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
013166  001403                          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
013170  012777  000040  170672          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
                                                                ;SYNC IF IN DEBUG
013176  013777  001124  170660  7$:     MOV     $GDDAT,@RHBA    ;WRITE INTO RH'XA.
013204  017737  170654  001126          MOV     @RHBA,$BDDAT    ;READ RHBA
013212  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
013220  043737  004654  001124          BIC CLRBIT,$GDDAT       ;CLEAR BITS ALWAYS 0
013226  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
013234  001404                          BEQ     8$              ;BRANCH IF GOOD
013236  013737  004064  001122          MOV     RHBA,$BDADR     ;GET REGISTER ADDRESS
013244  104137                          ERROR   137             ;ZERO WAS BEING FLOATED
                                                                ;THRU RHBA REGISTER
                                                                ;ON READING IT BACK IT
                                                                ;DID NOT CONTAIN WHAT
                                                                ;WAS EXPECTED

013246  000261                  8$:     SEC
013250  006105                          ROL     R5
013252  103735                          BCS     5$


                        ;;*******************************************************************
                        ;*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.

                        ;;*******************************************************************
013254  000004          TST10:  SCOPE
013256  012706  001000          MOV     #STACK,SP       ;RESET STACK
013262  012737  000010  004656  MOV     #10,@#TSTNM     ;SAVE TEST NUMBER

013270  004737  040306          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
```

```
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

013274  012737  177700  004650          MOV     #177700,WRTBIT  ;177700 ARE BITS NOT WRITTEN INTO
013302  012737  000000  004652          MOV     #0,SETBIT       ;0 ARE BITS ALWAYS SET
013310  012737  177700  004654          MOV #177700,CLRBIT      ;177700 ARE BITS ALWAYS CLEARED

                                        ;FLOAT 1'S THRU THE RHBAE REGISTER.
013316  012737  013342  001110          MOV     #2$,$LPERR      ;SET LOOP ON ERROR ADDRESS
013324  012705  000001                  MOV     #1,R5           ;GETTING READY TO FLOAT A ONE
013330  010537  001124          1$:     MOV     R5,$GDDAT       ;START WITH DATA
013334  042737  177700  001124          BIC     #177700,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
013342  032737  041400  001140  2$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
013350  001403                          BEQ     3$              ;BRANCH IF ANY ONE SET
013352  052777  000040  170510          BIS     #CLR,@RHCS2     ;GIVE CLEAR FOR SCOPE
                                                                ;SYNC IF IN DEBUG:
013360  013777  001124  170542  3$:     MOV     $GDDAT,@RHBAE   ;WRITE RHBAE REGISTER
013366  017737  170536  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE REGISTER
013374  043737  004654  001124          BIC     CLRBIT,$GDDAT   ;CLEAR ALWAYS CLEARED BITS
013402  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET ALWAYS SET BITS
013410  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
013416  001404                          BEQ     4$              ;BRANCH IF GOOD
013420  013737  004130  001122          MOV     RHBAE,$BDADR    ;GET REGISTER ADDRESS
013426  104137                          ERROR   137             ;ONE WAS BEING FLOATED
                                                                ;THRU RHBAE REGISTER
                                                                ;ON READING RHBAE BACK
                                                                ;IT DID NOT CONTAIN WHAT
                                                                ;WAS EXPECTED.
013430  000241                  4$:     CLC                     ;CLEAR CARRY
013432  006305                          ASL     R5              ;GET 1 ONE LEFT
013434  103335                          BCC     1$              ;BRANCH IF 16 NOT DONE

                                        ;FLOAT A ZERO THRU RHBAE REGISTER.
013436  012737  013462  001110          MOV     #6$,$LPERR      ;SET LOOP BACK POINT.
013444  012705  177776                  MOV     #177776,R5      ;GET READY TO FLOAT A ZERO
013450  010537  001124          5$:     MOV     R5,$GDDAT       ;GET READY TO WRITE DATA
013454  042737  177700  001124          BIC     #177700,$GDDAT  ;CLEAR BITS NOT WRITTEN INTO
013462  032737  041400  001140  6$:     BIT     #BIT14!BIT9!BIT8,SWR ;ARE SWITCHES 14 OR 9 OR 8 SET
013470  001403                          BEQ     7$              ;BRANCH IF ANY OF THE ABOVE SET
013472  012777  000040  170370          MOV     #CLR,@RHCS2     ;CLEAR FOR SCOPE AID TO
                                                                ;SYNC IF IN DEBUG
013500  013777  001124  170422  7$:     MOV     $GDDAT,@RHBAE   ;WRITE INTO RH'XA.
013506  017737  170416  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE
013514  053737  004652  001124          BIS     SETBIT,$GDDAT   ;SET BITS ALWAYS SET
013522  043737  004654  001124          BIC  CLRBIT,$GDDAT      ;CLEAR BITS ALWAYS 0
013530  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST
013536  001404                          BEQ     8$              ;BRANCH IF GOOD
013540  013737  004130  001122          MOV     RHBAE,$BDADR    ;GET REGISTER ADDRESS
013546  104137                          ERROR   137             ;ZERO WAS BEING FLOATED
                                                                ;THRU RHBAE REGISTER
                                                                ;ON READING IT BACK IT
                                                                ;DID NOT CONTAIN WHAT
                                                                ;WAS EXPECTED

013550  000261                  8$:     SEC
013552  006105                          ROL     R5
```

```
 4157  013554  103735                          BCS     5$
 4158
 4159
 4160
 4161
 4162
 4163
 4164                                  ;;*****************************************************************
 4165                                  ;*TEST 11        SILO TEST 1 (ONE WORD WRITE)
 4166                                  ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
 4167                                  ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH
 4168                                  ;*      TOGETHER WITH UNIT NUMBER
 4169                                  ;*      ONE WORD OF ALL ZEROS IS WRITTEN INTO RHDB
 4170                                  ;*      BY "CLR"
 4171                                  ;*      "OR" (BIT #7) IS WAITED FOR BY A TRAP INSTRUCTION
 4172                                  ;*      CALLED "WAT" (NO TIMING IS DONE)
 4173                                  ;*      HOWEVER IF "OR" DOES NOT SET WITHIN "WAT" COUNT
 4174                                  ;*      DOWN AN ERROR IS REPORTED
 4175                                  ;*      RHDB IS READ AND CHECKED TO CONTAIN ZEROS
 4176                                  ;*      RHCS2 WILL BE CHECKED TO HAVE "IR" AND UNIT NUMBER
 4177                                  ;*      RHCS1,RHCS3,RHBA,RHBAE,RHWC, WILL BE CHECKED TO HAVE
 4178                                  ;*      APPROPIATE VALUES
 4179                                  ;;*****************************************************************
 4180  013556  000004                  TST11:  SCOPE
 4181  013560  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 4182  013564  012737  000011  004656          MOV     #11,@#TSTNM     ;SAVE TEST NUMBER
 4183
 4184  013572  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
 4185                                                                  ;SETUP UNIT NUBER
 4186                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
 4187
 4188                                  ;CHECK THAT RHCS2 HAS IR
 4189  013576  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
 4190  013604  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 4191
 4192  013612  017737  170252  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 4193  013620  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 4194                                                                  ;DATA WITH DATA READ FROM
 4195                                                                  ;RHCS2
 4196  013626  001401                          BEQ     65$             ;BRANCH IF GOOD
 4197  013630  104006                          ERROR   6
 4198                                                                  ;AFTER SETTING CLR BIT #5
 4199                                                                  ;IN RHCS2 TO INIT THE RH
 4200                                                                  ;AND HAVING DONE NOTHING ELSE
 4201                                                                  ;RHCS2 SHOULD HAVE IR
 4202                                                                  ;=100
 4203                                                                  ;TOGETHER WITH UNIT NUMBER
 4204                                                                  ;BUT CONTAINED WHAT IS
 4205                                                                  ;GIVEN IN BAD RHCS2
 4206  013632                          65$:
 4207
 4208                                  ;WRITE ONE WORD OF ALL ZEROS INTO RHDB
 4209  013632  005077  170244                  CLR     @RHDB           ;WRITE 0 IN RHDB
 4210
 4211                                  ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
 4212  013636  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
```

```
                                        RHCS2                   ;AND WAIT FOR OR BIT IN
                                        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 0
013644  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =

013652  017737  170224  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
013660  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
013666  001401                          BEQ     67$             ;BRANCH IF GOOD
013670  104007                          ERROR   7
                                                                ;AFTER CLEARING THE RH
                                                                ;AND WRITING ALL ZEROS
                                                                ;INTO RHDB
                                                                ;THEN READING IT BACK
                                                                ;RHDB SHOULD HAVE 0
                                                                ;=
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
013672                          67$:
                                        ;CHECK THAT RHCS2 HAS IR
013672  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
013700  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

013706  017737  170156  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
013714  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
013722  001401                          BEQ     69$             ;BRANCH IF GOOD
013724  104010                          ERROR   10
                                                                ;AFTER CLEARING THE RH
                                                                ;AND WRITING ALL ZEROS
                                                                ;INTO RHDB
                                                                ;THEN READING IT BACK
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
013726                          69$:
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
013726  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

013734  017737  170120  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
013742  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
013750  001401                          BEQ     71$             ;BRANCH IF GOOD
```

```
4269  013752  104011                          ERROR   11
4270                                                    ;AFTER CLEARING THE RH
4271                                                    ;AND WRITING ALL ZEROS
4272                                                    ;INTO RHDB
4273                                                    ;THEN READING IT BACK
4274                                                    ;RHCS1 SHOULD HAVE DVA!RDY
4275                                                    ;=4200
4276                                                    ;BUT CONTAINED WHAT IS
4277                                                    ;GIVEN IN BAD RHCS1
4278  013754                            71$:
4279                                    ;CHECK THAT RHCS3 HAS 0
4280  013754  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD = 0
4281
4282  013762  017737  170144  001126    MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
4283  013770  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4284                                                    ;DATA WITH DATA READ FROM
4285                                                    ;RHCS3
4286  013776  001401                    BEQ     73$             ;BRANCH IF GOOD
4287  014000  104012                    ERROR   12
4288                                                    ;AFTER CLEARING THE RH
4289                                                    ;AND WRITING ALL ZEROS
4290                                                    ;INTO RHDB
4291                                                    ;THEN READING IT BACK
4292                                                    ;RHCS3 SHOULD HAVE 0
4293                                                    ;BUT CONTAINED WHAT IS
4294                                                    ;GIVEN IN BAD RHCS3
4295  014002                            73$:
4296                                    ;CHECK THAT RHBA HAS 0
4297  014002  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD =
4298
4299  014010  017737  170050  001126    MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
4300  014016  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4301                                                    ;DATA WITH DATA READ FROM
4302                                                    ;RHBA
4303  014024  001401                    BEQ     75$             ;BRANCH IF GOOD
4304  014026  104013                    ERROR   13
4305                                                    ;AFTER CLEARING THE RH
4306                                                    ;AND WRITING ALL ZEROS
4307                                                    ;INTO RHDB
4308                                                    ;THEN READING IT BACK
4309                                                    ;RHBA SHOULD HAVE 0
4310                                                    ;=
4311                                                    ;BUT CONTAINED WHAT IS
4312                                                    ;GIVEN IN BAD RHBA
4313  014030                            75$:
4314                                    ;CHECK THAT RHBAE HAS 0
4315  014030  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD =
4316
4317  014036  017737  170066  001126    MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
4318  014044  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4319                                                    ;DATA WITH DATA READ FROM
4320                                                    ;RHBAE
4321  014052  001401                    BEQ     77$             ;BRANCH IF GOOD
4322  014054  104014                    ERROR   14
4323                                                    ;AFTER CLEARING THE RH
4324                                                    ;AND WRITING ALL ZEROS
```

```
                                                        ;INTO RHDB
                                                        ;THEN READING IT BACK
                                                        ;RHBAE SHOULD HAVE 0
                                                        ;=
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHBAE
014056                                  77$:
                                        ;CHECK THAT RHWC HAS 0
014056  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =
014064  017737  167772  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
014072  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
014100  001401                          BEQ     79$             ;BRANCH IF GOOD
014102  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
014104                                  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
                                ;*********************************************************************
014104  000004                 TST12:  SCOPE
014106  012706  001000                  MOV     #STACK,SP       ;RESET STACK
014112  012737  000012  004656          MOV     #12,@#†STNM     ;SAVE TEST NUMBER

014120  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS2 HAS IR
014124  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
014132  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
4381
4382  014140  017737  167724  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4383  014146  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4384                                                                  ;DATA WITH DATA READ FROM
4385                                                                  ;RHCS2
4386  014154  001401                          BEQ     65$             ;BRANCH IF GOOD
4387  014156  104006                          ERROR   6
4388                                                                  ;AFTER SETTING CLR BIT #5
4389                                                                  ;IN RHCS2 TO INIT THE RH
4390                                                                  ;AND HAVING DONE NOTHING ELSE
4391                                                                  ;RHCS2 SHOULD HAVE IR
4392                                                                  ;=100
4393                                                                  ;TOGETHER WITH UNIT NUMBER
4394                                                                  ;BUT CONTAINED WHAT IS
4395                                                                  ;GIVEN IN BAD RHCS2
4396  014160                          65$:
4397
4398                                          ;WRITE ONE WORD OF ALL ONES INTO RHDB
4399  014160  012777  177777  167714          MOV     #177777,@RHDB   ;WRITE 1 IN RHDB
4400
4401                                          ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
4402  014166  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
4403  014170  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
4404  014172  000200                          OR                      ;RHCS2 REGISTER
4405                                                                  ;IF ERROR OCCURS HERE
4406                                                                  ;IT MEANS "OR" DID NOT
4407                                                                  ;SET FOR THE FULL COUNT
4408                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
4409                                                                  ;THIS TIME IS APPROXIMATELY
4410                                                                  ;LARGER THAN 500 MILLISECONDS
4411
4412
4413                                          ;CHECK THAT RHDB HAS 177777
4414  014174  012737  177777  001124          MOV     #177777,$GDDAT  ;GET GOOD = 0
4415
4416  014202  017737  167674  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
4417  014210  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4418                                                                  ;DATA WITH DATA READ FROM
4419                                                                  ;RHDB
4420  014216  001401                          BEQ     67$             ;BRANCH IF GOOD
4421  014220  104007                          ERROR   7
4422                                                                  ;AFTER CLEARING THE RH
4423                                                                  ;AND WRITING ALL ONES
4424                                                                  ;INTO RHDB
4425                                                                  ;THEN READING IT BACK
4426                                                                  ;RHDB SHOULD HAVE 177777
4427                                                                  ;BUT CONTAINED WHAT IS
4428                                                                  ;GIVEN IN BAD RHDB
4429  014222                          67$:
4430                                          ;CHECK THAT RHCS2 HAS IR
4431  014222  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
4432  014230  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4433
4434  014236  017737  167626  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4435  014244  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4436                                                                  ;DATA WITH DATA READ FROM
```

```
                                                                ;RHCS2
014252  001401                          BEQ     69$             ;BRANCH IF GOOD
014254  104010                          ERROR   10
                                                                ;AFTER CLEARING THE RH
                                                                ;AND WRITING ALL ONES
                                                                ;INTO RHDB
                                                                ;THEN READING IT BACK
                                                                ;RHCS2 SHOULD HAVE IR
                                                                ;=100
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
014256                          69$:
                                ;CHECK THAT RHCS3 HAS 0
014256  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =

014264  017737  167642  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
014272  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
014300  001401                          BEQ     71$             ;BRANCH IF GOOD
014302  104012                          ERROR   12
                                                                ;AFTER CLEARING THE RH
                                                                ;AND WRITING ALL ONES
                                                                ;INTO RHDB
                                                                ;THEN READING IT BACK
                                                                ;RHCS3 SHOULD HAVE 0
                                                                ;=
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
014304                          71$:
                                ;CHECK THAT RHBA HAS 0
014304  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =

014312  017737  167546  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
014320  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBA
014326  001401                          BEQ     73$             ;BRANCH IF GOOD
014330  104013                          ERROR   13
                                                                ;AFTER CLEARING THE RH
                                                                ;AND WRITING ALL ONES
                                                                ;INTO RHDB
                                                                ;THEN READING IT BACK
                                                                ;RHBA SHOULD HAVE 0
                                                                ;=
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
014332                          73$:
                                ;CHECK THAT RHBAE HAS 0
014332  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =

014340  017737  167564  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
014346  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
```

```
4493  014354  001401                          BEQ     75$             ;BRANCH IF GOOD
4494  014356  104014                          ERROR   14
4495                                                                  ;AFTER CLEARING THE RH
4496                                                                  ;AND WRITING ALL ONES
4497                                                                  ;INTO RHDB
4498                                                                  ;THEN READING IT BACK
4499                                                                  ;RHBAE SHOULD HAVE 0
4500                                                                  ;=
4501                                                                  ;BUT CONTAINED WHAT IS
4502                                                                  ;GIVEN IN BAD RHBAE
4503  014360                          75$:
4504                                          ;CHECK THAT RHWC HAS 0
4505  014360  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD =
4506
4507  014366  017737  167470  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
4508  014374  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4509                                                                  ;DATA WITH DATA READ FROM
4510                                                                  ;RHWC
4511  014402  001401                          BEQ     77$             ;BRANCH IF GOOD
4512  014404  104015                          ERROR   15
4513                                                                  ;AFTER CLEARING THE RH
4514                                                                  ;AND WRITING ALL ONES
4515                                                                  ;INTO RHDB
4516                                                                  ;THEN READING IT BACK
4517                                                                  ;RHWC SHOULD HAVE 0
4518                                                                  ;=
4519                                                                  ;BUT CONTAINED WHAT IS
4520                                                                  ;GIVEN IN BAD RHWC
4521  014406                          77$:
4522
4523
4524                                  ;;*****************************************************************
4525                                  ;*TEST 13       SILO TEST 3 (TWO WORD WRITE)
4526
4527                                  ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
4528                                  ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
4529                                  ;*      TOGETHER WITH UNIT NUMBER
4530                                  ;*      ONE WORD = 52525 IS WRITTEN INTO RHDB
4531                                  ;*      RHCS2 IS CHECKED TO HAVE "IR" AND UNIT NUMBER
4532                                  ;*      A SECOND WORD = 12525 IS WRITTEN INTO RHDB
4533                                  ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
4534                                  ;*      INSTRUCTION CALLED "WAT"
4535                                  ;*      RHDB IS READ AND CHECKED TO CONTAIN 52525
4536                                  ;*      RHCS2 IS CHECKED TO HAVE "IR"  "OR" AND UNIT NUMBER
4537                                  ;*      RHDB IS READ A SECOND TIME AND CHECKED TO
4538                                  ;*      CONTAIN 125252
4539                                  ;*      RHCS2 IS CHECKED TO HAVE "IR" AND UNIT NUMBER
4540                                  ;*      THEN ALL REGISTERS RHCS1, RHCS3, RHBA, RHBAE, RHWC
4541                                  ;*      ARE CHECKED TO HAVE APPROPIATE VALUE
4542                                  ;*
4543                                  ;*
4544                                  ;*****************************************************************
4545  014406  000004                  TST13:  SCOPE
4546  014410  012706  001000                  MOV     #STACK,SP       ;RESET STACK
4547  014414  012737  000013  004656          MOV     #13,@#TSTNM     ;SAVE TEST NUMBER
4548
```

```
4549  014422  004737  040306                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
4550                                                                      ;SETUP UNIT NUBER
4551                                                                      ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
4552
4553                                              ;CHECK THAT RHCS2 HAS IR
4554  014426  012737  000100  001124              MOV     #IR,$GDDAT      ;GET GOOD = 100
4555  014434  053737  005010  001124              BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4556
4557  014442  017737  167422  001126              MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4558  014450  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4559                                                                      ;DATA WITH DATA READ FROM
4560                                                                      ;RHCS2
4561  014456  001401                              BEQ     65$             ;BRANCH IF GOOD
4562  014460  104006                              ERROR   6
4563                                                                      ;AFTER SETTING "CLR" BIT #5
4564                                                                      ;IN RHCS2 TO INIT THE RH
4565                                                                      ;AND HAVING DONE NOTHING ELSE
4566
4567                                                                      ;RHCS2 SHOULD HAVE IR
4568                                                                      ;=100
4569                                                                      ;TOGETHER WITH UNIT NUMBER
4570                                                                      ;BUT CONTAINED WHAT IS
4571                                                                      ;GIVEN IN BAD RHCS2
4572  014462                            65$:
4573
4574                                              ;WRITE ONE WORD = 52525 INTO RHDB
4575  014462  012777  052525  167412              MOV     #52525,@RHDB    ;WRITE IN RHDB
4576
4577                                              ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
4578  014470  104411                              WAT                     ;TRAP TO WAIT.T SUBROUTINE
4579  014472  004070                              RHCS2                   ;AND WAIT FOR OR BIT IN
4580  014474  000200                              OR                      ;RHCS2 REGISTER
4581                                                                      ;IF ERROR OCCURS HERE
4582                                                                      ;IT MEANS "OR" DID NOT
4583                                                                      ;SET FOR THE FULL COUNT
4584                                                                      ;DOWN OF THE WAIT.T SUBROUTINE
4585                                                                      ;THIS TIME IS APPROXIMATELY
4586                                                                      ;LARGER THAN 500 MILLISECONDS
4587
4588
4589                                              ;CHECK THAT RHCS2 HAS IR!OR
4590  014476  012737  000300  001124              MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
4591  014504  053737  005010  001124              BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4592
4593  014512  017737  167352  001126              MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4594  014520  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4595                                                                      ;DATA WITH DATA READ FROM
4596                                                                      ;RHCS2
4597  014526  001401                              BEQ     67$             ;BRANCH IF GOOD
4598  014530  104016                              ERROR   16
4599                                                                      ;AFTER SETTING "CLR" BIT #5
4600                                                                      ;IN RHCS2 TO INIT THE RH
4601                                                                      ;THEN WRITING 52525 INTO
4602                                                                      ;RHDB
4603                                                                      ;RHCS2 SHOULD HAVE IR!OR
4604                                                                      ;=300
```

```
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
014532                                  67$:

                                        ;WRITE A SECOND WORD = 125252 INTO RHDB
014532  012777  125252  167342          MOV     #125252,@RHDB   ;WRITE 125252 INTO RHDB

                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
014540  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
014542  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
014544  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
014546  012737  052525  001124          MOV     #52525,$GDDAT   ;GET GOOD = 0

014554  017737  167322  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
014562  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
014570  001401                          BEQ     69$             ;BRANCH IF GOOD
014572  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
014574                                  69$:
                                        ;CHECK THAT RHCS2 HAS IR!OR
014574  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
014602  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

014610  017737  167254  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
014616  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
014624  001401                          BEQ     71$             ;BRANCH IF GOOD
014626  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
014630                                  71$:
```

```
4661
4662
4663                                                    ;READ RHDB A SECOND TIME
4664                                                    ;CHECK THAT RHDB HAS 125252
4665  014630  012737  125252  001124                  MOV     #125252,$GDDAT  ;GET GOOD = 0
4666
4667  014636  017737  167240  001126                  MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
4668  014644  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4669                                                                          ;DATA WITH DATA READ FROM
4670                                                                          ;RHDB
4671  014652  001401                                  BEQ     73$             ;BRANCH IF GOOD
4672  014654  104021                                  ERROR   21
4673                                                                          ;AFTER CLEARING RH THEN
4674                                                                          ;WRITING TWO WORDS INTO
4675                                                                          ;RHDB 52525 AND 125252
4676                                                                          ;THEN READING RHDB THE
4677                                                                          ;SECOND TIME
4678                                                                          ;RHDB SHOULD HAVE 125252
4679                                                                          ;BUT CONTAINED WHAT IS
4680                                                                          ;GIVEN IN BAD RHDB
4681  014656                                  73$:
4682                                                    ;CHECK THAT RHCS2 HAS IR
4683  014656  012737  000100  001124                  MOV     #IR,$GDDAT      ;GET GOOD = 100
4684  014664  053737  005010  001124                  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4685
4686  014672  017737  167172  001126                  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4687  014700  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4688                                                                          ;DATA WITH DATA READ FROM
4689                                                                          ;RHCS2
4690  014706  001401                                  BEQ     75$             ;BRANCH IF GOOD
4691  014710  104022                                  ERROR   22
4692                                                                          ;AFTER CLEARING RH THEN
4693                                                                          ;WRITING TWO WORDS INTO
4694                                                                          ;RHDB 52525 AND 125252
4695                                                                          ;THEN READING RHDB THE
4696                                                                          ;SECOND TIME
4697                                                                          ;RHCS2 SHOULD HAVE IR
4698                                                                          ;=100
4699                                                                          ;TOGETHER WITH UNIT NUMBER
4700                                                                          ;BUT CONTAINED WHAT IS
4701                                                                          ;GIVEN IN BAD RHCS2
4702  014712                                  75$:
4703                                                    ;CHECK THAT RHCS1 HAS DVA!RDY
4704  014712  012737  004200  001124                  MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
4705
4706  014720  017737  167134  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
4707  014726  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4708                                                                          ;DATA WITH DATA READ FROM
4709                                                                          ;RHCS1
4710  014734  001401                                  BEQ     77$             ;BRANCH IF GOOD
4711  014736  104141                                  ERROR   141
4712                                                                          ;AFTER CLEARING RH THEN
4713                                                                          ;WRITING TWO WORDS INTO
4714                                                                          ;RHDB 52525 AND 125252
4715                                                                          ;THEN READING RHDB THE
4716                                                                          ;SECOND TIME
```

```
                                                        ;RHCS1 SHOULD HAVE DVA!RDY
                                                        ;=4200
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS1
014740                                  77$:
                                        ;CHECK THAT RHCS3 HAS 0
014740  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

014746  017737  167160  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
014754  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
014762  001401                          BEQ     79$             ;BRANCH IF GOOD
014764  104142                          ERROR   142
                                                        ;AFTER CLEARING RH THEN
                                                        ;WRITING TWO WORDS INTO
                                                        ;RHDB 52525 AND 125252
                                                        ;THEN READING RHDB THE
                                                        ;SECOND TIME
                                                        ;RHCS3 SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS3
014766                                  79$:
                                        ;CHECK THAT RHBA HAS 0
014766  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

014774  017737  167064  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
015002  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBA
015010  001401                          BEQ     81$             ;BRANCH IF GOOD
015012  104143                          ERROR   143
                                                        ;AFTER CLEARING RH THEN
                                                        ;WRITING TWO WORDS INTO
                                                        ;RHDB 52525 AND 125252
                                                        ;THEN READING RHDB THE
                                                        ;SECOND TIME
                                                        ;RHBA SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHBA
015014                                  81$:
                                        ;CHECK THAT RHBAE HAS 0
015014  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

015022  017737  167102  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
015030  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBAE
015036  001401                          BEQ     83$             ;BRANCH IF GOOD
015040  104144                          ERROR   144
                                                        ;AFTER CLEARING RH THEN
                                                        ;WRITING TWO WORDS INTO
                                                        ;RHDB 52525 AND 125252
                                                        ;THEN READING RHDB THE
                                                        ;SECOND TIME
                                                        ;RHBAE SHOULD HAVE 0
```

```
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHBAE
015042                                  83$:
                                        ;CHECK THAT RHWC HAS 0
015042  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

015050  017737  167006  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
015056  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
015064  001401                          BEQ     85$             ;BRANCH IF GOOD
015066  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
015070                                  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
                                        ;;**************************************************************
015070  000004                        ↑ST14:  SCOPE
015072  012706  001000                  MOV     #STACK,SP       ;RESET STACK
015076  012737  000014  004656          MOV     #14,@#↑STNM     ;SAVE TEST NUMBER

015104  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS2 HAS IR
015110  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
015116  053737  005010  001124          BIS     UNI↑,$GDDAT     ;INCLUDE UNIT NUMBER

015124  017737  166740  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
015132  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
```

```
4829  015140  001401                          BEQ     65$             ;BRANCH IF GOOD
4830  015142  104006                          ERROR   6
4831                                                                  ;AFTER SETTING "CLR" BIT #5
4832                                                                  ;IN RHCS2 TO INIT THE RH
4833                                                                  ;AND HAVING DONE NOTHING ELSE
4834                                                                  ;RHCS2 SHOULD HAVE IR
4835                                                                  ;=100
4836                                                                  ;TOGETHER WITH UNIT NUMBER
4837                                                                  ;BUT CONTAINED WHAT IS
4838                                                                  ;GIVEN IN BAD RHCS2
4839  015144                          65$:
4840
4841                                          ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
4842                                          ;THIS IS A COUNT PATTERN 0 THRU 7
4843  015144  012702  000010                 MOV     #8.,R2          ;8 NUMBERS TO BE WRITTEN
4844  015150  005001                          CLR     R1              ;FIRST WORD TO BE WRITTEN = 0
4845  015152  010177  166724          1$:    MOV     R1,@RHDB        ;WRITE INTO RHDB - A COUNT
4846  015156  005201                          INC     R1              ;GET NEXT WORD
4847  015160  005302                          DEC     R2              ;COUNT TO 8
4848  015162  001373                          BNE     1$              ;BRANCH IF 8 NOT DONE
4849
4850                                          ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
4851  015164  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
4852  015166  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
4853  015170  000200                          OR                      ;RHCS2 REGISTER
4854                                                                  ;IF ERROR OCCURS HERE
4855                                                                  ;IT MEANS "OR" DID NOT
4856                                                                  ;SET FOR THE FULL COUNT
4857                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
4858                                                                  ;THIS TIME IS APPROXIMATELY
4859                                                                  ;LARGER THAN 500 MILLISECONDS
4860
4861                                          ;CHECK THAT RHCS2 HAS OR
4862  015172  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
4863  015200  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
4864
4865  015206  017737  166656  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
4866  015214  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
4867                                                                  ;DATA WITH DATA READ FROM
4868                                                                  ;RHCS2
4869  015222  001401                          BEQ     67$             ;BRANCH IF GOOD
4870  015224  104023                          ERROR   23
4871                                                                  ;AFTER SETTING "CLR" BIT #5
4872                                                                  ;IN RHCS2 TO INIT THE RH
4873                                                                  ;AND HAVING WRITTEN A
4874                                                                  ;COUNT PATTERN 0 THRU 7
4875                                                                  ;INTO RHDB TO FILL IT
4876                                                                  ;RHCS2 SHOULD HAVE OR
4877                                                                  ;=200
4878                                                                  ;TOGETHER WITH UNIT NUMBER
4879                                                                  ;BUT CONTAINED WHAT IS
4880                                                                  ;GIVEN IN BAD RHCS2
4881  015226                          67$:
4882
4883
4884
```

```
 4885  015226  012737  000001  004664          MOV     #1,SILONM          ;FIRST WORD TO BE READ
 4886                                          ;CHECK THAT RHDB HAS 0
 4887  015234  012737  000000  001124          MOV     #0,$GDDAT          ;GET GOOD = 0
 4888
 4889  015242  017737  166634  001126          MOV     @RHDB,$BDDAT       ;READ RHDB FOR COMPARISON
 4890  015250  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
 4891                                                                     ;DATA WITH DATA READ FROM
 4892                                                                     ;RHDB
 4893  015256  001401                          BEQ     69$                ;BRANCH IF GOOD
 4894  015260  104024                          ERROR   24
 4895                                                                     ;AFTER SETTING "CLR" BIT #5
 4896                                                                     ;IN RHCS2 TO INIT THE RH
 4897                                                                     ;AND WRITING A
 4898                                                                     ;COUNT PATTERN 0 THRU 7
 4899                                                                     ;INTO RHDB TO FILL IT
 4900                                                                     ;THEN ON READING RHDB THE
 4901                                                                     ;FIRST TIME
 4902                                                                     ;RHDB SHOULD HAVE 0
 4903                                                                     ;BUT CONTAINED WHAT IS
 4904                                                                     ;GIVEN IN BAD RHDB
 4905  015262                          69$:
 4906
 4907  015262  012737  000002  004664          MOV     #2,SILONM          ;SECOND WORD TO BE READ
 4908                                          ;CHECK THAT RHDB HAS 1
 4909  015270  012737  000001  001124          MOV     #1,$GDDAT          ;GET GOOD = 0
 4910
 4911  015276  017737  166600  001126          MOV     @RHDB,$BDDAT       ;READ RHDB FOR COMPARISON
 4912  015304  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
 4913                                                                     ;DATA WITH DATA READ FROM
 4914                                                                     ;RHDB
 4915  015312  001401                          BEQ     71$                ;BRANCH IF GOOD
 4916  015314  104024                          ERROR   24
 4917                                                                     ;AFTER SETTING "CLR" BIT #5
 4918                                                                     ;IN RHCS2 TO INIT THE RH
 4919                                                                     ;AND WRITING A
 4920                                                                     ;COUNT PATTERN 0 THRU 7
 4921                                                                     ;INTO RHDB TO FILL IT
 4922                                                                     ;THEN ON READING RHDB THE
 4923                                                                     ;SECOND TIME
 4924                                                                     ;RHDB SHOULD HAVE 1
 4925                                                                     ;BUT CONTAINED WHAT IS
 4926                                                                     ;GIVEN IN BAD RHDB
 4927  015316                          71$:
 4928
 4929  015316  012737  000003  004664          MOV     #3,SILONM          ;THIRD WORD TO BE READ
 4930                                          ;CHECK THAT RHDB HAS 2
 4931  015324  012737  000002  001124          MOV     #2,$GDDAT          ;GET GOOD = 0
 4932
 4933  015332  017737  166544  001126          MOV     @RHDB,$BDDAT       ;READ RHDB FOR COMPARISON
 4934  015340  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
 4935                                                                     ;DATA WITH DATA READ FROM
 4936                                                                     ;RHDB
 4937  015346  001401                          BEQ     73$                ;BRANCH IF GOOD
 4938  015350  104024                          ERROR   24
 4939                                                                     ;AFTER SETTING "CLR" BIT #5
 4940                                                                     ;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
015352                                  73$:
015352  012737  000004  004664                MOV     #4,SILONM       ;FOURTH WORD TO BE READ
                                              ;CHECK THAT RHDB HAS 3
015360  012737  000003  001124                MOV     #3,$GDDAT       ;GET GOOD = 0
015366  017737  166510  001126                MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015374  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHDB
015402  001401                              BEQ     75$             ;BRANCH IF GOOD
015404  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
015406                                  75$:
015406  012737  000005  004664                MOV     #5,SILONM       ;FIFTH WORD TO BE READ
                                              ;CHECK THAT RHDB HAS 4
015414  012737  000004  001124                MOV     #4,$GDDAT       ;GET GOOD = 0
015422  017737  166454  001126                MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
015430  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHDB
015436  001401                              BEQ     77$             ;BRANCH IF GOOD
015440  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
015442                                  77$:
015442  012737  000006  004664                MOV     #6,SILONM       ;SIXTH WORD TO BE READ
                                              ;CHECK THAT RHDB HAS 5
```

```
 4997  015450  012737  000005  001124          MOV     #5,$GDDAT       ;GET GOOD = 0
 4998
 4999  015456  017737  166420  001126          MOV     @RHDB,$EDDAT    ;READ RHDB FOR COMPARISON
 5000  015464  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 5001                                                                  ;DATA WITH DATA READ FROM
 5002                                                                  ;RHDB
 5003  015472  001401                          BEQ     79$             ;BRANCH IF GOOD
 5004  015474  104024                          ERROR   24
 5005                                                                  ;AFTER SETTING "CLR" BIT #5
 5006                                                                  ;IN RHCS2 TO INIT THE RH
 5007                                                                  ;AND WRITING A
 5008                                                                  ;COUNT PATTERN 0 THRU 7
 5009                                                                  ;INTO RHDB TO FILL IT
 5010                                                                  ;THEN ON READING RHDB THE
 5011                                                                  ;SIXTH TIME
 5012                                                                  ;RHDB SHOULD HAVE 5
 5013                                                                  ;BUT CONTAINED WHAT IS
 5014                                                                  ;GIVEN IN BAD RHDB
 5015  015476                          79$:
 5016
 5017  015476  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
 5018                                          ;CHECK THAT RHDB HAS 6
 5019  015504  012737  000006  001124          MOV     #6,$GDDAT       ;GET GOOD = 0
 5020
 5021  015512  017737  166364  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
 5022  015520  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 5023                                                                  ;DATA WITH DATA READ FROM
 5024                                                                  ;RHDB
 5025  015526  001401                          BEQ     81$             ;BRANCH IF GOOD
 5026  015530  104024                          ERROR   24
 5027                                                                  ;AFTER SETTING "CLR" BIT #5
 5028                                                                  ;IN RHCS2 TO INIT THE RH
 5029                                                                  ;AND WRITING A
 5030                                                                  ;COUNT PATTERN 0 THRU 7
 5031                                                                  ;INTO RHDB TO FILL IT
 5032                                                                  ;THEN ON READING RHDB THE
 5033                                                                  ;SEVENTH TIME
 5034                                                                  ;RHDB SHOULD HAVE 6
 5035                                                                  ;BUT CONTAINED WHAT IS
 5036                                                                  ;GIVEN IN BAD RHDB
 5037  015532                          81$:
 5038
 5039  015532  012737  000010  004664          MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
 5040                                          ;CHECK THAT RHDB HAS 7
 5041  015540  012737  000007  001124          MOV     #7,$GDDAT       ;GET GOOD = 0
 5042
 5043  015546  017737  166330  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
 5044  015554  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 5045                                                                  ;DATA WITH DATA READ FROM
 5046                                                                  ;RHDB
 5047  015562  001401                          BEQ     83$             ;BRANCH IF GOOD
 5048  015564  104024                          ERROR   24
 5049                                                                  ;AFTER SETTING "CLR" BIT #5
 5050                                                                  ;IN RHCS2 TO INIT THE RH
 5051                                                                  ;AND WRITING A
 5052                                                                  ;COUNT PATTERN 0 THRU 7
```

```
                                                                ;INTO RHDB TO FILL IT
                                                                ;THEN ON READING RHDB THE
                                                                ;EIGHTH TIME
                                                                ;RHDB SHOULD HAVE 7
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
015566                                  83$:

                                        ;CHECK THAT RHCS2 HAS IR
015566  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
015574  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

015602  017737  166262  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
015610  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
015616  001401                          BEQ     85$             ;BRANCH IF GOOD
015620  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
015622                                  85$:
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
015622  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

015630  017737  166224  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
015636  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
015644  001401                          BEQ     87$             ;BRANCH IF GOOD
015646  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
015650                                  87$:
                                        ;CHECK THAT RHCS3 HAS 0
015650  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
```

```
5109  015656  017737  166250  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
5110  015664  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5111                                                                  ;DATA WITH DATA READ FROM
5112                                                                  ;RHCS3
5113  015672  001401                          BEQ     89$             ;BRANCH IF GOOD
5114  015674  104027                          ERROR   27
5115                                                                  ;AFTER SETTING "CLR" BIT #5
5116                                                                  ;IN RHCS2 TO INIT THE RH
5117                                                                  ;AND WRITING A
5118                                                                  ;COUNT PATTERN 0 THRU 7
5119                                                                  ;INTO RHDB TO FILL IT
5120                                                                  ;THEN ON READING RHDB THE
5121                                                                  ;TOTAL 8 TIMES
5122                                                                  ;RHCS3 SHOULD HAVE 0
5123                                                                  ;BUT CONTAINED WHAT IS
5124                                                                  ;GIVEN IN BAD RHCS3
5125  015676                          89$:
5126                                          ;CHECK THAT RHBA HAS 0
5127  015676  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5128
5129  015704  017737  166154  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
5130  015712  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5131                                                                  ;DATA WITH DATA READ FROM
5132                                                                  ;RHBA
5133  015720  001401                          BEQ     91$             ;BRANCH IF GOOD
5134  015722  104030                          ERROR   30
5135                                                                  ;AFTER SETTING "CLR" BIT #5
5136                                                                  ;IN RHCS2 TO INIT THE RH
5137                                                                  ;AND WRITING A
5138                                                                  ;COUNT PATTERN 0 THRU 7
5139                                                                  ;INTO RHDB TO FILL IT
5140                                                                  ;THEN ON READING RHDB THE
5141                                                                  ;TOTAL 8 TIMES
5142                                                                  ;RHBA SHOULD HAVE 0
5143                                                                  ;BUT CONTAINED WHAT IS
5144                                                                  ;GIVEN IN BAD RHBA
5145  015724                          91$:
5146                                          ;CHECK THAT RHBAE HAS 0
5147  015724  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5148
5149  015732  017737  166172  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
5150  015740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5151                                                                  ;DATA WITH DATA READ FROM
5152                                                                  ;RHBAE
5153  015746  001401                          BEQ     93$             ;BRANCH IF GOOD
5154  015750  104031                          ERROR   31
5155                                                                  ;AFTER SETTING "CLR" BIT #5
5156                                                                  ;IN RHCS2 TO INIT THE RH
5157                                                                  ;AND WRITING A
5158                                                                  ;COUNT PATTERN 0 THRU 7
5159                                                                  ;INTO RHDB TO FILL IT
5160                                                                  ;THEN ON READING RHDB THE
5161                                                                  ;TOTAL 8 TIMES
5162                                                                  ;RHBAE SHOULD HAVE 0
5163                                                                  ;BUT CONTAINED WHAT IS
5164                                                                  ;GIVEN IN BAD RHBAE
```

```
5165  015752                          93$:
5166                                          ;CHECK THAT RHWC HAS 0
5167  015752  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5168
5169  015760  017737  166076  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
5170  015766  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5171                                                                  ;DATA WITH DATA READ FROM
5172                                                                  ;RHWC
5173  015774  001401                          BEQ     95$             ;BRANCH IF GOOD
5174  015776  104032                          ERROR   32
5175                                                                  ;AFTER SETTING "CLR" BIT #5
5176                                                                  ;IN RHCS2 TO INIT THE RH
5177                                                                  ;AND WRITING A
5178                                                                  ;COUNT PATTERN 0 THRU 7
5179                                                                  ;INTO RHDB TO FILL IT
5180                                                                  ;THEN ON READING RHDB THE
5181                                                                  ;TOTAL 8 TIMES
5182                                                                  ;RHWC SHOULD HAVE 0
5183                                                                  ;BUT CONTAINED WHAT IS
5184                                                                  ;GIVEN IN BAD RHWC
5185  016000                          95$:
5186
5187
5188                                  ;;*****************************************************************
5189                                  ;*TEST 15       SILO TEST 5 (FLOATING ONES)
5190
5191                                  ;*      AFTER ARH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
5192                                  ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
5193                                  ;*      TOGETHER WITH UNIT NUMBER
5194                                  ;*      EIGHT WORDS, A PATTERN OF FLOATING ONES (1,2,4,10,20,40,100,200)
5195                                  ;*      IS WRITTEN INTO RHDB
5196                                  ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
5197                                  ;*      INSTRUCTION CALLED "WAT"
5198                                  ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
5199                                  ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
5200                                  ;*      THEN RHDB IS READ AND COMPARED TO HAVE THE
5201                                  ;*      RIGHT VALUE AFTER EACH OF THE EIGHT READS.
5202                                  ;*      THEN RHCS2 IS CHECKED TO HAVE "IR"
5203                                  ;*      AND UNIT NUMBER
5204                                  ;*      THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
5205                                  ;*      CHECKED TO HAVE THE APPROPIATE VALUE
5206                                  ;*****************************************************************
5207  016000  000004                  ST15:   SCOPE
5208  016002  012706  001000                  MOV     #STACK,SP       ;RESET STACK
5209  016006  012737  000015  004656          MOV     #15,@#TSTNM     ;SAVE TEST NUMBER
5210
5211  016014  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
5212                                                                  ;SETUP UNIT NUBER
5213                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
5214
5215
5216                                          ;CHECK THAT RHCS2 HAS IR
5217  016020  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
5218  016026  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
5219
5220  016034  017737  166030  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
```

```
5221  016042  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5222                                                                  ;DATA WITH DATA READ FROM
5223                                                                  ;RHCS2
5224  016050  001401                          BEQ     65$             ;BRANCH IF GOOD
5225  016052  104006                          ERROR   6
5226                                                                  ;AFTER SETTING "CLR" BIT #5
5227                                                                  ;IN RHCS2 TO INIT THE RH
5228                                                                  ;AND HAVING DONE NOTHING
5229                                                                  ;ELSE
5230                                                                  ;RHCS2 SHOULD HAVE IR
5231                                                                  ;=100
5232                                                                  ;TOGETHER WITH UNIT NUMBER
5233                                                                  ;BUT CONTAINED WHAT IS
5234                                                                  ;GIVEN IN BAD RHCS2
5235  016054                          65$:
5236
5237                                          ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
5238                                          ;DATA IS 1,2,4,10,20,40,100,200
5239  016054  012701  000001                 MOV     #1,R1           ;GETTING READY TO FLOAT ONES
5240  016060  012702  000010                 MOV     #8.,R2          ;COUNTER -8 DATA WORDS
5241  016064  010177  166012          1$:    MOV     R1,@RHDB        ;WRITE INTO RHDB
5242  016070  006301                 ASL     R1              ;SHIFT 1 ONE POSITION LEFT
5243  016072  005302                 DEC     R2              ;COUNT TO 8
5244  016074  001373                 BNE     1$              ;BRANCH IF 8 NOT DONE
5245                                  ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
5246  016076  104411                 WAT                     ;TRAP TO WAIT.T SUBROUTINE
5247  016100  004070                 RHCS2                   ;AND WAIT FOR OR BIT IN
5248  016102  000200                 OR                      ;RHCS2 REGISTER
5249                                                          ;IF ERROR OCCURS HERE
5250                                                          ;IT MEANS "OR" DID NOT
5251                                                          ;SET FOR THE FULL COUNT
5252                                                          ;DOWN OF THE WAIT.T SUBROUTINE
5253                                                          ;THIS TIME IS APPROXIMATELY
5254                                                          ;LARGER THAN 500 MILLISECONDS
5255
5256
5257                                  ;CHECK THAT RHCS2 HAS OR
5258  016104  012737  000200  001124  MOV     #OR,$GDDAT      ;GET GOOD = 200
5259  016112  053737  005010  001124  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
5260
5261  016120  017737  165744  001126  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
5262  016126  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5263                                                          ;DATA WITH DATA READ FROM
5264                                                          ;RHCS2
5265  016134  001401                  BEQ     67$             ;BRANCH IF GOOD
5266  016136  104023                  ERROR   23
5267                                                          ;AFTER SETTING "CLR" BIT #5
5268                                                          ;IN RHCS2 TO INIT THE RH
5269                                                          ;AND HAVING WRITTEN A
5270                                                          ;FLOATING ONES PATTERN
5271                                                          ;1,2,4,10,20,40,100,200
5272                                                          ;INTO RHDB TO FILL IT
5273                                                          ;RHCS2 SHOULD HAVE OR
5274                                                          ;=200
5275                                                          ;TOGETHER WITH UNIT NUMBER
5276                                                          ;BUT CONTAINED WHAT IS
```

```
5277                                                    ;GIVEN IN BAD RHCS2
5278  016140                        67$:
5279
5280
```

```
5281
5282  016140  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
5283                                          ;CHECK THAT RHDB HAS 1
5284  016146  012737  000001  001124          MOV     #1,$GDDAT       ;GET GOOD = 0
5285
5286  016154  017737  165722  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5287  016162  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5288                                                                  ;DATA WITH DATA READ FROM
5289                                                                  ;RHDB
5290  016170  001401                          BEQ     69$             ;BRANCH IF GOOD
5291  016172  104024                          ERROR   24
5292                                                                  ;AFTER SETTING "CLR" BIT #5
5293                                                                  ;IN RHCS2 TO INIT THE RH
5294                                                                  ;AND WRITING A PATTERN OF
5295                                                                  ;FLOATING ONES (WHICH IS
5296                                                                  ;1,2,4,10,20,40,100,200)
5297                                                                  ;INTO RHDB TO FILL IT
5298                                                                  ;THEN ON READING RHDB THE
5299                                                                  ;FIRST TIME
5300                                                                  ;RHDB SHOULD HAVE 1
5301                                                                  ;BUT CONTAINED WHAT IS
5302                                                                  ;GIVEN IN BAD RHDB
5303  016174                          69$:
5304
5305
5306  016174  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
5307                                          ;CHECK THAT RHDB HAS 2
5308  016202  012737  000002  001124          MOV     #2,$GDDAT       ;GET GOOD = 0
5309
5310  016210  017737  165666  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5311  016216  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5312                                                                  ;DATA WITH DATA READ FROM
5313                                                                  ;RHDB
5314  016224  001401                          BEQ     71$             ;BRANCH IF GOOD
5315  016226  104024                          ERROR   24
5316                                                                  ;AFTER SETTING "CLR" BIT #5
5317                                                                  ;IN RHCS2 TO INIT THE RH
5318                                                                  ;AND WRITING A PATTERN OF
5319                                                                  ;FLOATING ONES (WHICH IS
5320                                                                  ;1,2,4,10,20,40,100,200)
5321                                                                  ;INTO RHDB TO FILL IT
5322                                                                  ;THEN ON READING RHDB THE
5323                                                                  ;SECOND TIME
5324                                                                  ;RHDB SHOULD HAVE 2
5325                                                                  ;BUT CONTAINED WHAT IS
5326                                                                  ;GIVEN IN BAD RHDB
5327  016230                          71$:
5328
5329
5330  016230  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
5331                                          ;CHECK THAT RHDB HAS 4
5332  016236  012737  000004  001124          MOV     #4,$GDDAT       ;GET GOOD = 0
5333
5334  016244  017737  165632  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5335  016252  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5336                                                                  ;DATA WITH DATA READ FROM
```

```
5337
5338  016260  001401                          BEQ     73$                 ;RHDB
5339  016262  104024                          ERROR   24                  ;BRANCH IF GOOD
5340                                                                      ;AFTER SETTING "CLR" BIT #5
5341                                                                      ;IN RHCS2 TO INIT THE RH
5342                                                                      ;AND WRITING A PATTERN OF
5343                                                                      ;FLOATING ONES (WHICH IS
5344                                                                      ;1,2,4,10,20,40,100,200)
5345                                                                      ;INTO RHDB TO FILL IT
5346                                                                      ;THEN ON READING RHDB THE
5347                                                                      ;THIRD TIME
5348                                                                      ;RHDB SHOULD HAVE 4
5349                                                                      ;BUT CONTAINED WHAT IS
5350                                                                      ;GIVEN IN BAD RHDB
5351  016264                          73$:
5352
5353
5354  016264  012737  000004  004664          MOV     #4,SILONM           ;FOURTH WORD TO BE READ
5355                                          ;CHECK THAT RHDB HAS 10
5356  016272  012737  000010  001124          MOV     #10,$GDDAT          ;GET GOOD = 0
5357
5358  016300  017737  165576  001126          MOV     @RHDB,$BDDAT        ;READ RHDB FOR COMPARISON
5359  016306  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;COMPARE EXPECTED
5360                                                                      ;DATA WITH DATA READ FROM
5361                                                                      ;RHDB
5362  016314  001401                          BEQ     75$                 ;BRANCH IF GOOD
5363  016316  104024                          ERROR   24
5364                                                                      ;AFTER SETTING "CLR" BIT #5
5365                                                                      ;IN RHCS2 TO INIT THE RH
5366                                                                      ;AND WRITING A PATTERN OF
5367                                                                      ;FLOATING ONES (WHICH IS
5368                                                                      ;1,2,4,10,20,40,100,200)
5369                                                                      ;INTO RHDB TO FILL IT
5370                                                                      ;THEN ON READING RHDB THE
5371                                                                      ;FOURTH TIME
5372                                                                      ;RHDB SHOULD HAVE 10
5373                                                                      ;BUT CONTAINED WHAT IS
5374                                                                      ;GIVEN IN BAD RHDB
5375  016320                          75$:
5376
5377
5378  016320  012737  000005  004664          MOV     #5,SILONM           ;FIFTH WORD TO BE READ
5379                                          ;CHECK THAT RHDB HAS 20
5380  016326  012737  000020  001124          MOV     #20,$GDDAT          ;GET GOOD = 0
5381
5382  016334  017737  165542  001126          MOV     @RHDB,$BDDAT        ;READ RHDB FOR COMPARISON
5383  016342  023737  001124  001126          CMP     $GDDAT,$BDDAT       ;COMPARE EXPECTED
5384                                                                      ;DATA WITH DATA READ FROM
5385                                                                      ;RHDB
5386  016350  001401                          BEQ     77$                 ;BRANCH IF GOOD
5387  016352  104024                          ERROR   24
5388                                                                      ;AFTER SETTING "CLR" BIT #5
5389                                                                      ;IN RHCS2 TO INIT THE RH
5390                                                                      ;AND WRITING A PATTERN OF
5391                                                                      ;FLOATING ONES (WHICH IS
5392                                                                      ;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
016354                                  77$:

016354  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 40
016362  012737  000040  001124          MOV     #40,$GDDAT      ;GET GOOD = 0

016370  017737  165506  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
016376  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
016404  001401                          BEQ     79$             ;BRANCH IF GOOD
016406  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
016410                                  79$:

016410  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 100
016416  012737  000100  001124          MOV     #100,$GDDAT     ;GET GOOD = 0

016424  017737  165452  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
016432  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
016440  001401                          BEQ     81$             ;BRANCH IF GOOD
016442  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
016444                                  81$:
```

```
                                                MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                                                ;CHECK THAT RHDB HAS 200
016452  012737  000200  001124                MOV     #200,$GDDAT     ;GET GOOD = 0

016460  017737  165416  001126                MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
016466  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHDB
016474  001401                                BEQ     83$             ;BRANCH IF GOOD
016476  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
016500                                  83$:

                                                ;CHECK THAT RHCS2 HAS IR
016500  012737  000100  001124                MOV     #IR,$GDDAT      ;GET GOOD = 100
016506  053737  005010  001124                BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

016514  017737  165350  001126                MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
016522  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS2
016530  001401                                BEQ     85$             ;BRANCH IF GOOD
016532  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
016534                                  85$:
                                                ;CHECK THAT RHCS1 HAS DVA!RDY
016534  012737  004200  001124                MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

016542  017737  165312  001126                MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
016550  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS1
```

```
5505  016556  001401                          BEQ     87$             ;BRANCH IF GOOD
5506  016560  104026                          ERROR   26
5507                                                                  ;AFTER SETTING "CLR" BIT #5
5508                                                                  ;IN RHCS2 TO INIT THE RH
5509                                                                  ;AND WRITING A PATTERN OF
5510                                                                  ;FLOATING ONES (WHICH IS
5511                                                                  ;1,2,4,10,20,40,100,200)
5512                                                                  ;INTO RHDB TO FILL IT
5513                                                                  ;THEN ON READING RHDB THE
5514                                                                  ;TOTAL 8 TIMES
5515                                                                  ;RHCS1 SHOULD HAVE DVA!RDY
5516                                                                  ;=4200
5517                                                                  ;BUT CONTAINED WHAT IS
5518                                                                  ;GIVEN IN BAD RHCS1
5519  016562                            87$:
5520                                          ;CHECK THAT RHCS3 HAS 0
5521  016562  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5522
5523  016570  017737  165336  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
5524  016576  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5525                                                                  ;DATA WITH DATA READ FROM
5526                                                                  ;RHCS3
5527  016604  001401                          BEQ     89$             ;BRANCH IF GOOD
5528  016606  104027                          ERROR   27
5529                                                                  ;AFTER SETTING "CLR" BIT #5
5530                                                                  ;IN RHCS2 TO INIT THE RH
5531                                                                  ;AND WRITING A PATTERN OF
5532                                                                  ;FLOATING ONES (WHICH IS
5533                                                                  ;1,2,4,10,20,40,100,200)
5534                                                                  ;INTO RHDB TO FILL IT
5535                                                                  ;THEN ON READING RHDB THE
5536                                                                  ;TOTAL 8 TIMES
5537                                                                  ;RHCS3 SHOULD HAVE 0
5538                                                                  ;BUT CONTAINED WHAT IS
5539                                                                  ;GIVEN IN BAD RHCS3
5540  016610                            89$:
5541                                          ;CHECK THAT RHBA HAS 0
5542  016610  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5543
5544  016616  017737  165242  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
5545  016624  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5546                                                                  ;DATA WITH DATA READ FROM
5547                                                                  ;RHBA
5548  016632  001401                          BEQ     91$             ;BRANCH IF GOOD
5549  016634  104030                          ERROR   30
5550                                                                  ;AFTER SETTING "CLR" BIT #5
5551                                                                  ;IN RHCS2 TO INIT THE RH
5552                                                                  ;AND WRITING A PATTERN OF
5553                                                                  ;FLOATING ONES (WHICH IS
5554                                                                  ;1,2,4,10,20,40,100,200)
5555                                                                  ;INTO RHDB TO FILL IT
5556                                                                  ;THEN ON READING RHDB THE
5557                                                                  ;TOTAL 8 TIMES
5558                                                                  ;RHBA SHOULD HAVE 0
5559                                                                  ;BUT CONTAINED WHAT IS
5560                                                                  ;GIVEN IN BAD RHBA
```

```
5561  016636                                  91$:
5562                                          ;CHECK THAT RHBAE HAS 0
5563  016636  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5564
5565  016644  017737  165260  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
5566  016652  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5567                                                                  ;DATA WITH DATA READ FROM
5568                                                                  ;RHBAE
5569  016660  001401                          BEQ     93$             ;BRANCH IF GOOD
5570  016662  104031                          ERROR   31
5571                                                                  ;AFTER SETTING "CLR" BIT #5
5572                                                                  ;IN RHCS2 TO INIT THE RH
5573                                                                  ;AND WRITING A PATTERN OF
5574                                                                  ;FLOATING ONES (WHICH IS
5575                                                                  ;1,2,4,10,20,40,100,200)
5576                                                                  ;INTO RHDB TO FILL IT
5577                                                                  ;THEN ON READING RHDB THE
5578                                                                  ;TOTAL 8 TIMES
5579                                                                  ;RHBAE SHOULD HAVE 0
5580                                                                  ;BUT CONTAINED WHAT IS
5581                                                                  ;GIVEN IN BAD RHBAE
5582  016664                                  93$:
5583                                          ;CHECK THAT RHWC HAS 0
5584  016664  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
5585
5586  016672  017737  165164  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
5587  016700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5588                                                                  ;DATA WITH DATA READ FROM
5589                                                                  ;RHWC
5590  016706  001401                          BEQ     95$             ;BRANCH IF GOOD
5591  016710  104032                          ERROR   32
5592                                                                  ;AFTER SETTING "CLR" BIT #5
5593                                                                  ;IN RHCS2 TO INIT THE RH
5594                                                                  ;AND WRITING A PATTERN OF
5595                                                                  ;FLOATING ONES (WHICH IS
5596                                                                  ;1,2,4,10,20,40,100,200)
5597                                                                  ;INTO RHDB TO FILL IT
5598                                                                  ;THEN ON READING RHDB THE
5599                                                                  ;TOTAL 8 TIMES
5600                                                                  ;RHWC SHOULD HAVE 0
5601                                                                  ;BUT CONTAINED WHAT IS
5602                                                                  ;GIVEN IN BAD RHWC
5603  016712                                  95$:
5604
5605
5606
5607                                          ;;*****************************************************************
5608                                          ;*TEST 16       SILO TEST 6 (FLOATING ONES IN UPPER BYTE)
5609
5610                                          ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
5611                                          ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
5612                                          ;*      TOGETHER WITH UNIT NUMBER
5613                                          ;*      EIGHT WORDS, A PATTERN OF FLOATING ONES IN UPPER BYTE
5614                                          ;*      400,1000,2000,4000,10000,20000,40000,100000
5615                                          ;*      IS WRITTEN INTO RHDB
5616                                          ;*      "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
                                        ;**************************************************************
016712  000004                        TST16:  SCOPE
016714  012706  001000                        MOV     #STACK,SP       ;RESET STACK
016720  012737  000016  004656                MOV     #16.,@#TSTNM    ;SAVE TEST NUMBER

016726  004737  040306                        JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS2 HAS IR
016732  012737  000100  001124                MOV     #IR,$GDDAT      ;GET GOOD = 100
016740  053737  005010  001124                BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

016746  017737  165116  001126                MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
016754  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS2
016762  001401                                BEQ     65$             ;BRANCH IF GOOD
016764  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
016766                                  65$:

                                        ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
                                        ;DATA IS 400,1000,2000,4000,10000,20000,40000
                                        ;100000
016766  012701  000400                        MOV     #400,R1         ;GETTING READY TO FLOAT ONES
                                                                        ;IN UPPER BYTE
016772  012702  000010                        MOV     #8.,R2          ;COUNTER -8 DATA WORDS
016776  010177  165100                1$:     MOV     R1,@RHDB        ;WRITE INTO RHDB
017002  006301                                ASL     R1              ;SHIFT 1 ONE POSITION LEFT
017004  005302                                DEC     R2              ;COUNT TO 8
017006  001373                                BNE     1$              ;BRANCH IF 8 NOT DONE
                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
017010  104411                                WAT                     ;TRAP TO WAIT.T SUBROUTINE
017012  004070                                RHCS2                   ;AND WAIT FOR OR BIT IN
017014  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
017016  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
017024  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

017032  017737  165032  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
017040  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
017046  001401                          BEQ     67$             ;BRANCH IF GOOD
017050  104023                          ERROR   23
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;AND HAVING WRITTEN A
                                                        ;FLOATING ONES PATTERN INTO UPPER BYTE
                                                        ;400,1000,2000,4000,10000,20000
                                                        ;40000,100000
                                                        ;INTO RHDB TO FILL IT
                                                        ;RHCS2 SHOULD HAVE OR
                                                        ;=200
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
017052                          67$:


017052  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 400
017060  012737  000400  001124          MOV     #400,$GDDAT     ;GET GOOD = 0

017066  017737  165010  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
017074  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
017102  001401                          BEQ     69$             ;BRANCH IF GOOD
017104  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
                                                        ;FIRST TIME
                                                        ;RHDB SHOULD HAVE 400
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
017106                          69$:
```

```
5729  017106  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
5730                                          ;CHECK THAT RHDB HAS 1000
5731  017114  012737  001000  001124          MOV     #1000,$GDDAT    ;GET GOOD = 0
5732
5733  017122  017737  164754  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5734  017130  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5735                                                                  ;DATA WITH DATA READ FROM
5736                                                                  ;RHDB
5737  017136  001401                          BEQ     71$             ;BRANCH IF GOOD
5738  017140  104024                          ERROR   24
5739                                                                  ;AFTER SETTING "CLR" BIT #5
5740                                                                  ;IN RHCS2 TO INIT THE RH
5741                                                                  ;AND WRITING A PATTERN OF FLOATING
5742                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5743                                                                  ;10000,20000,40000,100000)
5744                                                                  ;INTO RHDB TO FILL IT
5745                                                                  ;THEN ON READING RHDB THE
5746                                                                  ;SECOND TIME
5747                                                                  ;RHDB SHOULD HAVE 1000
5748                                                                  ;BUT CONTAINED WHAT IS
5749                                                                  ;GIVEN IN BAD RHDB
5750  017142                          71$:
5751
5752
5753  017142  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
5754                                          ;CHECK THAT RHDB HAS 2000
5755  017150  012737  002000  001124          MOV     #2000,$GDDAT    ;GET GOOD = 0
5756
5757  017156  017737  164720  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5758  017164  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5759                                                                  ;DATA WITH DATA READ FROM
5760                                                                  ;RHDB
5761  017172  001401                          BEQ     73$             ;BRANCH IF GOOD
5762  017174  104024                          ERROR   24
5763                                                                  ;AFTER SETTING "CLR" BIT #5
5764                                                                  ;IN RHCS2 TO INIT THE RH
5765                                                                  ;AND WRITING A PATTERN OF FLOATING
5766                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5767                                                                  ;10000,20000,40000,100000)
5768                                                                  ;INTO RHDB TO FILL IT
5769                                                                  ;THEN ON READING RHDB THE
5770                                                                  ;THIRD TIME
5771                                                                  ;RHDB SHOULD HAVE 2000
5772                                                                  ;BUT CONTAINED WHAT IS
5773                                                                  ;GIVEN IN BAD RHDB
5774  017176                          73$:
5775
5776
5777  017176  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
5778                                          ;CHECK THAT RHDB HAS 4000
5779  017204  012737  004000  001124          MOV     #4000,$GDDAT    ;GET GOOD = 0
5780
5781  017212  017737  164664  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5782  017220  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5783                                                                  ;DATA WITH DATA READ FROM
5784                                                                  ;RHDB
```

```
5785  017226  001401                          BEQ     75$             ;BRANCH IF GOOD
5786  017230  104024                          ERROR   24
5787                                                                  ;AFTER SETTING "CLR" BIT #5
5788                                                                  ;IN RHCS2 TO INIT THE RH
5789                                                                  ;AND WRITING A PATTERN OF FLOATING
5790                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5791                                                                  ;10000,20000,40000,100000)
5792                                                                  ;INTO RHDB TO FILL IT
5793                                                                  ;THEN ON READING RHDB THE
5794                                                                  ;FOURTH TIME
5795                                                                  ;RHDB SHOULD HAVE 4000
5796                                                                  ;BUT CONTAINED WHAT IS
5797                                                                  ;GIVEN IN BAD RHDB
5798  017232                          75$:
5799
5800
5801  017232  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
5802                                          ;CHECK THAT RHDB HAS 10000
5803  017240  012737  010000  001124          MOV     #10000,$GDDAT   ;GET GOOD = 0
5804
5805  017246  017737  164630  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5806  017254  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5807                                                                  ;DATA WITH DATA READ FROM
5808                                                                  ;RHDB
5809  017262  001401                          BEQ     77$             ;BRANCH IF GOOD
5810  017264  104024                          ERROR   24
5811                                                                  ;AFTER SETTING "CLR" BIT #5
5812                                                                  ;IN RHCS2 TO INIT THE RH
5813                                                                  ;AND WRITING A PATTERN OF FLOATING
5814                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5815                                                                  ;10000,20000,40000,100000)
5816                                                                  ;INTO RHDB TO FILL IT
5817                                                                  ;THEN ON READING RHDB THE
5818                                                                  ;FIFTH TIME
5819                                                                  ;RHDB SHOULD HAVE 10000
5820                                                                  ;BUT CONTAINED WHAT IS
5821                                                                  ;GIVEN IN BAD RHDB
5822  017266                          77$:
5823
5824
5825  017266  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
5826                                          ;CHECK THAT RHDB HAS 20000
5827  017274  012737  020000  001124          MOV     #20000,$GDDAT   ;GET GOOD = 0
5828
5829  017302  017737  164574  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
5830  017310  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5831                                                                  ;DATA WITH DATA READ FROM
5832                                                                  ;RHDB
5833  017316  001401                          BEQ     79$             ;BRANCH IF GOOD
5834  017320  104024                          ERROR   24
5835                                                                  ;AFTER SETTING "CLR" BIT #5
5836                                                                  ;IN RHCS2 TO INIT THE RH
5837                                                                  ;AND WRITING A PATTERN OF FLOATING
5838                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
5839                                                                  ;10000,20000,40000,100000)
5840                                                                  ;INTO RHDB TO FILL IT
```

```
                                                                ;THEN ON READING RHDB THE
                                                                ;SIXTH TIME
                                                                ;RHDB SHOULD HAVE 20000
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
017322                                  79$:

017322  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 40000
017330  012737  040000  001124          MOV     #40000,$GDDAT   ;GET GOOD = 0

017336  017737  164540  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
017344  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
017352  001401                          BEQ     81$             ;BRANCH IF GOOD
017354  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
017356                                  81$:

017356  012737  000010  004664          MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 100000
017364  012737  100000  001124          MOV     #100000,$GDDAT  ;GET GOOD = 0

017372  017737  164504  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
017400  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
017406  001401                          BEQ     83$             ;BRANCH IF GOOD
017410  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
017412                                  83$:
```

```
5897                                            ;CHECK THAT RHCS2 HAS IR
5898  017412  012737  000100  001124            MOV     #IR,$GDDAT      ;GET GOOD = 100
5899  017420  053737  005010  001124            BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
5900
5901  017426  017737  164436  001126            MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
5902  017434  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5903                                                                    ;DATA WITH DATA READ FROM
5904                                                                    ;RHCS2
5905  017442  001401                            BEQ     85$             ;BRANCH IF GOOD
5906  017444  104025                            ERROR   25
5907                                                                    ;AFTER SETTING "CLR" BIT #5
5908                                                                    ;IN RHCS2 TO INIT THE RH
5909                                                                    ;AND WRITING A PATTERN OF FLOATING
5910                                                                    ;ONES (WHICH IS 400,1000,2000,4000,
5911                                                                    ;10000,20000,40000,100000)
5912                                                                    ;INTO RHDB TO FILL IT
5913                                                                    ;THEN ON READING RHDB THE
5914                                                                    ;TOTAL 8 TIMES
5915                                                                    ;RHCS2 SHOULD HAVE IR
5916                                                                    ;=100
5917                                                                    ;TOGETHER WITH UNIT NUMBER
5918                                                                    ;BUT CONTAINED WHAT IS
5919                                                                    ;GIVEN IN BAD RHCS2
5920  017446                            85$:
5921                                            ;CHECK THAT RHCS1 HAS DVA!RDY
5922  017446  012737  004200  001124            MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
5923
5924  017454  017737  164400  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
5925  017462  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5926                                                                    ;DATA WITH DATA READ FROM
5927                                                                    ;RHCS1
5928  017470  001401                            BEQ     87$             ;BRANCH IF GOOD
5929  017472  104026                            ERROR   26
5930                                                                    ;AFTER SETTING "CLR" BIT #5
5931                                                                    ;IN RHCS2 TO INIT THE RH
5932                                                                    ;AND WRITING A PATTERN OF FLOATING
5933                                                                    ;ONES (WHICH IS 400,1000,2000,4000,
5934                                                                    ;10000,20000,40000,100000)
5935                                                                    ;INTO RHDB TO FILL IT
5936                                                                    ;THEN ON READING RHDB THE
5937                                                                    ;TOTAL 8 TIMES
5938                                                                    ;RHCS1 SHOULD HAVE DVA!RDY
5939                                                                    ;=4200
5940                                                                    ;BUT CONTAINED WHAT IS
5941                                                                    ;GIVEN IN BAD RHCS1
5942  017474                            87$:
5943                                            ;CHECK THAT RHCS3 HAS 0
5944  017474  012737  000000  001124            MOV     #0,$GDDAT       ;GET GOOD = 0
5945
5946  017502  017737  164424  001126            MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
5947  017510  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
5948                                                                    ;DATA WITH DATA READ FROM
5949                                                                    ;RHCS3
5950  017516  001401                            BEQ     89$             ;BRANCH IF GOOD
5951  017520  104027                            ERROR   27
5952                                                                    ;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
017522                          89$:
                                        ;CHECK THAT RHBA HAS 0
017522  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

017530  017737  164330  001126        MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
017536  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBA
017544  001401                        BEQ     91$             ;BRANCH IF GOOD
017546  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
017550                          91$:
                                        ;CHECK THAT RHBAE HAS 0
017550  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

017556  017737  164346  001126        MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
017564  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBAE
017572  001401                        BEQ     93$             ;BRANCH IF GOOD
017574  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
017576                          93$:
                                        ;CHECK THAT RHWC HAS 0
017576  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
```

```
 6009  017604  017737  164252  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
 6010  017612  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 6011                                                                  ;DATA WITH DATA READ FROM
 6012                                                                  ;RHWC
 6013  017620  001401                          BEQ     95$             ;BRANCH IF GOOD
 6014  017622  104032                          ERROR   32
 6015                                                                  ;AFTER SETTING "CLR" BIT #5
 6016                                                                  ;IN RHCS2 TO INIT THE RH
 6017                                                                  ;AND WRITING A PATTERN OF FLOATING
 6018                                                                  ;ONES (WHICH IS 400,1000,2000,4000,
 6019                                                                  ;10000,20000,40000,100000)
 6020                                                                  ;INTO RHDB TO FILL IT
 6021                                                                  ;THEN ON READING RHDB THE
 6022                                                                  ;TOTAL 8 TIMES
 6023                                                                  ;RHWC SHOULD HAVE 0
 6024                                                                  ;BUT CONTAINED WHAT IS
 6025                                                                  ;GIVEN IN BAD RHWC
 6026  017624                          95$:
 6027
 6028
 6029                          ;;*****************************************************************
 6030                          ;*TEST 17       SILO TEST 7 (FLOATING 0 IN LOWER BYTE)
 6031
 6032                          ;*      AFTER A RH CLEAR IS GIVEN (SET BIT #5 IN RHCS2)
 6033                          ;*      RHCS2 IS CHECKED TO HAVE "IR" (BIT #6) HIGH,
 6034                          ;*      TOGETHER WITH UNIT NUMBER
 6035                          ;*      EIGHT WORDS OF FLOATING ZEROS 177776, 177775, 177773,
 6036                          ;*      177767, 177757, 177737, 177677, 177577
 6037                          ;*      IS WRITTEN INTO RHDB
 6038                          ;*      "OR" (BIT #7 IN RHCS2) IS WAITED FOR BY A TRAP
 6039                          ;*      INSTRUCTION CALLED "WAT"
 6040                          ;*      THEN RHCS2 IS CHECKED TO HAVE "IR" LOW AND
 6041                          ;*      "OR" HIGH TOGETHER WITH THE UNIT NUMBER
 6042                          ;*      THEN RHDB IS READ AND COMPARED TO HAVE THE
 6043                          ;*      RIGHT VALUE AFTER EACH OF THE EIGHT READS.
 6044                          ;*      THEN RHCS2 IS CHECKED TO HAVE "IR"
 6045                          ;*      AND UNIT NUMBER
 6046                          ;*      THEN RHCS1, RHCS3, RHBA, RHBAE, RHWC ARE
 6047                          ;*      CHECKED TO HAVE THE APPROPIATE VALUE
 6048                          ;;*****************************************************************
 6049  017624  000004          TST17:  SCOPE
 6050  017626  012706  001000          MOV     #STACK,SP       ;RESET STACK
 6051  017632  012737  000017  004656  MOV     #17,@#TSTNM     ;SAVE TEST NUMBER
 6052
 6053  017640  004737  040306          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
 6054                                                          ;SETUP UNIT NUBER
 6055                                                          ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
 6056
 6057
 6058                                  ;CHECK THAT RHCS2 HAS IR
 6059  017644  012737  000100  001124  MOV     #IR,$GDDAT      ;GET GOOD = 100
 6060  017652  053737  005010  001124  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 6061
 6062  017660  017737  164204  001126  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 6063  017666  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 6064                                                          ;DATA WITH DATA READ FROM
```

```
6065
6066  017674  001401                          BEQ     65$             ;RHCS2
6067  017676  104006                          ERROR   6               ;BRANCH IF GOOD
6068                                                                  ;AFTER SETTING "CLR" BIT #5
6069                                                                  ;IN RHCS2 TO INIT THE RH
6070                                                                  ;AND HAVING DONE NOTHING
6071                                                                  ;ELSE
6072                                                                  ;RHCS2 SHOULD HAVE IR
6073                                                                  ;=100
6074                                                                  ;TOGETHER WITH UNIT NUMBER
6075                                                                  ;BUT CONTAINED WHAT IS
6076                                                                  ;GIVEN IN BAD RHCS2
6077  017700                            65$:
6078
6079                                          ;WRITE EIGHT WORDS INTO RHDB TO FILL IT
6080                                          ;DATA IS FLOATING ZEROS 177776, 177775, 177773, 177767, 177757, 177737,
6081                                          ;177677, 177577
6082  017700  012701  177776                 MOV     #177776,R1      ;GETTING READY TO FLOAT ZEROS
6083  017704  012702  000010                 MOV     #8.,R2          ;COUNTER -8 DATA WORDS
6084  017710  010177  164166           1$:   MOV     R1,@RHDB        ;WRITE INTO RHDB
6085  017714  000261                       SEC                     ;SET CARRY
6086  017716  006101                       ROL     R1              ;GET ZERO ONE BIT LEFT
6087  017720  005302                       DEC     R2              ;COUNT TO 8
6088  017722  001372                       BNE     1$              ;BRANCH IF 8 NOT DONE
6089                                          ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
6090  017724  104411                       WAT                     ;TRAP TO WAIT.T SUBROUTINE
6091  017726  004070                       RHCS2                   ;AND WAIT FOR OR BIT IN
6092  017730  000200                       OR                      ;RHCS2 REGISTER
6093                                                                ;IF ERROR OCCURS HERE
6094                                                                ;IT MEANS "OR" DID NOT
6095                                                                ;SET FOR THE FULL COUNT
6096                                                                ;DOWN OF THE WAIT.T SUBROUTINE
6097                                                                ;THIS TIME IS APPROXIMATELY
6098                                                                ;LARGER THAN 500 MILLISECONDS
6099
6100
6101                                          ;CHECK THAT RHCS2 HAS OR
6102  017732  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
6103  017740  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
6104
6105  017746  017737  164116  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
6106  017754  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6107                                                                ;DATA WITH DATA READ FROM
6108                                                                ;RHCS2
6109  017762  001401                          BEQ     67$             ;BRANCH IF GOOD
6110  017764  104023                          ERROR   23
6111                                                                ;AFTER SETTING "CLR" BIT #5
6112                                                                ;IN RHCS2 TO INIT THE RH
6113                                                                ;AND HAVING WRITTEN A PATTERN
6114                                                                ;OF FLOATING ZEROS - 177776, 177775,
6115                                                                ;177773, 177767, 177757, 177737, 177677, 177577
6116                                                                ;INTO RHDB TO FILL IT
6117                                                                ;RHCS2 SHOULD HAVE OR
6118                                                                ;=200
6119                                                                ;TOGETHER WITH UNIT NUMBER
6120                                                                ;BUT CONTAINED WHAT IS
```

```
                                                        ;GIVEN IN BAD RHCS2
017766                                  67$:
017766  012737  000001  004664            MOV     #1,SILONM       ;FIRST WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177776
017774  012737  177776  001124            MOV     #177776,$GDDAT  ;GET GOOD = 0
020002  017737  164074  001126            MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020010  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020016  001401                            BEQ     69$             ;BRANCH IF GOOD
020020  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
020022                                  69$:
020022  012737  000002  004664            MOV     #2,SILONM       ;SECOND WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177775
020030  012737  177775  001124            MOV     #177775,$GDDAT  ;GET GOOD = 0
020036  017737  164040  001126            MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020044  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020052  001401                            BEQ     71$             ;BRANCH IF GOOD
020054  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
020056                                  71$:
020056  012737  000003  004664            MOV     #3,SILONM       ;THIRD WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177773
020064  012737  177773  001124            MOV     #177773,$GDDAT  ;GET GOOD = 0
```

```
                                                                ;
020072  017737  164004  001126        MOV     @RHDB,$BDDAT      ;READ RHDB FOR COMPARISON
020100  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
020106  001401                      BEQ     73$                 ;BRANCH IF GOOD
020110  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
                                                                ;THIRD TIME
                                                                ;RHDB SHOULD HAVE 177773
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
020112                          73$:
                                                                ;
020112  012737  000004  004664        MOV     #4,SILONM         ;FOURTH WORD TO BE READ
                                    ;CHECK THAT RHDB HAS 177767
020120  012737  177767  001124        MOV     #177767,$GDDAT    ;GET GOOD = 0
                                                                ;
020126  017737  163750  001126        MOV     @RHDB,$BDDAT      ;READ RHDB FOR COMPARISON
020134  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
020142  001401                      BEQ     75$                 ;BRANCH IF GOOD
020144  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
                                                                ;FOURTH TIME
                                                                ;RHDB SHOULD HAVE 177767
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHDB
020146                          75$:
                                                                ;
020146  012737  000005  004664        MOV     #5,SILONM         ;FIFTH WORD TO BE READ
                                    ;CHECK THAT RHDB HAS 177757
020154  012737  177757  001124        MOV     #177757,$GDDAT    ;GET GOOD = 0
                                                                ;
020162  017737  163714  001126        MOV     @RHDB,$BDDAT      ;READ RHDB FOR COMPARISON
020170  023737  001124  001126        CMP     $GDDAT,$BDDAT     ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHDB
020176  001401                      BEQ     77$                 ;BRANCH IF GOOD
020200  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
                                                        ;FIFTH TIME
                                                        ;RHDB SHOULD HAVE 177757
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
020202                          77$:

020202  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177737
020210  012737  177737  001124          MOV     #177737,$GDDAT  ;GET GOOD = 0

020216  017737  163660  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020224  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020232  001401                          BEQ     79$             ;BRANCH IF GOOD
020234  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
                                                        ;SIXTH TIME
                                                        ;RHDB SHOULD HAVE 177737
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
020236                          79$:

020236  012737  000007  004664          MOV     #7,SILONM       ;SEVENTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177677
020244  012737  177677  001124          MOV     #177677,$GDDAT  ;GET GOOD = 0

020252  017737  163624  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020260  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020266  001401                          BEQ     81$             ;BRANCH IF GOOD
020270  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
                                                        ;SEVENTH TIME
                                                        ;RHDB SHOULD HAVE 177677
```

```
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
020272                          81$:
                                
                                
020272  012737  000010  004664          MOV     #8.,SILONM      ;EIGHTH WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 177577
020300  012737  177577  001124          MOV     #177577,$GDDAT  ;GET GOOD = 0
                                
020306  017737  163570  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020314  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020322  001401                          BEQ     83$             ;BRANCH IF GOOD
020324  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
020326                          83$:
                                
                                
                                        ;CHECK THAT RHCS2 HAS IR
020326  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
020334  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
                                
020342  017737  163522  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
020350  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
020356  001401                          BEQ     85$             ;BRANCH IF GOOD
020360  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
020362                          85$:
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
020362  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
```

SILO TEST 7 (FLOATING 0 IN LOWER BYTE) SEQ 0123

```
6345  020370  017737  163464  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
6346  020376  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6347                                                                  ;DATA WITH DATA READ FROM
6348                                                                  ;RHCS1
6349  020404  001401                          BEQ     87$             ;BRANCH IF GOOD
6350  020406  104026                          ERROR   26
6351                                                                  ;AFTER SETTING "CLR" BIT #5
6352                                                                  ;IN RHCS2 TO INIT THE RH
6353                                                                  ;AND WRITING A PATTERN OF FLOATING
6354                                                                  ;ZEROS - 177776, 177775, 177773, 177767,
6355                                                                  ;177757, 177737, 177677, 177577
6356                                                                  ;INTO RHDB TO FILL IT
6357                                                                  ;THEN ON READING RHDB THE
6358                                                                  ;TOTAL 8 TIMES
6359                                                                  ;RHCS1 SHOULD HAVE DVA!RDY
6360                                                                  ;=4200
6361                                                                  ;BUT CONTAINED WHAT IS
6362                                                                  ;GIVEN IN BAD RHCS1
6363  020410                          87$:
6364                                          ;CHECK THAT RHCS3 HAS 0
6365  020410  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
6366
6367  020416  017737  163510  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
6368  020424  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6369                                                                  ;DATA WITH DATA READ FROM
6370                                                                  ;RHCS3
6371  020432  001401                          BEQ     89$             ;BRANCH IF GOOD
6372  020434  104027                          ERROR   27
6373                                                                  ;AFTER SETTING "CLR" BIT #5
6374                                                                  ;IN RHCS2 TO INIT THE RH
6375                                                                  ;AND WRITING A PATTERN OF FLOATING
6376                                                                  ;ZEROS - 177776, 177775, 177773, 177767,
6377                                                                  ;177757, 177737, 177677, 177577
6378                                                                  ;INTO RHDB TO FILL IT
6379                                                                  ;THEN ON READING RHDB THE
6380                                                                  ;TOTAL 8 TIMES
6381                                                                  ;RHCS3 SHOULD HAVE 0
6382                                                                  ;BUT CONTAINED WHAT IS
6383                                                                  ;GIVEN IN BAD RHCS3
6384  020436                          89$:
6385                                          ;CHECK THAT RHBA HAS 0
6386  020436  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
6387
6388  020444  017737  163414  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
6389  020452  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6390                                                                  ;DATA WITH DATA READ FROM
6391                                                                  ;RHBA
6392  020460  001401                          BEQ     91$             ;BRANCH IF GOOD
6393  020462  104030                          ERROR   30
6394                                                                  ;AFTER SETTING "CLR" BIT #5
6395                                                                  ;IN RHCS2 TO INIT THE RH
6396                                                                  ;AND WRITING A PATTERN OF FLOATING
6397                                                                  ;ZEROS - 177776, 177775, 177773, 177767,
6398                                                                  ;177757, 177737, 177677, 177577
6399                                                                  ;INTO RHDB TO FILL IT
6400                                                                  ;THEN ON READING RHDB THE
```

```
                                                                ;TOTAL 8 TIMES
                                                                ;RHBA SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBA
020464                                  91$:
                                        ;CHECK THAT RHBAE HAS 0
020464  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

020472  017737  163432  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
020500  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
020506  001401                          BEQ     93$             ;BRANCH IF GOOD
020510  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
020512                                  93$:
                                        ;CHECK THAT RHWC HAS 0
020512  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

020520  017737  163336  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
020526  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
020534  001401                          BEQ     95$             ;BRANCH IF GOOD
020536  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
020540                          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
                                        ;*****************************************************************
020540  000004                    TST20:  SCOPE
020542  012706  001000                    MOV     #STACK,SP       ;RESET STACK
020546  012737  000020  004656            MOV     #20,@#TSTNM     ;SAVE TEST NUMBER

020554  004737  040306                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                  ;SETUP UNIT NUBER
                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS2 HAS IR
020560  012737  000100  001124            MOV     #IR,$GDDAT      ;GET GOOD = 100
020566  053737  005010  001124            BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

020574  017737  163270  001126            MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
020602  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                  ;DATA WITH DATA READ FROM
                                                                  ;RHCS2
020610  001401                            BEQ     65$             ;BRANCH IF GOOD
020612  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
020614                              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
020614  012701  177377                    MOV     #177377,R1      ;GETTING READY TO FLOAT ZEROS
                                                                  ;IN UPPER BYTE
020620  012702  000010                    MOV     #8.,R2          ;COUNTER -8 DATA WORDS
020624  010177  163252            1$:     MOV     R1,@RHDB        ;WRITE INTO RHDB
020630  000261                            SEC                     ;SET CARRY
020632  006101                            ROL     R1              ;GET ZERO ONE BIT LEFT
020634  005302                            DEC     R2              ;COUNT TO 8
020636  001372                            BNE     1$              ;BRANCH IF 8 NOT DONE
                                        ;WAIT FOR OR BIT IN RHCS2 REGISTER TO SET
```

```
6513  020640  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
6514  020642  004070                          RHCS2                   ;AND WAIT FOR OR BIT IN
6515  020644  000200                          OR                      ;RHCS2 REGISTER
6516                                                                  ;IF ERROR OCCURS HERE
6517                                                                  ;IT MEANS "OR" DID NOT
6518                                                                  ;SET FOR THE FULL COUNT
6519                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
6520                                                                  ;THIS TIME IS APPROXIMATELY
6521                                                                  ;LARGER THAN 500 MILLISECONDS
6522
6523
6524                                          ;CHECK THAT RHCS2 HAS OR
6525  020646  012737  000200  001124          MOV     #OR,$GDDAT      ;GET GOOD = 200
6526  020654  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
6527
6528  020662  017737  163202  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
6529  020670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6530                                                                  ;DATA WITH DATA READ FROM
6531                                                                  ;RHCS2
6532  020676  001401                          BEQ     67$             ;BRANCH IF GOOD
6533  020700  104023                          ERROR   23
6534                                                                  ;AFTER SETTING "CLR" BIT #5
6535                                                                  ;IN RHCS2 TO INIT THE RH
6536                                                                  ;AND HAVING WRITTEN A
6537                                                                  ;PATTERN OF FLOATING ZEROS IN UPPER BYTE
6538                                                                  ;177377, 176777, 175777, 173777,
6539                                                                  ;167777, 157777, 137777, 77777
6540                                                                  ;INTO RHDB TO FILL IT
6541                                                                  ;RHCS2 SHOULD HAVE OR
6542                                                                  ;=200
6543                                                                  ;TOGETHER WITH UNIT NUMBER
6544                                                                  ;BUT CONTAINED WHAT IS
6545                                                                  ;GIVEN IN BAD RHCS2
6546  020702                          67$:
6547
6548
6549
6550  020702  012737  000001  004664          MOV     #1,SILONM       ;FIRST WORD TO BE READ
6551                                          ;CHECK THAT RHDB HAS 177377
6552  020710  012737  177377  001124          MOV     #177377,$GDDAT  ;GET GOOD = 0
6553
6554  020716  017737  163160  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
6555  020724  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6556                                                                  ;DATA WITH DATA READ FROM
6557                                                                  ;RHDB
6558  020732  001401                          BEQ     69$             ;BRANCH IF GOOD
6559  020734  104024                          ERROR   24
6560                                                                  ;AFTER SETTING "CLR" BIT #5
6561                                                                  ;IN RHCS2 TO INIT THE RH
6562                                                                  ;AND WRITING A PATTERN OF FLOATING
6563                                                                  ;ZEROS IN UPPER BYTE
6564                                                                  ;177377, 176777, 175777, 173777,
6565                                                                  ;167777, 157777, 137777, 77777
6566                                                                  ;INTO RHDB TO FILL IT
6567                                                                  ;THEN ON READING RHDB THE
6568                                                                  ;FIRST TIME
```

```
                                                        ;RHDB SHOULD HAVE 177377
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
020736                              69$:

020736  012737  000002  004664          MOV     #2,SILONM       ;SECOND WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 176777
020744  012737  176777  001124          MOV     #176777,$GDDAT  ;GET GOOD = 0

020752  017737  163124  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
020760  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
020766  001401                          BEQ     71$             ;BRANCH IF GOOD
020770  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
020772                              71$:

020772  012737  000003  004664          MOV     #3,SILONM       ;THIRD WORD TO BE READ
                                        ;CHECK THAT RHDB HAS 175777
021000  012737  175777  001124          MOV     #175777,$GDDAT  ;GET GOOD = 0

021006  017737  163070  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
021014  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021022  001401                          BEQ     73$             ;BRANCH IF GOOD
021024  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
021026                              73$:
```

```
 6625  021026  012737  000004  004664          MOV     #4,SILONM       ;FOURTH WORD TO BE READ
 6626                                          ;CHECK THAT RHDB HAS 173777
 6627  021034  012737  173777  001124          MOV     #173777,$GDDAT  ;GET GOOD = 0
 6628
 6629  021042  017737  163034  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
 6630  021050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 6631                                                                  ;DATA WITH DATA READ FROM
 6632                                                                  ;RHDB
 6633  021056  001401                          BEQ     75$             ;BRANCH IF GOOD
 6634  021060  104024                          ERROR   24
 6635                                                                  ;AFTER SETTING "CLR" BIT #5
 6636                                                                  ;IN RHCS2 TO INIT THE RH
 6637                                                                  ;AND WRITING A PATTERN OF FLOATING
 6638                                                                  ;ZEROS IN UPPER BYTE
 6639                                                                  ;177377, 176777, 175777, 173777,
 6640                                                                  ;167777, 157777, 137777, .7777
 6641                                                                  ;INTO RHDB TO FILL IT
 6642                                                                  ;THEN ON READING RHDB THE
 6643                                                                  ;FOURTH TIME
 6644                                                                  ;RHDB SHOULD HAVE 173777
 6645                                                                  ;BUT CONTAINED WHAT IS
 6646                                                                  ;GIVEN IN BAD RHDB
 6647  021062                          75$:
 6648
 6649
 6650  021062  012737  000005  004664          MOV     #5,SILONM       ;FIFTH WORD TO BE READ
 6651                                          ;CHECK THAT RHDB HAS 167777
 6652  021070  012737  167777  001124          MOV     #167777,$GDDAT  ;GET GOOD = 0
 6653
 6654  021076  017737  163000  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
 6655  021104  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 6656                                                                  ;DATA WITH DATA READ FROM
 6657                                                                  ;RHDB
 6658  021112  001401                          BEQ     77$             ;BRANCH IF GOOD
 6659  021114  104024                          ERROR   24
 6660                                                                  ;AFTER SETTING "CLR" BIT #5
 6661                                                                  ;IN RHCS2 TO INIT THE RH
 6662                                                                  ;AND WRITING A PATTERN OF FLOATING
 6663                                                                  ;ZEROS IN UPPER BYTE
 6664                                                                  ;177377, 176777, 175777, 173777,
 6665                                                                  ;167777, 157777, 137777, 77777
 6666                                                                  ;INTO RHDB TO FILL IT
 6667                                                                  ;THEN ON READING RHDB THE
 6668                                                                  ;FIFTH TIME
 6669                                                                  ;RHDB SHOULD HAVE 167777
 6670                                                                  ;BUT CONTAINED WHAT IS
 6671                                                                  ;GIVEN IN BAD RHDB
 6672  021116                          77$:
 6673
 6674
 6675  021116  012737  000006  004664          MOV     #6,SILONM       ;SIXTH WORD TO BE READ
 6676                                          ;CHECK THAT RHDB HAS 157777
 6677  021124  012737  157777  001124          MOV     #157777,$GDDAT  ;GET GOOD = 0
 6678
 6679  021132  017737  162744  001126          MOV     @RHDB,$BDDAT    ;READ RHDB FOR COMPARISON
 6680  021140  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021146  001401                      BEQ     79$           ;BRANCH IF GOOD
021150  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
021152                          79$:

021152  012737  000007  004664      MOV     #7,SILONM     ;SEVENTH WORD TO BE READ
                                ;CHECK THAT RHDB HAS 137777
021160  012737  137777  001124      MOV     #137777,$GDDAT ;GET GOOD = 0

021166  017737  162710  001126      MOV     @RHDB,$BDDAT  ;READ RHDB FOR COMPARISON
021174  023737  001124  001126      CMP     $GDDAT,$BDDAT ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021202  001401                      BEQ     81$           ;BRANCH IF GOOD
021204  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
021206                          81$:

021206  012737  000010  004664      MOV     #8.,SILONM    ;EIGHTH WORD TO BE READ
                                ;CHECK THAT RHDB HAS 77777
021214  012737  077777  001124      MOV     #77777,$GDDAT ;GET GOOD = 0

021222  017737  162654  001126      MOV     @RHDB,$BDDAT  ;READ RHDB FOR COMPARISON
021230  023737  001124  001126      CMP     $GDDAT,$BDDAT ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHDB
021236  001401                      BEQ     83$           ;BRANCH IF GOOD
021240  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
                                                        ;EIGHTH TIME
                                                        ;RHDB SHOULD HAVE 77777
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHDB
021242                          83$:

                                        ;CHECK THAT RHCS2 HAS IR
021242  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
021250  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

021256  017737  162606  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
021264  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
021272  001401                          BEQ     85$             ;BRANCH IF GOOD
021274  104025                          ERROR   25
                                                        ;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
                                                        ;RHCS2 SHOULD HAVE IR
                                                        ;=100
                                                        ;TOGETHER WITH UNIT NUMBER
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS2
021276                          85$:
                                        ;CHECK THAT RHCS1 HAS DVA!RDY
021276  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

021304  017737  162550  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
021312  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
021320  001401                          BEQ     87$             ;BRANCH IF GOOD
021322  104026                          ERROR   26

                                                        ;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
```

```
                                                        ;RHCS1 SHOULD HAVE DVA!RDY
                                                        ;=4200
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS1
021324                                  87$:
                                        ;CHECK THAT RHCS3 HAS 0
021324  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

021332  017737  162574  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
021340  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
021346  001401                          BEQ     89$             ;BRANCH IF GOOD
021350  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
021352                                  89$:
                                        ;CHECK THAT RHBA HAS 0
021352  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

021360  017737  162500  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
021366  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBA
021374  001401                          BEQ     91$             ;BRANCH IF GOOD
021376  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
021400                                  91$:
                                        ;CHECK THAT RHBAE HAS 0
021400  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

021406  017737  162516  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
021414  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBAE
```

```
6849  021422  001401                          BEQ     93$             ;BRANCH IF GOOD
6850  021424  104031                          ERROR   31
6851                                                                  ;AFTER SETTING "CLR" BIT #5
6852                                                                  ;IN RHCS2 TO INIT THE RH
6853                                                                  ;AND WRITING A PATTERN OF FLOATING
6854                                                                  ;ZEROS IN UPPER BYTE
6855                                                                  ;177377, 176777, 175777, 173777,
6856                                                                  ;167777, 157777, 137777, 77777
6857                                                                  ;INTO RHDB TO FILL IT
6858                                                                  ;THEN ON READING RHDB THE
6859                                                                  ;TOTAL 8 TIMES
6860                                                                  ;RHBAE SHOULD HAVE 0
6861                                                                  ;BUT CONTAINED WHAT IS
6862                                                                  ;GIVEN IN BAD RHBAE
6863  021426                          93$:
6864                                          ;CHECK THAT RHWC HAS 0
6865  021426  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
6866
6867  021434  017737  162422  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
6868  021442  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
6869                                                                  ;DATA WITH DATA READ FROM
6870                                                                  ;RHWC
6871  021450  001401                          BEQ     95$             ;BRANCH IF GOOD
6872  021452  104032                          ERROR   32
6873                                                                  ;AFTER SETTING "CLR" BIT #5
6874                                                                  ;IN RHCS2 TO INIT THE RH
6875                                                                  ;AND WRITING A PATTERN OF FLOATING
6876                                                                  ;ZEROS IN UPPER BYTE
6877                                                                  ;177377, 176777, 175777, 173777,
6878                                                                  ;167777, 157777, 137777, 77777
6879                                                                  ;INTO RHDB TO FILL IT
6880                                                                  ;THEN ON READING RHDB THE
6881                                                                  ;TOTAL 8 TIMES
6882                                                                  ;RHWC SHOULD HAVE 0
6883                                                                  ;BUT CONTAINED WHAT IS
6884                                                                  ;GIVEN IN BAD RHWC
6885  021454                          95$:
6886
6887
6888                          ;;*****************************************************************
6889                          ;*TEST 21       RHCS1 - MCPE BIT #13 (PARITY LINE = 0)
6890
6891                          ;*       AN RH CLEAR (SET BIT #5 IN RHCS2) IS GIVEN TO
6892                          ;*       CLEAR ALL DEVICE REGISTERS
6893                          ;*       THEN RHER1 (ERROR REGISTER 1) IS CHECKED TO HAVE
6894                          ;*       ZEROS
6895                          ;*       SET "PAT" (BIT #4 IN RHCS2) TO INVERT PARITY CHECKING
6896                          ;*       READ ANY DEVICE REGISTER - HERE RHER1
6897                          ;*       READ AND CHECK RHCS1 TO CONTAIN SC (BIT #15)
6898                          ;*       AND MCPE (BIT #13)
6899                          ;*       WRITE "1" INTO TRE (BIT #14 IN RHCS1)
6900                          ;*       READ RHCS1 SC, MCPE, AND RDY SHOULD BE SET
6901                          ;*       GIVE AN RH CLEAR (CLR - BIT #5 IN RHCS2)
6902                          ;*       CHECK RHCS1 TO HAVE RDY
6903                          ;*       CHECK RHCS2 TO HAVE ONLY IR AND UNIT NUMBER
6904                          ;*       CHECK RHCS3, RHBA, RHBAE, RHWC TO HAVE APPROPIATE
```

```
                                ;*      VALUES.
                                ;;*****************************************************************
021454  000004                  TST21:  SCOPE
021456  012706  001000                  MOV     #STACK,SP       ;RESET STACK
021462  012737  000021  004656          MOV     #21,@#TSTNM     ;SAVE TEST NUMBER

021470  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHER1 HAS 0
021474  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

021502  017737  162366  001126          MOV     @RHER1,$BDDAT   ;READ RHER1 FOR COMPARISON
021510  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHER1
021516  001401                          BEQ     65$             ;BRANCH IF GOOD
021520  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
021522                          65$:

                                        ;TO INVERT THE PARITY CHECKING ON THE CONTROL BUS
                                        ;"PAT" BIT #4 IN RHCS2 IS SET
021522  052777  000020  162340          BIS     #PAT,@RHCS2     ;SET PAT = 20 IN RHCS2


                                        ;CHECK THAT RHER1 HAS 0
021530  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

021536  017737  162332  001126          MOV     @RHER1,$BDDAT   ;READ RHER1 FOR COMPARISON
021544  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHER1
021552  001401                          BEQ     67$             ;BRANCH IF GOOD
021554  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
021556                          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
021556  012737  004200  001124          MOV     #4200,$GDDAT    ;GET GOOD = 124200
```

```
 6961
 6962  021564  017737  162270  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 6963  021572  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 6964                                                                  ;DATA WITH DATA READ FROM
 6965                                                                  ;RHCS1
 6966  021600  001401                          BEQ     69$             ;BRANCH IF GOOD
 6967  021602  104035                          ERROR   35
 6968                                                                  ;AFTER CLEARING THE RH AND
 6969                                                                  ;DEVICE BY AN RH CLEAR
 6970                                                                  ;AND READING RHER1 WITH
 6971                                                                  ;WRONG PARITY CHECKING
 6972                                                                  ;RHCS1 SHOULD HAVE 4200
 6973                                                                  ;=124200
 6974                                                                  ;BUT CONTAINED WHAT IS
 6975                                                                  ;GIVEN IN BAD RHCS1
 6976  021604                          69$:
 6977
 6978                                          ;WRITE A ONE INTO TRE (RHCS1 - BIT #14)
 6979                                          ;THIS SHOULD NOT CLEAR MCPE OR SC
 6980  021604  012777  040000  162246          MOV     #TRE,@RHCS1     ;WRITE "1" INTO TRE IN RHCS1
 6981
 6982
 6983
 6984                                          ;CHECK THAT RHCS1 HAS 104200
 6985  021612  012737  104200  001124          MOV     #104200,$GDDAT  ;GET GOOD = 124200
 6986
 6987  021620  017737  162234  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 6988  021626  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 6989                                                                  ;DATA WITH DATA READ FROM
 6990                                                                  ;RHCS1
 6991  021634  001401                          BEQ     71$             ;BRANCH IF GOOD
 6992  021636  104036                          ERROR   36
 6993                                                                  ;AFTER CLEARING THE RH AND
 6994                                                                  ;DEVICE REGISTERS BY AN
 6995                                                                  ;RH CLEAR
 6996                                                                  ;RHER1 WAS CHECKED TO HAVE
 6997                                                                  ;ZEROS
 6998                                                                  ;WRONG PARITY CHECKING WAS
 6999                                                                  ;SET BY "PAT" BIT IN RHCS2
 7000                                                                  ;RHER1 WAS READ TO SET
 7001                                                                  ;MCPE AND SC IN RHCS1
 7002                                                                  ;ON WRITING "1" INTO TRE
 7003                                                                  ;RHCS1 SHOULD HAVE 104200
 7004                                                                  ;=124200
 7005                                                                  ;BUT CONTAINED WHAT IS
 7006                                                                  ;GIVEN IN BAD RHCS1
 7007  021640                          71$:
 7008
 7009                                          ;NOW ERRORS WILL BE CLEARED BY RH CLEAR
 7010                                          ;SET "CLR" BIT #5 IN RHCS2
 7011  021640  012777  000040  162222          MOV     #CLR,@RHCS2     ;CLEAR BY RH INIT
 7012  021646  013777  005010  162214          MOV     UNIT,@RHCS2     ;REINSTATE UNIT NUMBER
 7013
 7014
 7015                                          ;CHECK THAT RHCS1 HAS DVA!RDY
 7016  021654  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
```

```
7017
7018  021662  017737  162172  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
7019  021670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7020                                                                  ;DATA WITH DATA READ FROM
7021                                                                  ;RHCS1
7022  021676  001401                          BEQ     73$             ;BRANCH IF GOOD
7023  021700  104037                          ERROR   37
7024                                                                  ;AFTER CLEARING THE RH AND
7025                                                                  ;DEVICE REGISTERS BY AN
7026                                                                  ;RH CLEAR
7027                                                                  ;RHER1 WAS CHECKED TO HAVE
7028                                                                  ;ZEROS
7029                                                                  ;WRONG PARITY CHECKING WAS
7030                                                                  ;SET BY "PAT" BIT IN RHCS2
7031                                                                  ;RHER1 WAS READ TO SET
7032                                                                  ;MCPE AND SC IN RHCS1
7033                                                                  ;ON WRITING "1" INTO TRE
7034                                                                  ;THEN SETTING "CLR" IN RHCS1
7035                                                                  ;RHCS1 SHOULD HAVE DVA!RDY
7036                                                                  ;=4200
7037                                                                  ;BUT CONTAINED WHAT IS
7038                                                                  ;GIVEN IN BAD RHCS1
7039  021702                          73$:
7040                                          ;CHECK THAT RHCS2 HAS IR
7041  021702  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
7042  021710  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7043
7044  021716  017737  162146  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7045  021724  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7046                                                                  ;DATA WITH DATA READ FROM
7047                                                                  ;RHCS2
7048  021732  001401                          BEQ     75$             ;BRANCH IF GOOD
7049  021734  104040                          ERROR   40
7050                                                                  ;AFTER CLEARING THE RH AND
7051                                                                  ;DEVICE REGISTERS BY AN
7052                                                                  ;RH CLEAR
7053                                                                  ;RHER1 WAS CHECKED TO HAVE
7054                                                                  ;ZEROS
7055                                                                  ;WRONG PARITY CHECKING WAS
7056                                                                  ;SET BY "PAT" BIT IN RHCS2
7057                                                                  ;RHER1 WAS READ TO SET
7058                                                                  ;MCPE AND SC IN RHCS1
7059                                                                  ;ON WRITING "1" INTO TRE
7060                                                                  ;THEN SETTING "CLR" IN RHCS1
7061                                                                  ;RHCS2 SHOULD HAVE IR
7062                                                                  ;=100
7063                                                                  ;TOGETHER WITH UNIT NUMBER
7064                                                                  ;BUT CONTAINED WHAT IS
7065                                                                  ;GIVEN IN BAD RHCS2
7066  021736                          75$:
7067                                          ;CHECK THAT RHCS3 HAS 0
7068  021736  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
7069
7070  021744  017737  162162  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7071  021752  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7072                                                                  ;DATA WITH DATA READ FROM
```

```
                                                                        ;RHCS3
021760  001401                                  BEQ     77$             ;BRANCH IF GOOD
021762  104041                                  ERROR   41
                                                                        ;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
                                                                        ;RHCS3 SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS3
021764                          77$:
                                        ;CHECK THAT RHBA HAS 0
021764  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

021772  017737  162066  001126          MOV     @RHBA,$BDDAT            ;READ RHBA FOR COMPARISON
022000  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBA
022006  001401                                  BEQ     79$             ;BRANCH IF GOOD
022010  104042                                  ERROR   42
                                                                        ;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
                                                                        ;RHBA SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHBA
022012                          79$:
                                        ;CHECK THAT RHBAE HAS 0
022012  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

022020  017737  162104  001126          MOV     @RHBAE,$BDDAT           ;READ RHBAE FOR COMPARISON
022026  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBAE
022034  001401                                  BEQ     81$             ;BRANCH IF GOOD
022036  104043                                  ERROR   43
                                                                        ;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
                                                        ;RHBAE SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHBAE
022040                                  81$:
                                                ;CHECK THAT RHWC HAS 0
022040  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

022046  017737  162010  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
022054  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
022062  001401                          BEQ     83$             ;BRANCH IF GOOD
022064  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
022066                                  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
                                        ;;*****************************************************************
022066  000004                          TST22:  SCOPE
022070  012706  001000                  MOV     #STACK,SP       ;RESET STACK
022074  012737  000022  004656          MOV     #22,@#TSTNM     ;SAVE TEST NUMBER

022102  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
```

```
                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;WRITE A ONE INTO RHDA - DISK ADDRESS REGISTER
                                        ;THIS REGISTER IN TAPE DRIVES IS CALLED "RHFC FRAME COUNT"
                                        ;AFTER WRITING THIS "1" THEN ON READING THIS REGISTER
                                        ;THE CONTROL PARITY LINE WILL HAVE ZERO
022106  012777  000001  161752          MOV     #BIT0,@RHDA     ;WRITE "1" INTO RHDA
                                                ;IN TAPE DRIVES CALLED RHFC

                                        ;INVERT THE PARITY TO BE CHECKED BY SETTING "PAT" BIT #4 IN RHCS2
022114  052777  000020  161746          BIS     #PAT,@RHCS2     ;SET PAT IN RHCS2


                                        ;CHECK THAT RHDA HAS BIT0
022122  012737  000001  001124          MOV     #BIT0,$GDDAT    ;GET GOOD = 1

022130  017737  161732  001126          MOV     @RHDA,$BDDAT    ;READ RHDA FOR COMPARISON
022136  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                ;DATA WITH DATA READ FROM
                                                ;RHDA
022144  001401                          BEQ     65$             ;BRANCH IF GOOD
022146  104045                          ERROR   45
                                                ;AFTER AN RH CLEAR "1"
                                                ;WAS WRITTEN INTO ADDRESS REGISTER
                                                ;(FRAME COUNT IN TAPE)
                                                ;PAT IN RHCS2 WAS SET
                                                ;THEN THE ADDRESS REGISTER
                                                ;WAS READ BACK
                                                ;RHDA SHOULD HAVE BIT0
                                                ;=1
                                                ;BUT CONTAINED WHAT IS
                                                ;GIVEN IN BAD RHDA
022150                          65$:


                                        ;CHECK THAT RHCS1 HAS 4200
022150  012737  004200  001124          MOV     #4200,$GDDAT    ;GET GOOD = 124000

022156  017737  161676  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022164  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                ;DATA WITH DATA READ FROM
                                                ;RHCS1
022172  001401                          BEQ     67$             ;BRANCH IF GOOD
022174  104046                          ERROR   46
                                                ;AFTER AN RH CLEAR "1"
                                                ;WAS WRITTEN INTO ADDRESS REGISTER
                                                ;(FRAME COUNT IN TAPE)
                                                ;PAT IN RHCS2 WAS SET
                                                ;THEN THE ADDRESS REGISTER
                                                ;WAS READ BACK
                                                ;RHCS1 SHOULD HAVE 4200
                                                ;=124000
                                                ;BUT CONTAINED WHAT IS
                                                ;GIVEN IN BAD RHCS1
022176                          67$:
                                        ;CHECK THAT RHCS2 HAS IR!PAT
022176  012737  000120  001124          MOV     #IR!PAT,$GDDAT  ;GET GOOD = 120
```

```
7241  022204  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7242
7243  022212  017737  161652  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7244  022220  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7245                                                                  ;DATA WITH DATA READ FROM
7246                                                                  ;RHCS2
7247  022226  001401                          BEQ     69$             ;BRANCH IF GOOD
7248  022230  104047                          ERROR   47
7249                                                                  ;AFTER AN RH CLEAR "1"
7250                                                                  ;WAS WRITTEN INTO ADDRESS REGISTER
7251                                                                  ;(FRAME COUNT IN TAPE)
7252                                                                  ;PAT IN RHCS2 WAS SET
7253                                                                  ;THEN THE ADDRESS REGISTER
7254                                                                  ;WAS READ BACK
7255                                                                  ;RHCS2 SHOULD HAVE IR!PAT
7256                                                                  ;=120
7257                                                                  ;TOGETHER WITH UNIT NUMBER
7258                                                                  ;BUT CONTAINED WHAT IS
7259                                                                  ;GIVEN IN BAD RHCS2
7260  022232                          69$:
7261                                          ;CHECK THAT RHCS3 HAS 0
7262  022232  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
7263
7264  022240  017737  161666  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7265  022246  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7266                                                                  ;DATA WITH DATA READ FROM
7267                                                                  ;RHCS3
7268  022254  001401                          BEQ     71$             ;BRANCH IF GOOD
7269  022256  104050                          ERROR   50
7270                                                                  ;AFTER AN RH CLEAR "1"
7271                                                                  ;WAS WRITTEN INTO ADDRESS REGISTER
7272                                                                  ;(FRAME COUNT IN TAPE)
7273                                                                  ;PAT IN RHCS2 WAS SET
7274                                                                  ;THEN THE ADDRESS REGISTER
7275                                                                  ;WAS READ BACK
7276                                                                  ;RHCS3 SHOULD HAVE 0
7277                                                                  ;BUT CONTAINED WHAT IS
7278                                                                  ;GIVEN IN BAD RHCS3
7279  022260                          71$:
7280                                          ;CHECK THAT RHBA HAS 0
7281  022260  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
7282
7283  022266  017737  161572  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
7284  022274  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7285                                                                  ;DATA WITH DATA READ FROM
7286                                                                  ;RHBA
7287  022302  001401                          BEQ     73$             ;BRANCH IF GOOD
7288  022304  104051                          ERROR   51
7289                                                                  ;AFTER AN RH CLEAR "1"
7290                                                                  ;WAS WRITTEN INTO ADDRESS REGISTER
7291                                                                  ;(FRAME COUNT IN TAPE)
7292                                                                  ;PAT IN RHCS2 WAS SET
7293                                                                  ;THEN THE ADDRESS REGISTER
7294                                                                  ;WAS READ BACK
7295                                                                  ;RHBA SHOULD HAVE 0
7296                                                                  ;BUT CONTAINED WHAT IS
```

```
                                                        ;GIVEN IN BAD RHBA
022306                                  73$:
                                        ;CHECK THAT RHBAE HAS 0
022306  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

022314  017737  161610  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
022322  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
022330  001401                          BEQ     75$             ;BRANCH IF GOOD
022332  104052                          ERROR   52

                                                                ;AFTER AN RH CLEAR "1"
                                                                ;WAS WRITTEN INTO ADDRESS REGISTER
                                                                ;(FRAME COUNT IN TAPE)
                                                                ;PAT IN RHCS2 WAS SET
                                                                ;THEN THE ADDRESS REGISTER
                                                                ;WAS READ BACK
                                                                ;RHBAE SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
022334                                  75$:
                                        ;CHECK THAT RHWC HAS 0
022334  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

022342  017737  161514  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
022350  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
022356  001401                          BEQ     77$             ;BRANCH IF GOOD
022360  104053                          ERROR   53

                                                                ;AFTER AN RH CLEAR "1"
                                                                ;WAS WRITTEN INTO ADDRESS REGISTER
                                                                ;(FRAME COUNT IN TAPE)
                                                                ;PAT IN RHCS2 WAS SET
                                                                ;THEN THE ADDRESS REGISTER
                                                                ;WAS READ BACK
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
022362                                  77$:


                                        ;;*******************************************************************
                                        ;*TEST 23       TEST DUPLICATED A16 (RHCS1 BIT #8)

                                        ;*      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)
                                        ;*
                                        ;*
```

```
                                ;;*****************************************************************
022362  000004                 TST23:  SCOPE
022364  012706  001000                 MOV     #STACK,SP       ;RESET STACK
022370  012737  000023  004656         MOV     #23,@#TSTNM     ;SAVE TEST NUMBER
                                
022376  004737  040306                 JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                
                                        ;WRITE "1" INTO A16 IN RHCS1
022402  012777  000400  161450         MOV     #A16,@RHCS1     ;MOVE 400 INTO RHCS1
                                
                                
                                        ;CHECK THAT RHCS1 HAS A16!DVA!RDY
022410  012737  004600  001124         MOV     #A16!DVA!RDY,$GDDAT     ;GET GOOD = 4600
                                
022416  017737  161436  001126         MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022424  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
022432  001401                       BEQ     65$             ;BRANCH IF GOOD
022434  104054                       ERROR   54
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;A16 WAS WRITTEN INTO RHCS1
                                                                ;RHCS1 WAS READ
                                                                ;RHCS1 SHOULD HAVE A16!DVA!RDY
                                                                ;=4600
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
022436                          65$:
                                
                                        ;CHECK THAT RHBAE HAS BIT0
022436  012737  000001  001124         MOV     #BIT0,$GDDAT    ;GET GOOD = 1
                                
022444  017737  161460  001126         MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
022452  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
022460  001401                       BEQ     67$             ;BRANCH IF GOOD
022462  104055                       ERROR   55
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;A16 WAS WRITTEN INTO RHCS1
                                                                ;RHCS1 WAS READ
                                                                ;THEN RHBAE WAS READ
                                                                ;RHBAE SHOULD HAVE BIT0
                                                                ;=1
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHBAE
022464                          67$:
                                
                                        ;NOW MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #0)
022464  005077  161440                 CLR     @RHBAE          ;WRITE ZERO INTO RHBAE
```

```
7409
7410
7411                                        ;CHECK THAT RHBAE HAS 0
7412  022470  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
7413
7414  022476  017737  161426  001126        MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
7415  022504  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7416                                                                ;DATA WITH DATA READ FROM
7417                                                                ;RHBAE
7418  022512  001401                        BEQ     69$             ;BRANCH IF GOOD
7419  022514  104056                        ERROR   56
7420                                                                ;AFTER SETTING "CLR" BIT #5
7421                                                                ;IN RHCS2 TO INIT THE RH
7422                                                                ;A16 WAS WRITTEN INTO RHCS1
7423                                                                ;RHCS1 WAS READ
7424                                                                ;RHBAE WAS READ
7425                                                                ;THEN ZERO WAS WRITTEN
7426                                                                ;INTO RHBAE
7427                                                                ;ON READING RHBAE
7428                                                                ;RHBAE SHOULD HAVE 0
7429                                                                ;BUT CONTAINED WHAT IS
7430                                                                ;GIVEN IN BAD RHBAE
7431  022516                            69$:
7432
7433
7434                                        ;CHECK THAT RHCS1 HAS DVA!RDY
7435  022516  012737  004200  001124        MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
7436
7437  022524  017737  161330  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
7438  022532  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7439                                                                ;DATA WITH DATA READ FROM
7440                                                                ;RHCS1
7441  022540  001401                        BEQ     71$             ;BRANCH IF GOOD
7442  022542  104057                        ERROR   57
7443                                                                ;AFTER SETTING "CLR" BIT #5
7444                                                                ;IN RHCS2 TO INIT THE RH
7445                                                                ;A16 WAS WRITTEN INTO RHCS1
7446                                                                ;RHCS1 WAS READ
7447                                                                ;RHBAE WAS READ
7448                                                                ;THEN ZERO WAS WRITTEN
7449                                                                ;INTO RHBAE
7450                                                                ;RHBAE WAS READ
7451                                                                ;ON READING RHCS1
7452                                                                ;RHCS1 SHOULD HAVE DVA!RDY
7453                                                                ;=4200
7454                                                                ;BUT CONTAINED WHAT IS
7455                                                                ;GIVEN IN BAD RHCS1
7456  022544                            71$:
7457
7458
7459                                        ;;*******************************************************************
7460                                        ;*TEST 24       TEST DUPLICATED A17 (RHCS1 BIT #9)
7461
7462                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
7463                                        ;*      MOVE 400 (ONE INTO A17) IN RHCS1
7464                                        ;*      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)
                                        ;*
                                        ;*
                                        ;*************************************************************
022544  000004                  TST24:  SCOPE
022546  012706  001000                  MOV     #STACK,SP       ;RESET STACK
022552  012737  000024  004656          MOV     #24,@#TSTNM     ;SAVE TEST NUMBER
022560  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                        ;WRITE "1" INTO A17 IN RHCS1
022564  012777  001000  161266          MOV     #A17,@RHCS1     ;MOVE 1200 INTO RHCS1
                                        ;CHECK THAT RHCS1 HAS A17!DVA!RDY
022572  012737  005200  001124          MOV     #A17!DVA!RDY,$GDDAT     ;GET GOOD = 4600
022600  017737  161254  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022606  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
022614  001401                          BEQ     65$             ;BRANCH IF GOOD
022616  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
022620                          65$:
                                        ;CHECK THAT RHBAE HAS BIT1
022620  012737  000002  001124          MOV     #BIT1,$GDDAT    ;GET GOOD = 2
022626  017737  161276  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
022634  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHBAE
022642  001401                          BEQ     67$             ;BRANCH IF GOOD
022644  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
022646                          67$:

                                        ;NOW MOVE 0 INTO RHBAE (WRITING 0 INTO RHBAE BIT #0)
022646  005077  161256                CLR     @RHBAE          ;WRITE ZERO INTO RHBAE

                                        ;CHECK THAT RHBAE HAS 0
022652  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0

022660  017737  161244  001126        MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
022666  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FRO
                                                        ;RHBAE
022674  001401                        BEQ     69$             ;BRANCH IF GOOD
022676  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
022700                          69$:

                                        ;CHECK THAT RHCS1 HAS DVA!RDY
022700  012737  004200  001124        MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

022706  017737  161146  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022714  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
022722  001401                        BEQ     71$             ;BRANCH IF GOOD
022724  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
022726                          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)
                                        ;*
                                        ;*
                                        ;;*****************************************************************
022726  000004                    TST25:  SCOPE
022730  012706  001000                    MOV     #STACK,SP       ;RESET STACK
022734  012737  000025  004656            MOV     #25,@#TSTNM     ;SAVE TEST NUMBER

022742  004737  040306                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                  ;SETUP UNIT NUBER
                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;WRITE "1" INTO IE IN RHCS1
022746  012777  000100  161104            MOV     #IE,@RHCS1      ;MOVE 100 INTO RHCS1


                                        ;CHECK THAT RHCS1 HAS IE!DVA!RDY
022754  012737  004300  001124            MOV     #IE!DVA!RDY,$GDDAT      ;GET GOOD = 4300

022762  017737  161072  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
022770  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                  ;DATA WITH DATA READ FROM
                                                                  ;RHCS1
022776  001401                            BEQ     65$             ;BRANCH IF GOOD
023000  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
023002                              65$:


                                        ;CHECK THAT RHCS3 HAS IE
023002  012737  000100  001124            MOV     #IE,$GDDAT      ;GET GOOD = 1

023010  017737  161116  001126            MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
023016  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                  ;DATA WITH DATA READ FROM
```

```
                                                        ;RHCS3
023024  001401                          BEQ     67$     ;BRANCH IF GOOD
023026  104065                          ERROR   65
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;IE WAS WRITTEN INTO RHCS1
                                                        ;RHCS1 WAS READ
                                                        ;THEN RHCS3 WAS READ
                                                        ;RHCS3 SHOULD HAVE IE
                                                        ;=1
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS3
023030                          67$:

                                        ;NOW MOVE 0 INTO RHCS3 (WRITING 0 INTO RHCS3 BIT #1)
023030  005077  161076                  CLR     @RHCS3  ;WRITE ZERO INTO RHCS3

                                        ;CHECK THAT RHCS3 HAS 0
023034  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

023042  017737  161064  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
023050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
023056  001401                          BEQ     69$     ;BRANCH IF GOOD
023060  104066                          ERROR   66
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;IE WAS WRITTEN INTO RHCS1
                                                        ;RHCS1 WAS READ
                                                        ;RHCS3 WAS READ
                                                        ;THEN ZERO WAS WRITTEN
                                                        ;INTO RHCS3
                                                        ;ON READING RHCS3
                                                        ;RHCS3 SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHCS3
023062                          69$:

                                        ;CHECK THAT RHCS1 HAS DVA!RDY
023062  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

023070  017737  160764  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
023076  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
023104  001401                          BEQ     71$     ;BRANCH IF GOOD
023106  104067                          ERROR   67
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;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
023110                          71$:


                                ;;*****************************************************************
                                ;*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
                                ;*
                                ;;*****************************************************************
023110  000004                TST26:  SCOPE
023112  012706  CJ1000              MOV     #STACK,SP       ;RESET STACK
023116  012737  000026  004656      MOV     #26,@#TSTNM     ;SAVE TEST NUMBER

023124  004737  040306              JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                            ;SETUP UNIT NUBER
                                                            ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                ;SET UP RH FOR A 10 WORD WRITE
023130  012777  177766  160724  MOV     #-10.,@RHWC     ;WORD COUNT REGISTER=10
023136  012777  004210  160720  MOV     #WRFROM,@RHBA   ;BUS ADDRESS REGISTER=WRITBUF
                                ;WRITE 10 WORDS FROM "WRFROM" INTO
                                ;WHAT EVER RH DEVICE IS AVAILABLE
023144  012737  177777  004644  MOV     #-1,NOGO        ;DO NOT GIVE GO IN COMMAND ROUTINE
023152  004077  161462          JSR     R0,@COMND       ;GO TO DO COMMAND
023156  004154                  WRIDAT                  ;WRITE DATA
023160  052777  000001  160672  BIS     #GO,@RHCS1      ;SET GO
023166  000240                  NOP                     ;TWO
023170  000240                  NOP                     ;MICROSECOND
023172  000240                  NOP                     ;DELAY
023174  052777  000001  160656  BIS     #GO,@RHCS1      ;SET GO WITHOUT TIME TO COMPLETE LAST GO


                                ;CHECK THAT RHCS1 HAS SC!DVA!TRE!61
023202  012737  144061  001124  MOV     #SC!DVA!TRE!61,$GDDAT   ;GET GOOD = 144000

023210  017737  160644  001126  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
023216  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
```

```
7745  023224  001401                        BEQ     65$             ;BRANCH IF GOOD
7746  023226  104070                        ERROR   70
7747                                                                  ;AFTER SETTING "CLR" BIT #5
7748                                                                  ;IN RHCS2 TO INIT THE RH
7749                                                                  ;AND GIVING TWO SUCCESSIVE
7750                                                                  ;"GO" WITHOUT GIVING TIME FOR
7751                                                                  ;THE FIRST TO COMPLETE
7752                                                                  ;RHCS1 SHOULD HAVE SC!DVA!TRE!61
7753                                                                  ;=144000
7754                                                                  ;BUT CONTAINED WHAT IS
7755                                                                  ;GIVEN IN BAD RHCS1
7756  023230                            65$:
7757                                    ;CHECK THAT RHCS2 HAS PGE!OR
7758  023230  012737  002200  001124        MOV     #PGE!OR,$GDDAT  ;GET GOOD = 2000
7759  023236  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7760
7761  023244  017737  160620  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7762  023252  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7763                                                                  ;DATA WITH DATA READ FROM
7764                                                                  ;RHCS2
7765  023260  001401                        BEQ     67$             ;BRANCH IF GOOD
7766  023262  104071                        ERROR   71
7767                                                                  ;AFTER SETTING "CLR" BIT #5
7768                                                                  ;IN RHCS2 TO INIT THE RH
7769                                                                  ;AND GIVING TWO SUCCESSIVE
7770                                                                  ;"GO" WITHOUT GIVING TIME FOR
7771                                                                  ;THE FIRST TO COMPLETE
7772                                                                  ;RHCS2 SHOULD HAVE PGE!OR
7773                                                                  ;=2000
7774                                                                  ;TOGETHER WITH UNIT NUMBER
7775                                                                  ;BUT CONTAINED WHAT IS
7776                                                                  ;GIVEN IN BAD RHCS2
7777  023264                            67$:
7778                                    ;CHECK THAT RHCS3 HAS DBL
7779  023264  012737  002000  001124        MOV     #DBL,$GDDAT     ;GET GOOD = 0
7780
7781  023272  017737  160634  001126        MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7782  023300  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7783                                                                  ;DATA WITH DATA READ FROM
7784                                                                  ;RHCS3
7785  023306  001401                        BEQ     69$             ;BRANCH IF GOOD
7786  023310  104072                        ERROR   72
7787                                                                  ;AFTER SETTING "CLR" BIT #5
7788                                                                  ;IN RHCS2 TO INIT THE RH
7789                                                                  ;AND GIVING TWO SUCCESSIVE
7790                                                                  ;"GO" WITHOUT GIVING TIME FOR
7791                                                                  ;THE FIRST TO COMPLETE
7792                                                                  ;RHCS3 SHOULD HAVE DBL
7793                                                                  ;BUT CONTAINED WHAT IS
7794                                                                  ;GIVEN IN BAD RHCS3
7795  023312                            69$:
7796
7797
7798                                    ;;*****************************************************************
7799                                    ;*TEST 27        RHCS2 - BUS ADDRESS INHIBIT BIT #3
7800
```

```
                                        ;*      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
                                        ;;**************************************************************
023312  000004                          TST27:  SCOPE
023314  012706  001000                          MOV     #STACK,SP       ;RESET STACK
023320  012737  000027  004656                  MOV     #27,@#TSTNM     ;SAVE TEST NUMBER

023326  004737  040306                          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
023332  012777  177766  160522                  MOV     #-10.,@RHWC     ;WORD COUNT REGISTER=10
023340  052777  000010  160522                  BIS     #BAI,@RHCS2     ;BUS ADDRESS INHIBIT IN RHCS2
023346  012777  004210  160510                  MOV     #WRFROM,@RHBA   ;BUS ADDRESS REGISTER=WRFROM
023354  004077  161260                          JSR     R0,@COMND       ;GO TO DO COMMAND
023360  004154                                  WRIDAT                  ;WRITE DATA


                                                ;CHECK THAT RHCS1 HAS DVA!RDY!60
023362  012737  004260  001124                  MOV     #DVA!RDY!60,$GDDAT      ;GET GOOD = 4252

023370  017737  160464  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
023376  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS1
023404  001401                                  BEQ     65$             ;BRANCH IF GOOD
023406  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
```

```
7857  023410                          65$:
7858                                  ;CHECK THAT RHCS2 HAS BAI!IR
7859  023410  012737  000110  001124  MOV     #BAI!IP,$GDDAT  ;GET GOOD = 10
7860  023416  053737  005010  001124  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
7861
7862  023424  017737  160440  001126  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
7863  023432  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7864                                                          ;DATA WITH DATA READ FROM
7865                                                          ;RHCS2
7866  023440  001401                  BEQ     67$             ;BRANCH IF GOOD
7867  023442  104077                  ERROR   77
7868                                                          ;AFTER SETTING "CLR" BIT #5
7869                                                          ;IN RHCS2 TO INIT THE RH
7870                                                          ;A 2 WORD WRITE WAS DONE
7871                                                          ;WITH BAI BIT IN RHCS2 SET
7872                                                          ;AT END OF WRITE
7873                                                          ;RHCS2 SHOULD HAVE BAI!IR
7874                                                          ;=10
7875                                                          ;TOGETHER WITH UNIT NUMBER
7876                                                          ;BUT CONTAINED WHAT IS
7877                                                          ;GIVEN IN BAD RHCS2
7878  023444                          67$:
7879                                  ;CHECK THAT RHCS3 HAS 0
7880  023444  012737  000000  001124  MOV     #0,$GDDAT       ;GET GOOD = 0
7881
7882  023452  017737  160454  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
7883  023460  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7884                                                          ;DATA WITH DATA READ FROM
7885                                                          ;RHCS3
7886  023466  001401                  BEQ     69$             ;BRANCH IF GOOD
7887  023470  104100                  ERROR   100
7888                                                          ;AFTER SETTING "CLR" BIT #5
7889                                                          ;IN RHCS2 TO INIT THE RH
7890                                                          ;A 2 WORD WRITE WAS DONE
7891                                                          ;WITH BAI BIT IN RHCS2 SET
7892                                                          ;AT END OF WRITE
7893                                                          ;RHCS3 SHOULD HAVE 0
7894                                                          ;BUT CONTAINED WHAT IS
7895                                                          ;GIVEN IN BAD RHCS3
7896  023472                          69$:
7897                                  ;CHECK THAT RHBA HAS WRFROM
7898  023472  012737  004210  001124  MOV     #WRFROM,$GDDAT  ;GET GOOD = 0
7899
7900  023500  017737  160360  001126  MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
7901  023506  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
7902                                                          ;DATA WITH DATA READ FROM
7903                                                          ;RHBA
7904  023514  001401                  BEQ     71$             ;BRANCH IF GOOD
7905  023516  104101                  ERROR   101
7906                                                          ;AFTER SETTING "CLR" BIT #5
7907                                                          ;IN RHCS2 TO INIT THE RH
7908                                                          ;A 2 WORD WRITE WAS DONE
7909                                                          ;WITH BAI BIT IN RHCS2 SET
7910                                                          ;AT END OF WRITE
7911                                                          ;RHBA SHOULD HAVE WRFROM
7912                                                          ;BUT CONTAINED WHAT IS
```

```
                                                                        ;GIVEN IN BAD RHBA
023520                                  71$:
                                        ;CHECK THAT RHBAE HAS 0
023520  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

023526  017737  160376  001126          MOV     @RHBAE,$BDDAT           ;READ RHBAE FOR COMPARISON
023534  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHBAE
023542  001401                          BEQ     73$                     ;BRANCH IF GOOD
023544  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
023546                                  73$:
                                        ;CHECK THAT RHWC HAS 0
023546  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

023554  017737  160302  001126          MOV     @RHWC,$BDDAT            ;READ RHWC FOR COMPARISON
023562  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHWC
023570  001401                          BEQ     75$                     ;BRANCH IF GOOD
023572  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
023574                                  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

                                        ;;*****************************************************************
023574  000004                  ;ST30:  SCOPE
023576  012706  001000                  MOV     #STACK,SP               ;RESET STACK
```

```
 7969  023602  012737  000030  004656          MOV     #30,@#TSTNM     ;SAVE TEST NUMBER
 7970
 7971  023610  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
 7972                                                                  ;SETUP UNIT NUBER
 7973                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
 7974  023614  012737  000000  004662          MOV     #0,IP           ;IPCK BIT NO. FOR PRINTOUT
 7975
 7976  023622  052777  000001  160302          BIS     #IPCK0,@RHCS3 ;SET IPCK0 IN RHCS3-BIT #0
 7977  023630  005037  004210                  CLR     WRFROM          ;WRITE ZERO INTO LOCATION
 7978  023634  013777  004210  160240          MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
 7979
 7980                                          ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
 7981  023642  012737  144200  001124          MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200
 7982
 7983  023650  017737  160204  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 7984  023656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 7985                                                                  ;DATA WITH DATA READ FROM
 7986                                                                  ;RHCS1
 7987  023664  001401                          BEQ     65$             ;BRANCH IF GOOD
 7988  023666  104140                          ERROR   140
 7989                                                                  ;AFTER SETTING "CLR" BIT #5
 7990                                                                  ;IN RHCS2 TO INIT THE RH
 7991                                                                  ;IPCK0 BIT #0 RHCS3 WAS SET
 7992                                                                  ;ZEROS WERE MOVED INTO RHDB
 7993                                                                  ;RHCS1 SHOULD HAVE SC!TRE!DVA!RDY
 7994                                                                  ;=144200
 7995                                                                  ;BUT CONTAINED WHAT IS
 7996                                                                  ;GIVEN IN BAD RHCS1
 7997  023670                          65$:
 7998  023670  017737  160206  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE
 7999
 8000
 8001                                          ;CHECK THAT RHCS3 HAS 1
 8002  023676  012737  000001  001124          MOV     #1,$GDDAT       ;GET GOOD = 1
 8003
 8004  023704  017737  160222  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 8005  023712  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8006                                                                  ;DATA WITH DATA READ FROM
 8007                                                                  ;RHCS3
 8008  023720  001401                          BEQ     67$             ;BRANCH IF GOOD
 8009  023722  104104                          ERROR   104
 8010                                                                  ;AFTER SETTING "CLR" BIT #5
 8011                                                                  ;IN RHCS2 TO INIT THE RH
 8012                                                                  ;IPCK0 BIT #0 RHCS3 WAS SET
 8013                                                                  ;ZEROS WERE MOVED INTO
 8014                                                                  ;RHDB
 8015                                                                  ;RHDB WAS READ THEN
 8016                                                                  ;RHCS3 SHOULD HAVE 1
 8017                                                                  ;=1
 8018                                                                  ;BUT CONTAINED WHAT IS
 8019                                                                  ;GIVEN IN BAD RHCS3
 8020  023724                          67$:
 8021
 8022                                          ;CHECK THAT RHCS2 HAS MDPE!IR
 8023  023724  012737  000500  001124          MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
 8024  023732  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
8025
8026  023740  017737  160124  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8027  023746  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8028                                                                  ;DATA WITH DATA READ FROM
8029                                                                  ;RHCS2
8030  023754  001401                          BEQ     69$             ;BRANCH IF GOOD
8031  023756  104105                          ERROR   105
8032                                                                  ;AFTER SETTING "CLR" BIT #5
8033                                                                  ;IN RHCS2 TO INIT THE RH
8034                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8035                                                                  ;ZEROS WERE MOVED INTO
8036                                                                  ;RHDB
8037                                                                  ;RHDB WAS READ THEN
8038                                                                  ;RHCS2 SHOULD HAVE MDPE!IR
8039                                                                  ;=500
8040                                                                  ;TOGETHER WITH UNIT NUMBE.
8041                                                                  ;BUT CONTAINED WHAT IS
8042                                                                  ;GIVEN IN BAD RHCS2
8043  023760                          69$:
8044
8045
8046  023760  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
8047                                                                  ;SETUP UNIT NUBER
8048                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8049
8050
8051                                          ;CHECK THAT RHCS1 HAS RDY!DVA
8052  023764  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
8053
8054  023772  017737  160062  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8055  024000  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8056                                                                  ;DATA WITH DATA READ FROM
8057                                                                  ;RHCS1
8058  024006  001401                          BEQ     71$             ;BRANCH IF GOOD
8059  024010  104106                          ERROR   106
8060                                                                  ;AFTER SETTING "CLR" BIT #5
8061                                                                  ;IN RHCS2 TO INIT THE RH
8062                                                                  ;IPCKO BIT #0 RHCS3 WAS SET
8063                                                                  ;ZEROS WERE MOVED INTO
8064                                                                  ;RHDB
8065                                                                  ;RHDB WAS READ THEN
8066                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8067                                                                  ;WAS GIVEN THIS SHOULD
8068                                                                  ;CLEAR ALL ERRORS
8069                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
8070                                                                  ;=4200
8071                                                                  ;BUT CONTAINED WHAT IS
8072                                                                  ;GIVEN IN BAD RHCS1
8073  024012                          71$:
8074                                          ;CHECK THAT RHCS2 HAS IR
8075  024012  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
8076  024020  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8077
8078  024026  017737  160036  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8079  024034  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8080                                                                  ;DATA WITH DATA READ FROM
```

```
8081
8082  024042  001401                          BEQ     73$             ;RHCS2
8083  024044  104107                          ERROR   107             ;BRANCH IF GOOD
8084                                                                  ;AFTER SETTING "CLR" BIT #5
8085                                                                  ;IN RHCS2 TO INIT THE RH
8086                                                                  ;IPCKD BIT #0 RHCS3 WAS SET
8087                                                                  ;ZEROS WERE MOVED INTO
8088                                                                  ;RHDB
8089                                                                  ;RHDB WAS READ THEN
8090                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8091                                                                  ;WAS GIVEN THIS SHOULD
8092                                                                  ;CLEAR ALL ERRORS
8093                                                                  ;RHCS2 SHOULD HAVE IR
8094                                                                  ;=100
8095                                                                  ;TOGETHER WITH UNIT NUMBER
8096                                                                  ;BUT CONTAINED WHAT IS
8097                                                                  ;GIVEN IN BAD RHCS2
8098  024046                          73$:
8099                                          ;CHECK THAT RHCS3 HAS 0
8100  024046  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8101
8102  024054  017737  160052  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8103  024062  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8104                                                                  ;DATA WITH DATA READ FROM
8105                                                                  ;RHCS3
8106  024070  001401                          BEQ     75$             ;BRANCH IF GOOD
8107  024072  104110                          ERROR   110
8108                                                                  ;AFTER SETTING "CLR" BIT #5
8109                                                                  ;IN RHCS2 TO INIT THE RH
8110                                                                  ;IPCKD BIT #0 RHCS3 WAS SET
8111                                                                  ;ZEROS WERE MOVED INTO
8112                                                                  ;RHDB
8113                                                                  ;RHDB WAS READ THEN
8114                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8115                                                                  ;WAS GIVEN THIS SHOULD
8116                                                                  ;CLEAR ALL ERRORS
8117                                                                  ;RHCS3 SHOULD HAVE 0
8118                                                                  ;BUT CONTAINED WHAT IS
8119                                                                  ;GIVEN IN BAD RHCS3
8120  024074                          75$:
8121                                          ;CHECK THAT RHBA HAS 0
8122  024074  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8123
8124  024102  017737  157756  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
8125  024110  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8126                                                                  ;DATA WITH DATA READ FROM
8127                                                                  ;RHBA
8128  024116  001401                          BEQ     77$             ;BRANCH IF GOOD
8129  024120  104111                          ERROR   111
8130                                                                  ;AFTER SETTING "CLR" BIT #5
8131                                                                  ;IN RHCS2 TO INIT THE RH
8132                                                                  ;IPCKD BIT #0 RHCS3 WAS SET
8133                                                                  ;ZEROS WERE MOVED INTO
8134                                                                  ;RHDB
8135                                                                  ;RHDB WAS READ THEN
8136                                                                  ;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
024122                                  77$:
                                        ;CHECK THAT RHBAE HAS 0
024122  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

024130  017737  157774  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
024136  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBAE
024144  001401                          BEQ     79$             ;BRANCH IF GOOD
024146  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
024150                                  79$:
                                        ;CHECK THAT RHWC HAS 0
024150  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

024156  017737  157700  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
024164  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
024172  001401                          BEQ     81$             ;BRANCH IF GOOD
024174  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
024176                                  81$:


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

T31 RHSC2 MDPE BIT #8 AND RHCS3 IPCK1 BIT #1 SEQ 0156

```
                                ;*      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
                                ;;**************************************************************
024176  000004                  ↑ST31:  SCOPE
024200  012706  001000                  MOV     #STACK,SP       ;RESET STACK
024204  012737  000031  004656          MOV     #31,@#↑STNM     ;SAVE TEST NUMBER
024212  004737  040306          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
024216  012737  000001  004662  MOV     #1,IP           ;IPCK BIT NO. FOR PRINTOUT
024224  052777  000002  157700  BIS     #IPCK1,@RHCS3 ;SET IPCK1 IN RHCS3-BIT #1
024232  005037  004210          CLR     WRFROM          ;WRITE ZERO INTO LOCATION
024236  013777  004210  157636  MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
                                ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
024244  012737  144200  001124  MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200
024252  017737  157602  001126  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
024260  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
024266  001401                  BEQ     65$             ;BRANCH IF GOOD
024270  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
024272                  65$:
024272  017737  157604  001200  MOV     @RHDB,$TMP1     ;READ RHDB ONCE
                                ;CHECK THAT RHCS3 HAS 2
024300  012737  000002  001124  MOV     #2,$GDDAT       ;GET GOOD = 2
024306  017737  157620  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
024314  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
024322  001401                  BEQ     67$             ;BRANCH IF GOOD
024324  104104                  ERROR   104
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;IPCK1 BIT #1 RHCS3 WAS SET
```

```
8249                                                    ;ZEROS WERE MOVED INTO
8250                                                    ;RHDB
8251                                                    ;RHDB WAS READ THEN
8252                                                    ;RHCS3 SHOULD HAVE 2
8253                                                    ;=2
8254                                                    ;BUT CONTAINED WHAT IS
8255                                                    ;GIVEN IN BAD RHCS3
8256  024326                            67$:
8257
8258                                    ;CHECK THAT RHCS2 HAS MDPE!IR
8259  024326  012737  000500  001124    MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
8260  024334  053737  005010  001124    BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8261
8262  024342  017737  157522  001126    MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8263  024350  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8264                                                    ;DATA WITH DATA READ FRO
8265                                                    ;RHCS2
8266  024356  001401                    BEQ     69$             ;BRANCH IF GOOD
8267  024360  104105                    ERROR   105
8268                                                    ;AFTER SETTING "CLR" BIT #5
8269                                                    ;IN RHCS2 TO INIT THE RH
8270                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8271                                                    ;ZEROS WERE MOVED INTO
8272                                                    ;RHDB
8273                                                    ;RHDB WAS READ THEN
8274                                                    ;RHCS2 SHOULD HAVE MDPE!IR
8275                                                    ;=500
8276                                                    ;TOGETHER WITH UNIT NUMBER
8277                                                    ;BUT CONTAINED WHAT IS
8278                                                    ;GIVEN IN BAD RHCS2
8279  024362                            69$:
8280
8281
8282  024362  004737  040306            JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
8283                                                    ;SETUP UNIT NUBER
8284                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8285
8286
8287                                    ;CHECK THAT RHCS1 HAS RDY!DVA
8288  024366  012737  004200  001124    MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
8289
8290  024374  017737  157460  001126    MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8291  024402  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8292                                                    ;DATA WITH DATA READ FROM
8293                                                    ;RHCS1
8294  024410  001401                    BEQ     71$             ;BRANCH IF GOOD
8295  024412  104106                    ERROR   106
8296                                                    ;AFTER SETTING "CLR" BIT #5
8297                                                    ;IN RHCS2 TO INIT THE RH
8298                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8299                                                    ;ZEROS WERE MOVED INTO
8300                                                    ;RHDB
8301                                                    ;RHDB WAS READ THEN
8302                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8303                                                    ;WAS GIVEN THIS SHOULD
8304                                                    ;CLEAR ALL ERRORS
```

```
8305                                                    ;RHCS1 SHOULD HAVE RDY!DVA
8306                                                    ;=4200
8307                                                    ;BUT CONTAINED WHAT IS
8308                                                    ;GIVEN IN BAD RHCS1
8309  024414                            71$:
8310                                    ;CHECK THAT RHCS2 HAS IR
8311  024414  012737  000100  001124    MOV     #IR,$GDDAT      ;GET GOOD = 100
8312  024422  053737  005010  001124    BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8313
8314  024430  017737  157434  001126    MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8315  024436  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8316                                                    ;DATA WITH DATA READ FROM
8317                                                    ;RHCS2
8318  024444  001401                    BEQ     73$             ;BRANCH IF GOOD
8319  024446  104107                    ERROR   107
8320                                                    ;AFTER SETTING "CLR" BIT #5
8321                                                    ;IN RHCS2 TO INIT THE RH
8322                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8323                                                    ;ZEROS WERE MOVED INTO
8324                                                    ;RHDB
8325                                                    ;RHDB WAS READ THEN
8326                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8327                                                    ;WAS GIVEN THIS SHOULD
8328                                                    ;CLEAR ALL ERRORS
8329                                                    ;RHCS2 SHOULD HAVE IR
8330                                                    ;=100
8331                                                    ;TOGETHER WITH UNIT NUMBER
8332                                                    ;BUT CONTAINED WHAT IS
8333                                                    ;GIVEN IN BAD RHCS2
8334  024450                            73$:
8335                                    ;CHECK THAT RHCS3 HAS 0
8336  024450  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD = 0
8337
8338  024456  017737  157450  001126    MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8339  024464  023737  001124  001126    CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8340                                                    ;DATA WITH DATA READ FROM
8341                                                    ;RHCS3
8342  024472  001401                    BEQ     75$             ;BRANCH IF GOOD
8343  024474  104110                    ERROR   110
8344                                                    ;AFTER SETTING "CLR" BIT #5
8345                                                    ;IN RHCS2 TO INIT THE RH
8346                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8347                                                    ;ZEROS WERE MOVED INTO
8348                                                    ;RHDB
8349                                                    ;RHDB WAS READ THEN
8350                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8351                                                    ;WAS GIVEN THIS SHOULD
8352                                                    ;CLEAR ALL ERRORS
8353                                                    ;RHCS3 SHOULD HAVE 0
8354                                                    ;BUT CONTAINED WHAT IS
8355                                                    ;GIVEN IN BAD RHCS3
8356  024476                            75$:
8357                                    ;CHECK THAT RHBA HAS 0
8358  024476  012737  000000  001124    MOV     #0,$GDDAT       ;GET GOOD = 0
8359
8360  024504  017737  157354  001126    MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
```

```
8361   024512  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8362                                                                    ;DATA WITH DATA READ FROM
8363                                                                    ;RHBA
8364   024520  001401                          BEQ     77$             ;BRANCH IF GOOD
8365   024522  104111                          ERROR   111
8366                                                                    ;AFTER SETTING "CLR" BIT #5
8367                                                                    ;IN RHCS2 TO INIT THE RH
8368                                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8369                                                                    ;ZEROS WERE MOVED INTO
8370                                                                    ;RHDB
8371                                                                    ;RHDB WAS READ THEN
8372                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8373                                                                    ;WAS GIVEN THIS SHOULD
8374                                                                    ;CLEAR ALL ERRORS
8375                                                                    ;RHBA SHOULD HAVE 0
8376                                                                    ;BUT CONTAINED WHAT IS
8377                                                                    ;GIVEN IN BAD RHBA
8378   024524                          77$:
8379                                          ;CHECK THAT RHBAE HAS 0
8380   024524  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8381
8382   024532  017737  157372  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
8383   024540  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8384                                                                    ;DATA WITH DATA READ FROM
8385                                                                    ;RHBAE
8386   024546  001401                          BEQ     79$             ;BRANCH IF GOOD
8387   024550  104112                          ERROR   112
8388                                                                    ;AFTER SETTING "CLR" BIT #5
8389                                                                    ;IN RHCS2 TO INIT THE RH
8390                                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8391                                                                    ;ZEROS WERE MOVED INTO
8392                                                                    ;RHDB
8393                                                                    ;RHDB WAS READ THEN
8394                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8395                                                                    ;WAS GIVEN THIS SHOULD
8396                                                                    ;CLEAR ALL ERRORS
8397                                                                    ;RHBAE SHOULD HAVE 0
8398                                                                    ;BUT CONTAINED WHAT IS
8399                                                                    ;GIVEN IN BAD RHBAE
8400   024552                          79$:
8401                                          ;CHECK THAT RHWC HAS 0
8402   024552  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8403
8404   024560  017737  157276  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
8405   024566  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8406                                                                    ;DATA WITH DATA READ FROM
8407                                                                    ;RHWC
8408   024574  001401                          BEQ     81$             ;BRANCH IF GOOD
8409   024576  104113                          ERROR   113
8410                                                                    ;AFTER SETTING "CLR" BIT #5
8411                                                                    ;IN RHCS2 TO INIT THE RH
8412                                                                    ;IPCK1 BIT #1 RHCS3 WAS SET
8413                                                                    ;ZEROS WERE MOVED INTO
8414                                                                    ;RHDB
8415                                                                    ;RHDB WAS READ THEN
8416                                                                    ;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
024600                          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
                                ;;*****************************************************************
024600  000004                TST32:  SCOPE
024602  012706  001000                MOV     #STACK,SP       ;RESET STACK
024606  012737  000032  004656        MOV     #32,@#TSTNM     ;SAVE TEST NUMBER

024614  004737  040306                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
024620  012737  000002  004662        MOV     #2,IP           ;IPCK BIT NO. FOR PRINTOUT

024626  052777  000004  157276        BIS     #IPCK2,@RHCS3 ;SET IPCK2 IN RHCS3-BIT #2
024634  005037  004210                CLR     WRFROM          ;WRITE ZERO INTO LOCATION
024640  013777  004210  157234        MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
024646  013777  004210  157226        MOV     WRFROM,@RHDB    ;WRITE ZERO SECOND TIME

                                ;CHECK THAT RHCS1 HAS DVA!RDY
024654  012737  004200  001124        MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200

024662  017737  157172  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
024670  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
024676  001401                        BEQ     65$             ;BRANCH IF GOOD
024700  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
024702                          65$:
```

```
 8473  024702  017737  157174  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE
 8474
 8475
 8476                                          ;CHECK THAT RHCS3 HAS 4
 8477  024710  012737  000004  001124          MOV     #4,$GDDAT       ;GET GOOD = 4
 8478
 8479  024716  017737  157210  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 8480  024724  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8481                                                                  ;DATA WITH DATA READ FROM
 8482                                                                  ;RHCS3
 8483  024732  001401                          BEQ     67$             ;BRANCH IF GOOD
 8484  024734  104104                          ERROR   104
 8485                                                                  ;AFTER SETTING "CLR" BIT #5
 8486                                                                  ;IN RHCS2 TO INIT THE RH
 8487                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
 8488                                                                  ;ZEROS WERE MOVED INTO
 8489                                                                  ;RHDB
 8490                                                                  ;RHDB WAS READ THEN
 8491                                                                  ;RHCS3 SHOULD HAVE 4
 8492                                                                  ;=4
 8493                                                                  ;BUT CONTAINED WHAT IS
 8494                                                                  ;GIVEN IN BAD RHCS3
 8495  024736                          67$:
 8496
 8497                                          ;CHECK THAT RHCS2 HAS MDPE!OR!IR
 8498  024736  012737  000700  001124          MOV     #MDPE!OR!IR,$GDDAT      ;GET GOOD = 700
 8499  024744  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 8500
 8501  024752  017737  157112  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 8502  024760  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8503                                                                  ;DATA WITH DATA READ FROM
 8504                                                                  ;RHCS2
 8505  024766  001401                          BEQ     69$             ;BRANCH IF GOOD
 8506  024770  104105                          ERROR   105
 8507                                                                  ;AFTER SETTING "CLR" BIT #5
 8508                                                                  ;IN RHCS2 TO INIT THE RH
 8509                                                                  ;IPCK2 BIT #2 RHCS3 WAS SET
 8510                                                                  ;ZEROS WERE MOVED INTO
 8511                                                                  ;RHDB
 8512                                                                  ;RHDB WAS READ THEN
 8513                                                                  ;RHCS2 SHOULD HAVE MDPE!OR!IR
 8514                                                                  ;=700
 8515                                                                  ;TOGETHER WITH UNIT NUMBER
 8516                                                                  ;BUT CONTAINED WHAT IS
 8517                                                                  ;GIVEN IN BAD RHCS2
 8518  024772                          69$:
 8519
 8520  024772  017737  157104  001202          MOV     @RHDB,$TMP2     ;READ RHDB SECOND TIME
 8521                                          ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
 8522  025000  012737  144200  001124          MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200
 8523
 8524  025006  017737  157046  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 8525  025014  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8526                                                                  ;DATA WITH DATA READ FROM
 8527                                                                  ;RHCS1
 8528  025022  001401                          BEQ     71$             ;BRANCH IF GOOD
```

```
 8529  025024  104140                          ERROR   140
 8530                                                    ;AFTER SETTING "CLR" BIT #5
 8531                                                    ;IN RHCS2 TO INIT THE RH
 8532                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
 8533                                                    ;ZEROS WERE MOVED INTO
 8534                                                    ;RHDB
 8535                                                    ;RHDB WAS READ THEN
 8536                                                    ;RHCS1 SHOULD HAVE SC!TRE!DVA!RDY
 8537                                                    ;=144200
 8538                                                    ;BUT CONTAINED WHAT IS
 8539                                                    ;GIVEN IN BAD RHCS1
 8540  025026                            71$:
 8541                                          ;CHECK THAT RHCS2 HAS MDPE!IR
 8542  025026  012737  000500  001124          MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
 8543  025034  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 8544
 8545  025042  017737  157022  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 8546  025050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8547                                                    ;DATA WITH DATA READ FROM
 8548                                                    ;RHCS2
 8549  025056  001401                          BEQ     73$             ;BRANCH IF GOOD
 8550  025060  104105                          ERROR   105
 8551                                                    ;AFTER SETTING "CLR" BIT #5
 8552                                                    ;IN RHCS2 TO INIT THE RH
 8553                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
 8554                                                    ;ZEROS WERE MOVED INTO
 8555                                                    ;RHDB
 8556                                                    ;RHDB WAS READ THEN
 8557                                                    ;RHCS2 SHOULD HAVE MDPE!IR
 8558                                                    ;=500
 8559                                                    ;TOGETHER WITH UNIT NUMBER
 8560                                                    ;BUT CONTAINED WHAT IS
 8561                                                    ;GIVEN IN BAD RHCS2
 8562  025062                            73$:
 8563                                          ;CHECK THAT RHCS3 HAS 4
 8564  025062  012737  000004  001124          MOV     #4,$GDDAT       ;GET GOOD = 4
 8565
 8566  025070  017737  157036  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 8567  025076  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8568                                                    ;DATA WITH DATA READ FROM
 8569                                                    ;RHCS3
 8570  025104  001401                          BEQ     75$             ;BRANCH IF GOOD
 8571  025106  104104                          ERROR   104
 8572                                                    ;AFTER SETTING "CLR" BIT #5
 8573                                                    ;IN RHCS2 TO INIT THE RH
 8574                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
 8575                                                    ;ZEROS WERE MOVED INTO
 8576                                                    ;RHDB
 8577                                                    ;RHDB WAS READ THEN
 8578                                                    ;RHCS3 SHOULD HAVE 4
 8579                                                    ;=4
 8580                                                    ;BUT CONTAINED WHAT IS
 8581                                                    ;GIVEN IN BAD RHCS3
 8582  025110                            75$:
 8583
 8584  025110  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
```

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



                                        ;CHECK THAT RHCS1 HAS RDY!DVA
025114  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200

025122  017737  156732  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
025130  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
025136  001401                          BEQ     77$             ;BRANCH IF GOOD
025140  104106                          ERROR   106
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;IPCK2 BIT #2 RHCS3 WAS S_T
                                                        ;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
025142                          77$:
                                        ;CHECK THAT RHCS2 HAS IR
025142  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
025150  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

025156  017737  156706  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
025164  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
025172  001401                          BEQ     79$             ;BRANCH IF GOOD
025174  104107                          ERROR   107
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;IPCK2 BIT #2 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
025176                          79$:
                                        ;CHECK THAT RHCS3 HAS 0
025176  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

025204  017737  156722  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
```

```
 8641  025212  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8642                                                                    ;DATA WITH DATA READ FROM
 8643                                                                    ;RHCS3
 8644  025220  001401                          BEQ     81$             ;BRANCH IF GOOD
 8645  025222  104110                          ERROR   110
 8646                                                                    ;AFTER SETTING "CLR" BIT #5
 8647                                                                    ;IN RHCS2 TO INIT THE RH
 8648                                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
 8649                                                                    ;ZEROS WERE MOVED INTO
 8650                                                                    ;RHDB
 8651                                                                    ;RHDB WAS READ THEN
 8652                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
 8653                                                                    ;WAS GIVEN THIS SHOULD
 8654                                                                    ;CLEAR ALL ERRORS
 8655                                                                    ;RHCS3 SHOULD HAVE 0
 8656                                                                    ;BUT CONTAINED WHAT IS
 8657                                                                    ;GIVEN IN BAD RHCS3
 8658  025224                          81$:
 8659                                          ;CHECK THAT RHBA HAS 0
 8660  025224  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 8661
 8662  025232  017737  156626  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
 8663  025240  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8664                                                                    ;DATA WITH DATA READ FROM
 8665                                                                    ;RHBA
 8666  025246  001401                          BEQ     83$             ;BRANCH IF GOOD
 8667  025250  104111                          ERROR   111
 8668                                                                    ;AFTER SETTING "CLR" BIT #5
 8669                                                                    ;IN RHCS2 TO INIT THE RH
 8670                                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
 8671                                                                    ;ZEROS WERE MOVED INTO
 8672                                                                    ;RHDB
 8673                                                                    ;RHDB WAS READ THEN
 8674                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
 8675                                                                    ;WAS GIVEN THIS SHOULD
 8676                                                                    ;CLEAR ALL ERRORS
 8677                                                                    ;RHBA SHOULD HAVE 0
 8678                                                                    ;BUT CONTAINED WHAT IS
 8679                                                                    ;GIVEN IN BAD RHBA
 8680  025252                          83$:
 8681                                          ;CHECK THAT RHBAE HAS 0
 8682  025252  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 8683
 8684  025260  017737  156644  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
 8685  025266  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8686                                                                    ;DATA WITH DATA READ FROM
 8687                                                                    ;RHBAE
 8688  025274  001401                          BEQ     85$             ;BRANCH IF GOOD
 8689  025276  104112                          ERROR   112
 8690                                                                    ;AFTER SETTING "CLR" BIT #5
 8691                                                                    ;IN RHCS2 TO INIT THE RH
 8692                                                                    ;IPCK2 BIT #2 RHCS3 WAS SET
 8693                                                                    ;ZEROS WERE MOVED INTO
 8694                                                                    ;RHDB
 8695                                                                    ;RHDB WAS READ THEN
 8696                                                                    ;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
025300                                  85$:
                                                ;CHECK THAT RHWC HAS 0
025300  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0

025306  017737  156550  001126                MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
025314  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
025322  001401                              BEQ     87$             ;BRANCH IF GOOD
025324  104113                              ERROR   113
                                                        ;AFTER SETTING "CLR" BIT #5
                                                        ;IN RHCS2 TO INIT THE RH
                                                        ;IPCK2 BIT #2 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
025326                                  87$:


                                        ;;*****************************************************************
                                        ;*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)
                                        ;*      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

                                        ;;*****************************************************************
025326  000004                          T$T33:  SCOPE
025330  012706  001000                        MOV     #STACK,SP       ;RESET STACK
025334  012737  000033  004656                MOV     #33,@#T$TNM     ;SAVE TEST NUMBER

025342  004737  040306                        JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
025346  012737  000003  004662                MOV     #3,IP           ;IPCK BIT NO. FOR PRINTOUT

025354  052777  000010  156550                BIS     #IPCK3,@RHCS3 ;SET IPCK3 IN RHCS3-BIT #3
025362  005037  004210                        CLR     WRFROM          ;WRITE ZERO INTO LOCATION
```

```
 8753  025366  013777  004210  156506          MOV     WRFROM,@RHDB    ;WRITE ZEROS INTO RHDB ONCE
 8754  025374  013777  004210  156500          MOV     WRFROM,@RHDB    ;WRITE ZERO SECOND TIME
 8755
 8756                                          ;CHECK THAT RHCS1 HAS DVA!RDY
 8757  025402  012737  004200  001124          MOV     #DVA!RDY,$GDDAT ;GET GOOD = 4200
 8758
 8759  025410  017737  156444  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 8760  025416  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8761                                                                  ;DATA WITH DATA READ FROM
 8762                                                                  ;RHCS1
 8763  025424  001401                          BEQ     65$             ;BRANCH IF GOOD
 8764  025426  104140                          ERROR   140
 8765                                                                  ;AFTER SETTING "CLR" BIT #5
 8766                                                                  ;IN RHCS2 TO INIT THE RH
 8767                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
 8768                                                                  ;ZEROS WERE MOVED INTO RHDB
 8769                                                                  ;TWICE
 8770                                                                  ;RHCS1 SHOULD HAVE DVA!RDY
 8771                                                                  ;=4200
 8772                                                                  ;BUT CONTAINED WHAT IS
 8773                                                                  ;GIVEN IN BAD RHCS1
 8774  025430                          65$:
 8775  025430  017737  156446  001200          MOV     @RHDB,$TMP1     ;READ RHDB ONCE
 8776
 8777
 8778                                          ;CHECK THAT RHCS3 HAS 10
 8779  025436  012737  000010  001124          MOV     #10,$GDDAT      ;GET GOOD = 10
 8780
 8781  025444  017737  156462  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 8782  025452  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8783                                                                  ;DATA WITH DATA READ FROM
 8784                                                                  ;RHCS3
 8785  025460  001401                          BEQ     67$             ;BRANCH IF GOOD
 8786  025462  104104                          ERROR   104
 8787                                                                  ;AFTER SETTING "CLR" BIT #5
 8788                                                                  ;IN RHCS2 TO INIT THE RH
 8789                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
 8790                                                                  ;ZEROS WERE MOVED INTO
 8791                                                                  ;RHDB
 8792                                                                  ;RHDB WAS READ THEN
 8793                                                                  ;RHCS3 SHOULD HAVE 10
 8794                                                                  ;=10
 8795                                                                  ;BUT CONTAINED WHAT IS
 8796                                                                  ;GIVEN IN BAD RHCS3
 8797  025464                          67$:
 8798
 8799                                          ;CHECK THAT RHCS2 HAS MDPE!OR!IR
 8800  025464  012737  000700  001124          MOV     #MDPE!OR!IR,$GDDAT      ;GET GOOD = 700
 8801  025472  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 8802
 8803  025500  017737  156364  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 8804  025506  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 8805                                                                  ;DATA WITH DATA READ FROM
 8806                                                                  ;RHCS2
 8807  025514  001401                          BEQ     69$             ;BRANCH IF GOOD
 8808  025516  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
025520                          69$:

025520  017737  156356  001202          MOV     @RHDB,$TMP2     ;READ RHDB SECOND TIME
                                        ;CHECK THAT RHCS1 HAS SC!TRE!DVA!RDY
025526  012737  144200  001124          MOV     #SC!TRE!DVA!RDY,$GDDAT  ;GET GOOD = 144200

025534  017737  156320  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
025542  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
025550  001401                          BEQ     71$             ;BRANCH IF GOOD
025552  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
025554                          71$:
                                        ;CHECK THAT RHCS2 HAS MDPE!IR
025554  012737  000500  001124          MOV     #MDPE!IR,$GDDAT ;GET GOOD = 500
025562  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

025570  017737  156274  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
025576  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
025604  001401                          BEQ     73$             ;BRANCH IF GOOD
025606  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
025610                          73$:
```

```
8865
8866  025610  012737  000010  001124          MOV     #10,$GDDAT      ;GET GOOD = 10
8867
8868  025616  017737  156310  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8869  025624  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8870                                                                  ;DATA WITH DATA READ FROM
8871                                                                  ;RHCS3
8872  025632  001401                          BEQ     75$             ;BRANCH IF GOOD
8873  025634  104104                          ERROR   104
8874                                                                  ;AFTER SETTING "CLR" BIT #5
8875                                                                  ;IN RHCS2 TO INIT THE RH
8876                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8877                                                                  ;ZEROS WERE MOVED INTO
8878                                                                  ;RHDB
8879                                                                  ;RHDB WAS READ THEN
8880                                                                  ;RHCS3 SHOULD HAVE 10
8881                                                                  ;=10
8882                                                                  ;BUT CONTAINED WHAT IS
8883                                                                  ;GIVEN IN BAD RHCS3
8884  025636                          75$:
8885
8886  025636  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
8887                                                                  ;SETUP UNIT NUBER
8888                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
8889
8890
8891                                          ;CHECK THAT RHCS1 HAS RDY!DVA
8892  025642  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
8893
8894  025650  017737  156204  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
8895  025656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8896                                                                  ;DATA WITH DATA READ FROM
8897                                                                  ;RHCS1
8898  025664  001401                          BEQ     77$             ;BRANCH IF GOOD
8899  025666  104106                          ERROR   106
8900                                                                  ;AFTER SETTING "CLR" BIT #5
8901                                                                  ;IN RHCS2 TO INIT THE RH
8902                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8903                                                                  ;ZEROS WERE MOVED INTO
8904                                                                  ;RHDB
8905                                                                  ;RHDB WAS READ THEN
8906                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8907                                                                  ;WAS GIVEN THIS SHOULD
8908                                                                  ;CLEAR ALL ERRORS
8909                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
8910                                                                  ;=4200
8911                                                                  ;BUT CONTAINED WHAT IS
8912                                                                  ;GIVEN IN BAD RHCS1
8913  025670                          77$:
8914                                          ;CHECK THAT RHCS2 HAS IR
8915  025670  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
8916  025676  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
8917
8918  025704  017737  156160  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
8919  025712  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8920                                                                  ;DATA WITH DATA READ FROM
```

```
8921
8922  025720  001401                          BEQ     79$             ;RHCS2
8923  025722  104107                          ERROR   107             ;BRANCH IF GOOD
8924                                                                  ;AFTER SETTING "CLR" BIT #5
8925                                                                  ;IN RHCS2 TO INIT THE RH
8926                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8927                                                                  ;ZEROS WERE MOVED INTO
8928                                                                  ;RHDB
8929                                                                  ;RHDB WAS READ THEN
8930                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8931                                                                  ;WAS GIVEN THIS SHOULD
8932                                                                  ;CLEAR ALL ERRORS
8933                                                                  ;RHCS2 SHOULD HAVE IR
8934                                                                  ;=100
8935                                                                  ;TOGETHER WITH UNIT NUMBER
8936                                                                  ;BUT CONTAINED WHAT IS
8937                                                                  ;GIVEN IN BAD RHCS2
8938  025724                          79$:
8939                                          ;CHECK THAT RHCS3 HAS 0
8940  025724  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8941
8942  025732  017737  156174  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
8943  025740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8944                                                                  ;DATA WITH DATA READ FROM
8945                                                                  ;RHCS3
8946  025746  001401                          BEQ     81$             ;BRANCH IF GOOD
8947  025750  104110                          ERROR   110
8948                                                                  ;AFTER SETTING "CLR" BIT #5
8949                                                                  ;IN RHCS2 TO INIT THE RH
8950                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8951                                                                  ;ZEROS WERE MOVED INTO
8952                                                                  ;RHDB
8953                                                                  ;RHDB WAS READ THEN
8954                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
8955                                                                  ;WAS GIVEN THIS SHOULD
8956                                                                  ;CLEAR ALL ERRORS
8957                                                                  ;RHCS3 SHOULD HAVE 0
8958                                                                  ;BUT CONTAINED WHAT IS
8959                                                                  ;GIVEN IN BAD RHCS3
8960  025752                          81$:
8961                                          ;CHECK THAT RHBA HAS 0
8962  025752  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
8963
8964  025760  017737  156100  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
8965  025766  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8966                                                                  ;DATA WITH DATA READ FROM
8967                                                                  ;RHBA
8968  025774  001401                          BEQ     83$             ;BRANCH IF GOOD
8969  025776  104111                          ERROR   111
8970                                                                  ;AFTER SETTING "CLR" BIT #5
8971                                                                  ;IN RHCS2 TO INIT THE RH
8972                                                                  ;IPCK3 BIT #3 RHCS3 WAS SET
8973                                                                  ;ZEROS WERE MOVED INTO
8974                                                                  ;RHDB
8975                                                                  ;RHDB WAS READ THEN
8976                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
```

```
8977                                                                    ;WAS GIVEN THIS SHOULD
8978                                                                    ;CLEAR ALL ERRORS
8979                                                                    ;RHBA SHOULD HAVE 0
8980                                                                    ;BUT CONTAINED WHAT IS
8981                                                                    ;GIVEN IN BAD RHBA
8982  026000                                  83$:
8983                                                  ;CHECK THAT RHBAE HAS 0
8984  026000  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
8985
8986  026006  017737  156116  001126                  MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
8987  026014  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
8988                                                                    ;DATA WITH DATA READ FROM
8989                                                                    ;RHBAE
8990  026022  001401                                  BEQ     85$             ;BRANCH IF GOOD
8991  026024  104112                                  ERROR   112
8992                                                                    ;AFTER SETTING "CLR" BIT #5
8993                                                                    ;IN RHCS2 TO INIT THE RH
8994                                                                    ;IPCK3 BIT #3 RHCS3 WAS SET
8995                                                                    ;ZEROS WERE MOVED INTO
8996                                                                    ;RHDB
8997                                                                    ;RHDB WAS READ THEN
8998                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
8999                                                                    ;WAS GIVEN THIS SHOULD
9000                                                                    ;CLEAR ALL ERRORS
9001                                                                    ;RHBAE SHOULD HAVE 0
9002                                                                    ;BUT CONTAINED WHAT IS
9003                                                                    ;GIVEN IN BAD RHBAE
9004  026026                                  85$:
9005                                                  ;CHECK THAT RHWC HAS 0
9006  026026  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
9007
9008  026034  017737  156022  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
9009  026042  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9010                                                                    ;DATA WITH DATA READ FROM
9011                                                                    ;RHWC
9012  026050  001401                                  BEQ     87$             ;BRANCH IF GOOD
9013  026052  104113                                  ERROR   113
9014                                                                    ;AFTER SETTING "CLR" BIT #5
9015                                                                    ;IN RHCS2 TO INIT THE RH
9016                                                                    ;IPCK3 BIT #3 RHCS3 WAS SET
9017                                                                    ;ZEROS WERE MOVED INTO
9018                                                                    ;RHDB
9019                                                                    ;RHDB WAS READ THEN
9020                                                                    ;AN RH CLEAR (RHCS2 BIT #5)
9021                                                                    ;WAS GIVEN THIS SHOULD
9022                                                                    ;CLEAR ALL ERRORS
9023                                                                    ;RHWC SHOULD HAVE 0
9024                                                                    ;BUT CONTAINED WHAT IS
9025                                                                    ;GIVEN IN BAD RHWC
9026  026054                                  87$:
9027
9028
9029
9030
9031                                    ;;*****************************************************************
9032                                    ;*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

                                ;;*****************************************************************
026054  000004                  TST34:  SCOPE
026056  012706  001000                  MOV     #STACK,SP       ;RESET STACK
026062  012737  000034  004656          MOV     #34,@#TSTNM     ;SAVE TEST NUMBER

026070  004737  040306          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
026074  012737  177777  004210  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
026102  012777  177767  155752  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026110  012777  004210  155746  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

026116  004077  156516          JSR     R0,@COMND       ;GO TO DO COMMAND
026122  004154                  WRIDAT                  ;WRITE DATA

                                ;SET UP FOR WRITE CHECK
026124  005037  003002          CLR     BUFOW           ;WRITE DATA FOR WRITE CHECK ERROR
026130  012777  177767  155724  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026136  012777  003002  155720  MOV     #BUFOW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
                                                        ;FROM ODD WORD BOUNDRY

                                ;WRITE CHECK ON ODD WORD BOUNDARY
026144  004077  156470          JSR     R0,@COMND       ;GO TO DO COMMAND
026150  004150                  WRCHEK                  ;WRITE CHECK DATA


                                ;CHECK THAT RHCS2 HAS WCE!OR!IR
026152  012737  040300  001124  MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
026160  053737  005010  001124  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

026166  017737  155676  001126  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
026174  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
026202  001401                  BEQ     65$             ;BRANCH IF GOOD
026204  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
026206                          65$:

                                        ;CHECK THAT RHCS3 HAS DBL!WCEOW
026206  012737  012000  001124          MOV     #DBL!WCEOW,$GDDAT       ;GET GOOD = 12000

026214  017737  155712  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
026222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
026230  001401                          BEQ     67$             ;BRANCH IF GOOD
026232  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
026234                          67$:

026234  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                        ;SETUP UNIT NUBER
                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1


                                        ;CHECK THAT RHCS1 HAS RDY!DVA
026240  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200

026246  017737  155606  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
026254  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS1
026262  001401                          BEQ     69$             ;BRANCH IF GOOD
026264  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
026266                                  69$:
                                        ;CHECK THAT RHCS2 HAS IR
026266  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
026274  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

026302  017737  155562  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
026310  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS2
026316  001401                          BEQ     71$             ;BRANCH IF GOOD
026320  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
026322                                  71$:
                                        ;CHECK THAT RHCS3 HAS 0
026322  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

026330  017737  155576  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
026336  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
026344  001401                          BEQ     73$             ;BRANCH IF GOOD
026346  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
026350                                  73$:
                                        ;CHECK THAT RHBA HAS 0
026350  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

026356  017737  155502  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
026364  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
```

```
CERHAD0 MACY11 30(1046)  21-DEC-77  13:06  PAGE 175
CERHAD.P11    21-DEC-77 12:48          T34     RHCS2-WCE BIT #14 AND RHCS3-WCE OW BIT #12                    SEQ 0174

 9201
 9202  026372  001401                          BEQ     75$                ;RHBA
 9203  026374  104121                          ERROR   121                ;BRANCH IF GOOD
 9204                                                                     ;AFTER SETTING "CLR" BIT #5
 9205                                                                     ;IN RHCS2 TO INIT THE RH
 9206                                                                     ;ONE WORD OF ALL ONES IS
 9207                                                                     ;WRITTEN ON THE DEVICE
 9208                                                                     ;A WRITE CHECK WAS DONE
 9209                                                                     ;ON AN ODD WORD BOUNDARY THEN
 9210                                                                     ;AN RH CLEAR (RHCS2 BIT #5)
 9211                                                                     ;WAS GIVEN THIS SHOULD
 9212                                                                     ;CLEAR ALL ERRORS
 9213                                                                     ;RHBA SHOULD HAVE 0
 9214                                                                     ;BUT CONTAINED WHAT IS
 9215                                                                     ;GIVEN IN BAD RHBA
 9216  026376                          75$:
 9217                                          ;CHECK THAT RHBAE HAS 0
 9218  026376  012737  000000  001124          MOV     #0,$GDDAT          ;GET GOOD = 0
 9219
 9220  026404  017737  155520  001126          MOV     @RHBAE,$BDDAT      ;READ RHBAE FOR COMPARISON
 9221  026412  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
 9222                                                                     ;DATA WITH DATA READ FROM
 9223                                                                     ;RHBAE
 9224  026420  001401                          BEQ     77$                ;BRANCH IF GOOD
 9225  026422  104122                          ERROR   122
 9226                                                                     ;AFTER SETTING "CLR" BIT #5
 9227                                                                     ;IN RHCS2 TO INIT THE RH
 9228                                                                     ;ONE WORD OF ALL ONES IS
 9229                                                                     ;WRITTEN ON THE DEVICE
 9230                                                                     ;A WRITE CHECK WAS DONE
 9231                                                                     ;ON AN ODD WORD BOUNDARY THEN
 9232                                                                     ;AN RH CLEAR (RHCS2 BIT #5)
 9233                                                                     ;WAS GIVEN THIS SHOULD
 9234                                                                     ;CLEAR ALL ERRORS
 9235                                                                     ;RHBAE SHOULD HAVE 0
 9236                                                                     ;BUT CONTAINED WHAT IS
 9237                                                                     ;GIVEN IN BAD RHBAE
 9238  026424                          77$:
 9239                                          ;CHECK THAT RHWC HAS 0
 9240  026424  012737  000000  001124          MOV     #0,$GDDAT          ;GET GOOD = 0
 9241
 9242  026432  017737  155424  001126          MOV     @RHWC,$BDDAT       ;READ RHWC FOR COMPARISON
 9243  026440  023737  001124  001126          CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
 9244                                                                     ;DATA WITH DATA READ FROM
 9245                                                                     ;RHWC
 9246  026446  001401                          BEQ     78$                ;BRANCH IF GOOD
 9247  026450  104123                          ERROR   123
 9248
 9249                                                                     ;AFTER SETTING "CLR" BIT #5
 9250                                                                     ;IN RHCS2 TO INIT THE RH
 9251                                                                     ;ONE WORD OF ALL ONES IS
 9252                                                                     ;WRITTEN ON THE DEVICE
 9253                                                                     ;A WRITE CHECK WAS DONE
 9254                                                                     ;ON AN ODD WORD BOUNDARY THEN
 9255                                                                     ;AN RH CLEAR (RHCS2 BIT #5)
 9256                                                                     ;WAS GIVEN THIS SHOULD
                                                                          ;CLEAR ALL ERRORS
```

```
                                                        ;RHWC SHOULD HAVE 0
                                                        ;BUT CONTAINED WHAT IS
                                                        ;GIVEN IN BAD RHWC
026452                                  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

                                        ;;*****************************************************************
026452  000004                          T$T35:  SCOPE
026454  012706  001000                          MOV     #STACK,SP       ;RESET STACK
026460  012737  000035  004656                  MOV     #35,@#T$TNM     ;SAVE TEST NUMBER

026466  004737  040306                          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
026472  012737  177777  004210                  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
026500  012777  177767  155354                  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026506  012777  004210  155350                  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

026514  004077  156120                          JSR     R0,@COMND       ;GO TO DO COMMAND
026520  004154                                  WRIDAT                  ;WRITE DATA

                                        ;SET UP FOR WRITE CHECK
026522  005037  003002                          CLR     BUFOW           ;WRITE DATA FOR WRITE CHECK ERROR
026526  012777  177767  155326                  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
026534  012777  003002  155322                  MOV     #BUFOW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
                                                                        ;FROM ODD WORD BOUNDRY
                                        ;WRITE CHECK ON ODD WORD BOUNDARY
026542  004077  156072                  JSR     R0,@COMND               ;GO TO DO COMMAND
026546  004150                          WRCHEK                          ;WRITE CHECK DATA


                                        ;CHECK THAT RHCS2 HAS WCE!OR!IR
026550  012737  040300  001124          MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
026556  053737  005010  001124          BIS     UNIT,$GDDAT             ;INCLUDE UNIT NUMBER

026564  017737  155300  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
```

```
 9313  026572  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9314                                                                    ;DATA WITH DATA READ FROM
 9315                                                                    ;RHCS2
 9316  026600  001401                          BEQ     65$             ;BRANCH IF GOOD
 9317  026602  104114                          ERROR   114
 9318                                                                    ;AFTER SETTING "CLR" BIT #5
 9319                                                                    ;IN RHCS2 TO INIT THE RH
 9320                                                                    ;ONE WORD OF ALL ONES IS
 9321                                                                    ;WRITTEN ON THE DEVICE
 9322                                                                    ;A WRITE CHECK WAS DONE
 9323                                                                    ;ON AN ODD WORD BOUNDARY THEN
 9324                                                                    ;RHCS2 SHOULD HAVE WCE!OR!IR
 9325                                                                    ;=40300
 9326                                                                    ;TOGETHER WITH UNIT NUMBER
 9327                                                                    ;BUT CONTAINED WHAT IS
 9328                                                                    ;GIVEN IN BAD RHCS2
 9329  026604                          65$:
 9330
 9331                                          ;CHECK THAT RHCS3 HAS DBL!WCEOW
 9332  026604  012737  012000  001124          MOV     #DBL!WCEOW,$GDDAT       ;GET GOOD = 12000
 9333
 9334  026612  017737  155314  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 9335  026620  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9336                                                                    ;DATA WITH DATA READ FROM
 9337                                                                    ;RHCS3
 9338  026626  001401                          BEQ     67$             ;BRANCH IF GOOD
 9339  026630  104115                          ERROR   115
 9340                                                                    ;AFTER SETTING "CLR" BIT #5
 9341                                                                    ;IN RHCS2 TO INIT THE RH
 9342                                                                    ;ONE WORD OF ALL ONES IS
 9343                                                                    ;WRITTEN ON THE DEVICE
 9344                                                                    ;A WRITE CHECK WAS DONE
 9345                                                                    ;ON AN ODD WORD BOUNDARY THEN
 9346                                                                    ;RHCS3 SHOULD HAVE DBL!WCEOW
 9347                                                                    ;=12000
 9348                                                                    ;BUT CONTAINED WHAT IS
 9349                                                                    ;GIVEN IN BAD RHCS3
 9350  026632                          67$:
 9351
 9352
 9353  026632  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
 9354                                                                    ;SETUP UNIT NUBER
 9355                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
 9356
 9357
 9358                                          ;CHECK THAT RHCS1 HAS RDY!DVA
 9359  026636  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
 9360
 9361  026644  017737  155210  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 9362  026652  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9363                                                                    ;DATA WITH DATA READ FROM
 9364                                                                    ;RHCS1
 9365  026660  001401                          BEQ     69$             ;BRANCH IF GOOD
 9366  026662  104116                          ERROR   116
 9367                                                                    ;AFTER SETTING "CLR" BIT #5
 9368                                                                    ;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
026664                                  69$:
                                        ;CHECK THAT RHCS2 HAS IR
026664  012737  000100  001124          MOV     #IR,$GDDAT              ;GET GOOD = 100
026672  053737  005010  001124          BIS     UNIT,$GDDAT             ;INCLUDE UNIT NUMBER

026700  017737  155164  001126          MOV     @RHCS2,$BDDAT           ;READ RHCS2 FOR COMPARISON
026706  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS2
026714  001401                          BEQ     71$                     ;BRANCH IF GOOD
026716  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
026720                                  71$:
                                        ;CHECK THAT RHCS3 HAS 0
026720  012737  000000  001124          MOV     #0,$GDDAT               ;GET GOOD = 0

026726  017737  155200  001126          MOV     @RHCS3,$BDDAT           ;READ RHCS3 FOR COMPARISON
026734  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS3
026742  001401                          BEQ     73$                     ;BRANCH IF GOOD
026744  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
026746                                  73$:
                                        ;CHECK THAT RHBA HAS 0
026746  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

026754  017737  155104  001126          MOV     @RH6A,$BDDAT    ;READ RHBA FOR COMPARISON
026762  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBA
026770  001401                          BEQ     75$             ;BRANCH IF GOOD
026772  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
026774                                  75$:
                                        ;CHECK THAT RHBAE HAS 0
026774  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027002  017737  155122  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
027010  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBAE
027016  001401                          BEQ     77$             ;BRANCH IF GOOD
027020  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
027022                                  77$:
                                        ;CHECK THAT RHWC HAS 0
027022  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027030  017737  155026  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
027036  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
027044  001401                          BEQ     79$             ;BRANCH IF GOOD
027046  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
027050                          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

                                ;;*****************************************************************
027050  000004                  T36:    SCOPE
027052  012706  001000                  MOV     #STACK,SP       ;RESET STACK
027056  012737  000036  004656          MOV     #36,@#TSTNM     ;SAVE TEST NUMBER

027064  004737  040306          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
027070  012737  177777  004210  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
027076  012777  177767  154756  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
027104  012777  004210  154752  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

027112  004077  155522          JSR     R0,@COMND       ;GO TO DO COMMAND
027116  004154                  WRIDAT                  ;WRITE DATA

                                ;SET UP FOR WRITE CHECK
027120  005037  003000          CLR     BUFEW           ;WRITE DATA FOR WRITE CHECK ERROR
027124  012777  177767  154730  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
027132  012777  003000  154724  MOV     #BUFEW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
                                                        ;FROM EVEN WORD BOUNDRY

                                ;WRITE CHECK ON EVEN WORD BOUNDARY
027140  004077  155474          JSR     R0,@COMND       ;GO TO DO COMMAND
027144  004150                  WRCHEK                  ;WRITE CHECK DATA
```

```
                                        ;CHECK THAT RHCS2 HAS WCE!OR!IR
027146  012737  040300  001124        MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
027154  053737  005010  001124        BIS     UNIT,$GDDAT             ;INCLUDE UNIT NUMBER

027162  017737  154702  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
027170  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
027176  001401                        BEQ     65$             ;BRANCH IF GOOD
027200  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
027202                          65$:

                                        ;CHECK THAT RHCS3 HAS DBL!WCEEW
027202  012737  006000  001124        MOV     #DBL!WCEEW,$GDDAT       ;GET GOOD = 6000

027210  017737  154716  001126        MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
027216  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS3
027224  001401                        BEQ     67$             ;BRANCH IF GOOD
027226  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
027230                          67$:

027230  004737  040306                JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                ;SETUP UNIT NUBER
                                                                ;CLEAR RHWC AND FUNCTION BITS IN RHCS1

                                        ;CHECK THAT RHCS1 HAS RDY!DVA
027234  012737  004200  001124        MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
```

```
 9593
 9594  027242  017737  154612  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
 9595  027250  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9596                                                                  ;DATA WITH DATA READ FROM
 9597                                                                  ;RHCS1
 9598  027256  001401                          BEQ     69$             ;BRANCH IF GOOD
 9599  027260  104116                          ERROR   116
 9600                                                                  ;AFTER SETTING "CLR" BIT #5
 9601                                                                  ;IN RHCS2 TO INIT THE RH
 9602                                                                  ;ONE WORD OF ALL ONES IS
 9603                                                                  ;WRITTEN ON THE DEVICE
 9604                                                                  ;A WRITE CHECK WAS DONE
 9605                                                                  ;ON AN EVEN WORD BOUNDARY THEN
 9606                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 9607                                                                  ;WAS GIVEN THIS SHOULD
 9608                                                                  ;CLEAR ALL ERRORS
 9609                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
 9610                                                                  ;=4200
 9611                                                                  ;BUT CONTAINED WHAT IS
 9612                                                                  ;GIVEN IN BAD RHCS1
 9613  027262                          69$:
 9614                                          ;CHECK THAT RHCS2 HAS IR
 9615  027262  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
 9616  027270  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 9617
 9618  027276  017737  154566  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 9619  027304  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9620                                                                  ;DATA WITH DATA READ FROM
 9621                                                                  ;RHCS2
 9622  027312  001401                          BEQ     71$             ;BRANCH IF GOOD
 9623  027314  104117                          ERROR   117
 9624                                                                  ;AFTER SETTING "CLR" BIT #5
 9625                                                                  ;IN RHCS2 TO INIT THE RH
 9626                                                                  ;ONE WORD OF ALL ONES IS
 9627                                                                  ;WRITTEN ON THE DEVICE
 9628                                                                  ;A WRITE CHECK WAS DONE
 9629                                                                  ;ON AN EVEN WORD BOUNDARY THEN
 9630                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 9631                                                                  ;WAS GIVEN THIS SHOULD
 9632                                                                  ;CLEAR ALL ERRORS
 9633                                                                  ;RHCS2 SHOULD HAVE IR
 9634                                                                  ;=100
 9635                                                                  ;TOGETHER WITH UNIT NUMBER
 9636                                                                  ;BUT CONTAINED WHAT IS
 9637                                                                  ;GIVEN IN BAD RHCS2
 9638  027316                          71$:
 9639                                          ;CHECK THAT RHCS3 HAS 0
 9640  027316  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9641
 9642  027324  017737  154602  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 9643  027332  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9644                                                                  ;DATA WITH DATA READ FROM
 9645                                                                  ;RHCS3
 9646  027340  001401                          BEQ     73$             ;BRANCH IF GOOD
 9647  027342  104120                          ERROR   120
 9648                                                                  ;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
027344                                  73$:
                                                ;CHECK THAT RHBA HAS 0
027344  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027352  017737  154506  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
027360  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBA
027366  001401                          BEQ     75$             ;BRANCH IF GOOD
027370  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
027372                                  75$:
                                                ;CHECK THAT RHBAE HAS 0
027372  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027400  017737  154524  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
027406  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHBAE
027414  001401                          BEQ     77$             ;BRANCH IF GOOD
027416  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
027420                                  77$:
```

```
                                        ;CHECK THAT RHWC HAS 0
027420  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

027426  017737  154430  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
027434  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
027442  001401                          BEQ     79$             ;BRANCH IF GOOD
027444  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
027446                          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

;;*****************************************************************
027446  000004          ↑ST37:  SCOPE
027450  012706  001000          MOV     #STACK,SP       ;RESET STACK
027454  012737  000037  004656  MOV     #37,@#↑STNM     ;SAVE TEST NUMBER

027462  004737  040306          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
027466  012737  177777  004210  MOV     #-1,WRFROM      ;ALL ONES INTO WRITE FROM
027474  012777  177767  154360  MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
027502  012777  004210  154354  MOV     #WRFROM,@RHBA   ;WRITE FROM BUFFER INTO BUS ADDRESS

027510  004077  155124          JSR     R0,@COMND       ;GO TO DO COMMAND
027514  004154                  WRIDAT                  ;WRITE DATA
```

```
 9761
 9762
 9763  027516  005037  003000                    ;SET UP FOR WRITE CHECK
                                                  CLR     BUFEW           ;WRITE DATA FOR WRITE CHECK ERROR
 9764  027522  012777  177767  154332            MOV     #-9.,@RHWC      ;NINE WORD FOR WORD COUNT REGISTER
 9765  027530  012777  003000  154326            MOV     #BUFEW,@RHBA    ;WRITE FROM BUFFER INTO BUS ADDRESS
 9766                                            ;FROM EVEN WORD BOUNDRY
 9767                                            ;WRITE CHECK ON EVEN WORD BOUNDARY
 9768  027536  004077  155076                    JSR     R0,@COMND       ;GO TO DO COMMAND
 9769  027542  004150                            WRCHEK                  ;WRITE CHECK DATA
 9770
 9771
 9772
 9773
 9774                                            ;CHECK THAT RHCS2 HAS WCE!OR!IR
 9775  027544  012737  040300  001124            MOV     #WCE!OR!IR,$GDDAT       ;GET GOOD = 40300
 9776  027552  053737  005010  001124            BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
 9777
 9778  027560  017737  154304  001126            MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
 9779  027566  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                          ;DATA WITH DATA READ FROM
                                                                          ;RHCS2
 9782  027574  001401                            BEQ     65$             ;BRANCH IF GOOD
 9783  027576  104114                            ERROR   114
 9784                                                                     ;AFTER SETTING "CLR" BIT #5
 9785                                                                     ;IN RHCS2 TO INIT THE RH
 9786                                                                     ;ONE WORD OF ALL ONES IS
 9787                                                                     ;WRITTEN ON THE DEVICE
 9788                                                                     ;A WRITE CHECK WAS DONE
 9789                                                                     ;ON AN EVEN WORD BOUNDARY THEN
 9790                                                                     ;RHCS2 SHOULD HAVE WCE!OR!IR
 9791                                                                     ;=40300
 9792                                                                     ;TOGETHER WITH UNIT NUMBER
 9793                                                                     ;BUT CONTAINED WHAT IS
 9794                                                                     ;GIVEN IN BAD RHCS2
 9795  027600                                65$:
 9796
 9797                                            ;CHECK THAT RHCS3 HAS DBL!WCEEW
 9798  027600  012737  006000  001124            MOV     #DBL!WCEEW,$GDDAT       ;GET GOOD = 6000
 9799
 9800  027606  017737  154320  001126            MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 9801  027614  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9802                                                                     ;DATA WITH DATA READ FROM
 9803                                                                     ;RHCS3
 9804  027622  001401                            BEQ     67$             ;BRANCH IF GOOD
 9805  027624  104115                            ERROR   115
 9806                                                                     ;AFTER SETTING "CLR" BIT #5
 9807                                                                     ;IN RHCS2 TO INIT THE RH
 9808                                                                     ;ONE WORD OF ALL ONES IS
 9809                                                                     ;WRITTEN ON THE DEVICE
 9810                                                                     ;A WRITE CHECK WAS DONE
 9811                                                                     ;ON AN EVEN WORD BOUNDARY THEN
 9812                                                                     ;RHCS3 SHOULD HAVE DBL!WCEEW
 9813                                                                     ;=6000
 9814                                                                     ;BUT CONTAINED WHAT IS
 9815                                                                     ;GIVEN IN BAD RHCS3
 9816  027626                                67$:
```

```
9817
9818
9819  027626  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
9820                                                                  ;SETUP UNIT NUBER
9821                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
9822
9823
9824                                          ;CHECK THAT RHCS1 HAS RDY!DVA
9825  027632  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
9826
9827  027640  017737  154214  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
9828  027646  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9829                                                                  ;DATA WITH DATA READ FROM
9830                                                                  ;RHCS1
9831  027654  001401                          BEQ     69$             ;BRANCH IF GOOD
9832  027656  104116                          ERROR   116
9833                                                                  ;AFTER SETTING "CLR" BIT #5
9834                                                                  ;IN RHCS2 TO INIT THE RH
9835                                                                  ;ONE WORD OF ALL ONES IS
9836                                                                  ;WRITTEN ON THE DEVICE
9837                                                                  ;A WRITE CHECK WAS DONE
9838                                                                  ;ON AN EVEN WORD BOUNDARY THEN
9839                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
9840                                                                  ;WAS GIVEN THIS SHOULD
9841                                                                  ;CLEAR ALL ERRORS
9842                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
9843                                                                  ;=4200
9844                                                                  ;BUT CONTAINED WHAT IS
9845                                                                  ;GIVEN IN BAD RHCS1
9846  027660                          69$:
9847                                          ;CHECK THAT RHCS2 HAS IR
9848  027660  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
9849  027666  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
9850
9851  027674  017737  154170  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
9852  027702  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
9853                                                                  ;DATA WITH DATA READ FROM
9854                                                                  ;RHCS2
9855  027710  001401                          BEQ     71$             ;BRANCH IF GOOD
9856  027712  104117                          ERROR   117
9857                                                                  ;AFTER SETTING "CLR" BIT #5
9858                                                                  ;IN RHCS2 TO INIT THE RH
9859                                                                  ;ONE WORD OF ALL ONES IS
9860                                                                  ;WRITTEN ON THE DEVICE
9861                                                                  ;A WRITE CHECK WAS DONE
9862                                                                  ;ON AN EVEN WORD BOUNDARY THEN
9863                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
9864                                                                  ;WAS GIVEN THIS SHOULD
9865                                                                  ;CLEAR ALL ERRORS
9866                                                                  ;RHCS2 SHOULD HAVE IR
9867                                                                  ;=100
9868                                                                  ;TOGETHER WITH UNIT NUMBER
9869                                                                  ;BUT CONTAINED WHAT IS
9870                                                                  ;GIVEN IN BAD RHCS2
9871  027714                          71$:
9872                                          ;CHECK THAT RHCS3 HAS 0
```

```
 9873  027714  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9874
 9875  027722  017737  154204  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
 9876  027730  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9877                                                                  ;DATA WITH DATA READ FROM
 9878                                                                  ;RHCS3
 9879  027736  001401                          BEQ     73$             ;BRANCH IF GOOD
 9880  027740  104120                          ERROR   120
 9881                                                                  ;AFTER SETTING "CLR" BIT #5
 9882                                                                  ;IN RHCS2 TO INIT THE RH
 9883                                                                  ;ONE WORD OF ALL ONES IS
 9884                                                                  ;WRITTEN ON THE DEVICE
 9885                                                                  ;A WRITE CHECK WAS DONE
 9886                                                                  ;ON AN EVEN WORD BOUNDARY THEN
 9887                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 9888                                                                  ;WAS GIVEN THIS SHOULD
 9889                                                                  ;CLEAR ALL ERRORS
 9890                                                                  ;RHCS3 SHOULD HAVE 0
 9891                                                                  ;BUT CONTAINED WHAT IS
 9892                                                                  ;GIVEN IN BAD RHCS3
 9893  027742                          73$:
 9894                                          ;CHECK THAT RHBA HAS 0
 9895  027742  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9896
 9897  027750  017737  154110  001126          MOV     @RHBA,$BDDAT    ;READ RHBA FOR COMPARISON
 9898  027756  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9899                                                                  ;DATA WITH DATA READ FROM
 9900                                                                  ;RHBA
 9901  027764  001401                          BEQ     75$             ;BRANCH IF GOOD
 9902  027766  104121                          ERROR   121
 9903                                                                  ;AFTER SETTING "CLR" BIT #5
 9904                                                                  ;IN RHCS2 TO INIT THE RH
 9905                                                                  ;ONE WORD OF ALL ONES IS
 9906                                                                  ;WRITTEN ON THE DEVICE
 9907                                                                  ;A WRITE CHECK WAS DONE
 9908                                                                  ;ON AN EVEN WORD BOUNDARY THEN
 9909                                                                  ;AN RH CLEAR (RHCS2 BIT #5)
 9910                                                                  ;WAS GIVEN THIS SHOULD
 9911                                                                  ;CLEAR ALL ERRORS
 9912                                                                  ;RHBA SHOULD HAVE 0
 9913                                                                  ;BUT CONTAINED WHAT IS
 9914                                                                  ;GIVEN IN BAD RHBA
 9915  027770                          75$:
 9916                                          ;CHECK THAT RHBAE HAS 0
 9917  027770  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
 9918
 9919  027776  017737  154126  001126          MOV     @RHBAE,$BDDAT   ;READ RHBAE FOR COMPARISON
 9920  030004  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 9921                                                                  ;DATA WITH DATA READ FROM
 9922                                                                  ;RHBAE
 9923  030012  001401                          BEQ     77$             ;BRANCH IF GOOD
 9924  030014  104122                          ERROR   122
 9925                                                                  ;AFTER SETTING "CLR" BIT #5
 9926                                                                  ;IN RHCS2 TO INIT THE RH
 9927                                                                  ;ONE WORD OF ALL ONES IS
 9928                                                                  ;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
030016                                  77$:
                                        ;CHECK THAT RHWC HAS 0
030016  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0

030024  017737  154032  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
030032  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHWC
030040  001401                          BEQ     79$             ;BRANCH IF GOOD
030042  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
030044                                  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

                                        ;;*****************************************************************
030044  000004                          .TST40: SCOPE
030046  012706  001000                  MOV     #STACK,SP       ;RESET STACK
030052  012737  000040  004656          MOV     #40,@#.TSTNM    ;SAVE TEST NUMBER

030060  004737  040306                  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
030064  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
```

```
 9985  030070  012702  003000                MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
 9986                                                                  ;FROM AN EVEN WORD BOUNDARY
 9987  030074  012722  052525          1$:   MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
 9988  030100  005301                      DEC     R1              ;COUNT
 9989  030102  001374                      BNE     1$              ;BRANCH IF THREE NOT DONE
 9990  030104  012777  003000  153752      MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
 9991                                                                ;FROM AN EVEN WORD BOUNDARY
 9992  030112  012777  177767  153742      MOV #-9.,@RHWC ;WORD COUNT NINE
 9993  030120  004077  154514              JSR     R0,@COMND       ;GO TO DO COMMAND
 9994  030124  004154                      WRIDAT                  ;WRITE DATA
 9995
 9996
 9997                                      ;CHECK THAT RHCS3 HAS 0
 9998  030126  012737  000000  001124      MOV     #0,$GDDAT       ;GET GOOD = 0
 9999
10000  030134  017737  153772  001126      MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10001  030142  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10002                                                               ;DATA WITH DATA READ FROM
10003                                                               ;RHCS3
10004  030150  001401                      BEQ     65$             ;BRANCH IF GOOD
10005  030152  104124                      ERROR   124
10006                                                               ;AFTER SETTING "CLR" BIT #5
10007                                                               ;IN RHCS2 TO INIT THE RH
10008                                                               ;A NINE WORD WRITE FROM AN
10009                                                               ;EVEN WORD BOUNDARY WAS DONE
10010                                                               ;THEN
10011                                                               ;RHCS3 SHOULD HAVE 0
10012                                                               ;BUT CONTAINED WHAT IS
10013                                                               ;GIVEN IN BAD RHCS3
10014  030154                         65$:
10015  030154  042777  000076  153676      BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10016                                      ;CHECK THAT RHCS1 HAS RDY!DVA
10017  030162  012737  004200  001124      MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10018
10019  030170  017737  153664  001126      MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10020  030176  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10021                                                               ;DATA WITH DATA READ FROM
10022                                                               ;RHCS1
10023  030204  001401                      BEQ     67$             ;BRANCH IF GOOD
10024  030206  104125                      ERROR   125
10025                                                               ;AFTER SETTING "CLR" BIT #5
10026                                                               ;IN RHCS2 TO INIT THE RH
10027                                                               ;A NINE WORD WRITE FROM AN
10028                                                               ;EVEN WORD BOUNDARY WAS DONE
10029                                                               ;THEN
10030                                                               ;RHCS1 SHOULD HAVE RDY!DVA
10031                                                               ;=4200
10032                                                               ;BUT CONTAINED WHAT IS
10033                                                               ;GIVEN IN BAD RHCS1
10034  030210                         67$:
10035                                      ;CHECK THAT RHCS2 HAS IR
10036  030210  012737  000100  001124      MOV     #IR,$GDDAT      ;GET GOOD = 100
10037  030216  053737  005010  001124      BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10038
10039  030224  017737  153640  001126      MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10040  030232  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
10041
10042
10043  030240  001401                          BEQ     69$             ;BRANCH IF GOOD
10044  030242  104126                          ERROR   126
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                                                                   ;GIVEN IN BAD RHCS2
10055  030244                           69$:
10056                                          ;CHECK THAT RHWC HAS 0
10057  030244  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10058
10059  030252  017737  153604  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10060  030260  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10061                                                                   ;DATA WITH DATA READ FROM
10062                                                                   ;RHWC
10063  030266  001401                          BEQ     71$             ;BRANCH IF GOOD
10064  030270  104131                          ERROR   131
10065                                                                   ;AFTER SETTING "CLR" BIT #5
10066                                                                   ;IN RHCS2 TO INIT THE RH
10067                                                                   ;A NINE WORD WRITE FROM AN
10068                                                                   ;EVEN WORD BOUNDARY WAS DONE
10069                                                                   ;THEN
10070                                                                   ;RHWC SHOULD HAVE 0
10071                                                                   ;BUT CONTAINED WHAT IS
10072                                                                   ;GIVEN IN BAD RHWC
10073  030272                           71$:
10074
10075
10076
10077                                    ;;*****************************************************************
10078                                    ;*TEST 41        TEST DBL (RHCS3 BIT #10) TEST B
10079
10080                                    ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10081                                    ;*      SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10082                                    ;*      DO A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10083                                    ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
10084                                    ;*      CHECK RHCS3
10085                                    ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10086
10087                                    ;;*****************************************************************
10088  030272  000004                   TST41:  SCOPE
10089  030274  012706  001000                   MOV     #STACK,SP       ;RESET STACK
10090  030300  012737  000041  004656           MOV     #41,@#TSTNM     ;SAVE TEST NUMBER
10091
10092  030306  004737  040306                   JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10093                                                                   ;SETUP UNIT NUBER
10094                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10095
10096                                          ;SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
```

```
10097  030312  012701  000003                  MOV     #3,R1                   ;COUNT OF THREE
10098  030316  012702  003000                  MOV     #BFEVEN,R2              ;START THREE WORD BUFFER
10099                                                                          ;FROM AN EVEN WORD BOUNDARY
10100  030322  012722  052525          1$:     MOV     #52525,(R2)+            ;MOVE THREE 52525 INTO BUFFER
10101  030326  005301                          DEC     R1                      ;COUNT
10102  030330  001374                          BNE     1$                      ;BRANCH IF THREE NOT DONE
10103  030332  012777  003000  153524          MOV     #BFEVEN,@RHBA           ;SET BUS ADDRESS TO START
10104                                                                          ;FROM AN EVEN WORD BOUNDARY
10105  030340  012777  177766  153514          MOV #-10.,@RHWC ;WORD COUNT TEN
10106  030346  004077  154266                  JSR     R0,@COMND               ;GO TO DO COMMAND
10107  030352  004154                          WRIDAT                          ;WRITE DATA
10108
10109
10110                                          ;CHECK THAT RHCS3 HAS DBL
10111  030354  012737  002000  001124          MOV     #DBL,$GDDAT             ;GET GOOD = 2000
10112
10113  030362  017737  153544  001126          MOV     @RHCS3,$BDDAT           ;READ RHCS3 FOR COMPARISON
10114  030370  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
10115                                                                          ;DATA WITH DATA READ FROM
10116                                                                          ;RHCS3
10117  030376  001401                          BEQ     65$                     ;BRANCH IF GOOD
10118  030400  104124                          ERROR   124
10119                                                                          ;AFTER SETTING "CLR" BIT #5
10120                                                                          ;IN RHCS2 TO INIT THE RH
10121                                                                          ;A TEN WORD WRITE FROM AN
10122                                                                          ;EVEN WORD BOUNDARY WAS DONE
10123                                                                          ;THEN
10124                                                                          ;RHCS3 SHOULD HAVE DBL
10125                                                                          ;=2000
10126                                                                          ;BUT CONTAINED WHAT IS
10127                                                                          ;GIVEN IN BAD RHCS3
10128  030402                          65$:
10129  030402  042777  000076  153450          BIC     #76,@RHCS1              ;CLEAR FUNCTION BITS
10130                                          ;CHECK THAT RHCS1 HAS RDY!DVA
10131  030410  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10132
10133  030416  017737  153436  001126          MOV     @RHCS1,$BDDAT           ;READ RHCS1 FOR COMPARISON
10134  030424  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPECTED
10135                                                                          ;DATA WITH DATA READ FROM
10136                                                                          ;RHCS1
10137  030432  001401                          BEQ     67$                     ;BRANCH IF GOOD
10138  030434  104125                          ERROR   125
10139                                                                          ;AFTER SETTING "CLR" BIT #5
10140                                                                          ;IN RHCS2 TO INIT THE RH
10141                                                                          ;A TEN WORD WRITE FROM AN
10142                                                                          ;EVEN WORD BOUNDARY WAS DONE
10143                                                                          ;THEN
10144                                                                          ;RHCS1 SHOULD HAVE RDY!DVA
10145                                                                          ;=4200
10146                                                                          ;BUT CONTAINED WHAT IS
10147                                                                          ;GIVEN IN BAD RHCS1
10148  030436                          67$:
10149                                          ;CHECK THAT RHCS2 HAS IR
10150  030436  012737  000100  001124          MOV     #IR,$GDDAT              ;GET GOOD = 100
10151  030444  053737  005010  001124          BIS     UNIT,$GDDAT             ;INCLUDE UNIT NUMBER
10152
```

```
10153  030452  017737  153412  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10154  030460  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10155                                                                  ;DATA WITH DATA READ FROM
10156                                                                  ;RHCS2
10157  030466  001401                          BEQ     69$             ;BRANCH IF GOOD
10158  030470  104126                          ERROR   126
10159                                                                  ;AFTER SETTING "CLR" BIT #5
10160                                                                  ;IN RHCS2 TO INIT THE RH
10161                                                                  ;A TEN WORD WRITE FROM AN
10162                                                                  ;EVEN WORD BOUNDARY WAS DONE
10163                                                                  ;THEN
10164                                                                  ;RHCS2 SHOULD HAVE IR
10165                                                                  ;=100
10166                                                                  ;TOGETHER WITH UNIT NUMBER
10167                                                                  ;BUT CONTAINED WHAT IS
10168                                                                  ;GIVEN IN BAD RHCS2
10169  030472                          69$:
10170                                          ;CHECK THAT RHWC HAS 0
10171  030472  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10172
10173  030500  017737  153356  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10174  030506  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10175                                                                  ;DATA WITH DATA READ FROM
10176                                                                  ;RHWC
10177  030514  001401                          BEQ     71$             ;BRANCH IF GOOD
10178  030516  104131                          ERROR   131
10179                                                                  ;AFTER SETTING "CLR" BIT #5
10180                                                                  ;IN RHCS2 TO INIT THE RH
10181                                                                  ;A TEN WORD WRITE FROM AN
10182                                                                  ;EVEN WORD BOUNDARY WAS DONE
10183                                                                  ;THEN
10184                                                                  ;RHWC SHOULD HAVE 0
10185                                                                  ;BUT CONTAINED WHAT IS
10186                                                                  ;GIVEN IN BAD RHWC
10187  030520                          71$:
10188
10189
10190
10191                                  ;;*****************************************************************
10192                                  ;*TEST 42       TEST DBL (RHCS3 BIT #10) TEST C
10193
10194                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10195                                  ;*      SET UP FOR A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
10196                                  ;*      DO A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
10197                                  ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10198                                  ;*      CHECK RHCS3
10199                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10200
10201                                  ;;*****************************************************************
10202  030520  000004                  T$T42:  SCOPE
10203  030522  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10204  030526  012737  000042  004656          MOV     #42,@#T$TNM     ;SAVE TEST NUMBER
10205
10206  030534  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10207                                                                  ;SETUP UNIT NUBER
10208                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
```

```
10209
10210                                                   ;SET UP FOR A TEN WORD WRITE FROM AN ODD WORD BOUNDARY
10211  030540  012701  000003                          MOV     #3,R1           ;COUNT OF THREE
10212  030544  012702  003002                          MOV     #BFODD,R2       ;START THREE WORD BUFFER
10213                                                                          ;FROM AN ODD WORD BOUNDARY
10214  030550  012722  052525          1$:             MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10215  030554  005301                                  DEC     R1              ;COUNT
10216  030556  001374                                  BNE     1$              ;BRANCH IF THREE NOT DONE
10217  030560  012777  003002  153276                  MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
10218                                                                          ;FROM AN ODD WORD BOUNDARY
10219  030566  012777  177766  153266                  MOV #-10.,@RHWC ;WORD COUNT TEN
10220  030574  004077  154040                          JSR     R0,@COMND       ;GO TO DO COMMAND
10221  030600  004154                                  WRIDAT                  ;WRITE DATA
10222
10223
10224                                                   ;CHECK THAT RHCS3 HAS 0
10225  030602  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
10226
10227  030610  017737  153316  001126                  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10228  030616  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10229                                                                          ;DATA WITH DATA READ FROM
10230                                                                          ;RHCS3
10231  030624  001401                                  BEQ     65$             ;BRANCH IF GOOD
10232  030626  104124                                  ERROR   124
10233                                                                          ;AFTER SETTING "CLR" BIT #5
10234                                                                          ;IN RHCS2 TO INIT THE RH
10235                                                                          ;A TEN WORD WRITE FROM AN
10236                                                                          ;ODD WORD BOUNDARY WAS DONE
10237                                                                          ;THEN
10238                                                                          ;RHCS3 SHOULD HAVE 0
10239                                                                          ;BUT CONTAINED WHAT IS
10240                                                                          ;GIVEN IN BAD RHCS3
10241  030630                           65$:
10242  030630  042777  000076  153222                  BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10243                                                   ;CHECK THAT RHCS1 HAS RDY!DVA
10244  030636  012737  004200  001124                  MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10245
10246  030644  017737  153210  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10247  030652  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10248                                                                          ;DATA WITH DATA READ FROM
10249                                                                          ;RHCS1
10250  030660  001401                                  BEQ     67$             ;BRANCH IF GOOD
10251  030662  104125                                  ERROR   125
10252                                                                          ;AFTER SETTING "CLR" BIT #5
10253                                                                          ;IN RHCS2 TO INIT THE RH
10254                                                                          ;A TEN WORD WRITE FROM AN
10255                                                                          ;ODD WORD BOUNDARY WAS DONE
10256                                                                          ;THEN
10257                                                                          ;RHCS1 SHOULD HAVE RDY!DVA
10258                                                                          ;=4200
10259                                                                          ;BUT CONTAINED WHAT IS
10260                                                                          ;GIVEN IN BAD RHCS1
10261  030664                           67$:
10262                                                   ;CHECK THAT RHCS2 HAS IR
10263  030664  012737  000100  001124                  MOV     #IR,$GDDAT      ;GET GOOD = 100
10264  030672  053737  005010  001124                  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
```

```
10265
10266  030700  017737  153164  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10267  030706  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10268                                                                   ;DATA WITH DATA READ FROM
10269                                                                   ;RHCS2
10270  030714  001401                          BEQ     69$             ;BRANCH IF GOOD
10271  030716  104126                          ERROR   126
10272                                                                   ;AFTER SETTING "CLR" BIT #5
10273                                                                   ;IN RHCS2 TO INIT THE RH
10274                                                                   ;A TEN WORD WRITE FROM AN
10275                                                                   ;ODD WORD BOUNDARY WAS DONE
10276                                                                   ;THEN
10277                                                                   ;RHCS2 SHOULD HAVE IR
10278                                                                   ;=100
10279                                                                   ;TOGETHER WITH UNIT NUMBER
10280                                                                   ;BUT CONTAINED WHAT IS
10281                                                                   ;GIVEN IN BAD RHCS2
10282  030720                          69$:
10283                                          ;CHECK THAT RHWC HAS 0
10284  030720  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10285
10286  030726  017737  153130  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10287  030734  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10288                                                                   ;DATA WITH DATA READ FROM
10289                                                                   ;RHWC
10290  030742  001401                          BEQ     71$             ;BRANCH IF GOOD
10291  030744  104131                          ERROR   131
10292                                                                   ;AFTER SETTING "CLR" BIT #5
10293                                                                   ;IN RHCS2 TO INIT THE RH
10294                                                                   ;A TEN WORD WRITE FROM AN
10295                                                                   ;ODD WORD BOUNDARY WAS DONE
10296                                                                   ;THEN
10297                                                                   ;RHWC SHOULD HAVE 0
10298                                                                   ;BUT CONTAINED WHAT IS
10299                                                                   ;GIVEN IN BAD RHWC
10300  030746                          71$:
10301
10302
10303
10304                                  ;;*****************************************************************
10305                                  ;*TEST 43       TEST DBL (RHCS3 BIT #10) TEST D
10306
10307                                  ;*       CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10308                                  ;*       SET UP FOR A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10309                                  ;*       DO A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10310                                  ;*       THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10311                                  ;*       CHECK RHCS3
10312                                  ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10313
10314                                  ;;*****************************************************************
10315  030746  000004                  TST43:  SCOPE
10316  030750  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10317  030754  012737  000043  004656          MOV     #43,@#TSTNM     ;SAVE TEST NUMBER
10318
10319  030762  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10320                                                                   ;SETUP UNIT NUBER
```

```
10321                                                           ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10322
10323                                                   ;SET UP FOR A ELEVEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10324  030766  012701  000003                 MOV     #3,R1          ;COUNT OF THREE
10325  030772  012702  003000                 MOV     #BFEVEN,R2     ;START THREE WORD BUFFER
10326                                                                 ;FROM AN EVEN WORD BOUNDARY
10327  030776  012722  052525          1$:    MOV     #52525,(R2)+   ;MOVE THREE 52525 INTO BUFFER
10328  031002  005301                 DEC     R1             ;COUNT
10329  031004  001374                 BNE     1$             ;BRANCH IF THREE NOT DONE
10330  031006  012777  003000  153050 MOV     #BFEVEN,@RHBA  ;SET BUS ADDRESS TO START
10331                                                                 ;FROM AN EVEN WORD BOUNDARY
10332  031014  012777  177765  153040 MOV #-11.,@RHWC ;WORD COUNT ELEVEN
10333  031022  004077  153612         JSR     R0,@COMND      ;GO TO DO COMMAND
10334  031026  004154                 WRIDAT                 ;WRITE DATA
10335
10336
10337                                                   ;CHECK THAT RHCS3 HAS 0
10338  031030  012737  000000  001124 MOV     #0,$GDDAT      ;GET GOOD = 0
10339
10340  031036  017737  153070  001126 MOV     @RHCS3,$BDDAT  ;READ RHCS3 FOR COMPARISON
10341  031044  023737  001124  001126 CMP     $GDDAT,$BDDAT  ;COMPARE EXPECTED
10342                                                                 ;DATA WITH DATA READ FROM
10343                                                                 ;RHCS3
10344  031052  001401                 BEQ     65$            ;BRANCH IF GOOD
10345  031054  104124                 ERROR   124
10346                                                                 ;AFTER SETTING "CLR" BIT #5
10347                                                                 ;IN RHCS2 TO INIT THE RH
10348                                                                 ;A ELEVEN WORD WRITE FROM AN
10349                                                                 ;EVEN WORD BOUNDARY WAS DONE
10350                                                                 ;THEN
10351                                                                 ;RHCS3 SHOULD HAVE 0
10352                                                                 ;BUT CONTAINED WHAT IS
10353                                                                 ;GIVEN IN BAD RHCS3
10354  031056                  65$:
10355  031056  042777  000076  152774 BIC     #76,@RHCS1     ;CLEAR FUNCTION BITS
10356                                                   ;CHECK THAT RHCS1 HAS RDY!DVA
10357  031064  012737  004200  001124 MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10358
10359  031072  017737  152762  001126 MOV     @RHCS1,$BDDAT  ;READ RHCS1 FOR COMPARISON
10360  031100  023737  001124  001126 CMP     $GDDAT,$BDDAT  ;COMPARE EXPECTED
10361                                                                 ;DATA WITH DATA READ FROM
10362                                                                 ;RHCS1
10363  031106  001401                 BEQ     67$            ;BRANCH IF GOOD
10364  031110  104125                 ERROR   125
10365                                                                 ;AFTER SETTING "CLR" BIT #5
10366                                                                 ;IN RHCS2 TO INIT THE RH
10367                                                                 ;A ELEVEN WORD WRITE FROM AN
10368                                                                 ;EVEN WORD BOUNDARY WAS DONE
10369                                                                 ;THEN
10370                                                                 ;RHCS1 SHOULD HAVE RDY!DVA
10371                                                                 ;=4200
10372                                                                 ;BUT CONTAINED WHAT IS
10373                                                                 ;GIVEN IN BAD RHCS1
10374  031112                  67$:
10375                                                   ;CHECK THAT RHCS2 HAS IR
10376  031112  012737  000100  001124 MOV     #IR,$GDDAT     ;GET GOOD = 100
```

```
10377  031120  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10378
10379  031126  017737  152736  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10380  031134  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10381                                                                  ;DATA WITH DATA READ FROM
10382                                                                  ;RHCS2
10383  031142  001401                          BEQ     69$             ;BRANCH IF GOOD
10384  031144  104126                          ERROR   126
10385                                                                  ;AFTER SETTING "CLR" BIT #5
10386                                                                  ;IN RHCS2 TO INIT THE RH
10387                                                                  ;A ELEVEN WORD WRITE FROM AN
10388                                                                  ;EVEN WORD BOUNDARY WAS DONE
10389                                                                  ;THEN
10390                                                                  ;RHCS2 SHOULD HAVE IR
10391                                                                  ;=100
10392                                                                  ;TOGETHER WITH UNIT NUMBE.
10393                                                                  ;BUT CONTAINED WHAT IS
10394                                                                  ;GIVEN IN BAD RHCS2
10395  031146                          69$:
10396                                  ;CHECK THAT RHWC HAS 0
10397  031146  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10398
10399  031154  017737  152702  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10400  031162  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10401                                                                  ;DATA WITH DATA READ FROM
10402                                                                  ;RHWC
10403  031170  001401                          BEQ     71$             ;BRANCH IF GOOD
10404  031172  104131                          ERROR   131
10405                                                                  ;AFTER SETTING "CLR" BIT #5
10406                                                                  ;IN RHCS2 TO INIT THE RH
10407                                                                  ;A ELEVEN WORD WRITE FROM AN
10408                                                                  ;EVEN WORD BOUNDARY WAS DONE
10409                                                                  ;THEN
10410                                                                  ;RHWC SHOULD HAVE 0
10411                                                                  ;BUT CONTAINED WHAT IS
10412                                                                  ;GIVEN IN BAD RHWC
10413  031174                          71$:
10414
10415
10416
10417                                  ;;*****************************************************************
10418                                  ;*TEST 44       TEST DBL (RHCS3 BIT #10) TEST E
10419
10420                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10421                                  ;*      SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10422                                  ;*      WITH BAI IN RHCS2 BIT #3 SET
10423                                  ;*      DO A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10424                                  ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10425                                  ;*      CHECK RHCS3
10426                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10427
10428                                  ;;*****************************************************************
10429  031174  000004                  ↑ST44:  SCOPE
10430  031176  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10431  031202  012737  000044  004656          MOV     #44,@#↑STNM     ;SAVE TEST NUMBER
10432
```

```
10433  031210  004737  040306                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10434                                                                     ;SETUP UNIT NUBER
10435                                                                     ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10436
10437                                             ;SET UP FOR A TEN WORD WRITE FROM AN EVEN WORD BOUNDARY
10438  031214  012701  000003                    MOV     #3,R1           ;COUNT OF THREE
10439  031220  012702  003000                    MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10440                                                                     ;FROM AN EVEN WORD BOUNDARY
10441  031224  012722  052525          1$:      MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10442  031230  005301                           DEC     R1              ;COUNT
10443  031232  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
10444  031234  012777  003000  152622           MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10445                                                                     ;FROM AN EVEN WORD BOUNDARY
10446  031242  012777  177766  152612           MOV #-10.,@RHWC ;WORD COUNT TEN
10447  031250  052777  000010  152612  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
10448  031256  004077  153356                    JSR     R0,@COMND       ;GO TO DO COMMAND
10449  031262  004154                           WRIDAT                  ;WRITE DATA
10450
10451
10452                                             ;CHECK THAT RHCS3 HAS 0
10453  031264  012737  000000  001124           MOV     #0,$GDDAT       ;GET GOOD = 0
10454
10455  031272  017737  152634  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10456  031300  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10457                                                                     ;DATA WITH DATA READ FROM
10458                                                                     ;RHCS3
10459  031306  001401                           BEQ     65$             ;BRANCH IF GOOD
10460  031310  104124                           ERROR   124
10461                                                                     ;AFTER SETTING "CLR" BIT #5
10462                                                                     ;IN RHCS2 TO INIT THE RH
10463                                                                     ;A TEN WORD WRITE FROM AN
10464                                                                     ;EVEN WORD BOUNDARY WAS DONE
10465                                                                     ;WITH BAI IN RHCS2 SET
10466                                                                     ;THEN
10467                                                                     ;RHCS3 SHOULD HAVE 0
10468                                                                     ;BUT CONTAINED WHAT IS
10469                                                                     ;GIVEN IN BAD RHCS3
10470  031312                           65$:
10471  031312  042777  000076  152540           BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10472                                             ;CHECK THAT RHCS1 HAS RDY!DVA
10473  031320  012737  004200  001124           MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10474
10475  031326  017737  152526  001126           MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10476  031334  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10477                                                                     ;DATA WITH DATA READ FROM
10478                                                                     ;RHCS1
10479  031342  001401                           BEQ     67$             ;BRANCH IF GOOD
10480  031344  104125                           ERROR   125
10481                                                                     ;AFTER SETTING "CLR" BIT #5
10482                                                                     ;IN RHCS2 TO INIT THE RH
10483                                                                     ;A TEN WORD WRITE FROM AN
10484                                                                     ;EVEN WORD BOUNDARY WAS DONE
10485                                                                     ;WITH BAI IN RHCS2 SET
10486                                                                     ;THEN
10487                                                                     ;RHCS1 SHOULD HAVE RDY!DVA
10488                                                                     ;=4200
```

```
10489                                                                   ;BUT CONTAINED WHAT IS
10490                                                                   ;GIVEN IN BAD RHCS1
10491   031346                                  67$:
10492                                           ;CHECK THAT RHCS2 HAS IR!BAI
10493   031346  012737  000110  001124          MOV     #IR!BAI,$GDDAT  ;GET GOOD = 110
10494   031354  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10495
10496   031362  017737  152502  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10497   031370  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10498                                                                   ;DATA WITH DATA READ FROM
10499                                                                   ;RHCS2
10500   031376  001401                          BEQ     69$             ;BRANCH IF GOOD
10501   031400  104126                          ERROR   126
10502                                                                   ;AFTER SETTING "CLR" BIT #5
10503                                                                   ;IN RHCS2 TO INIT THE RH
10504                                                                   ;A TEN WORD WRITE FROM AN
10505                                                                   ;EVEN WORD BOUNDARY WAS DONE
10506                                                                   ;WITH BAI IN RHCS2 SET
10507                                                                   ;THEN
10508                                                                   ;RHCS2 SHOULD HAVE IR!BAI
10509                                                                   ;=110
10510                                                                   ;TOGETHER WITH UNIT NUMBER
10511                                                                   ;BUT CONTAINED WHAT IS
10512                                                                   ;GIVEN IN BAD RHCS2
10513   031402                                  69$:
10514                                           ;CHECK THAT RHWC HAS 0
10515   031402  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10516
10517   031410  017737  152446  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10518   031416  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10519                                                                   ;DATA WITH DATA READ FROM
10520                                                                   ;RHWC
10521   031424  001401                          BEQ     71$             ;BRANCH IF GOOD
10522   031426  104131                          ERROR   131
10523                                                                   ;AFTER SETTING "CLR" BIT #5
10524                                                                   ;IN RHCS2 TO INIT THE RH
10525                                                                   ;A TEN WORD WRITE FROM AN
10526                                                                   ;EVEN WORD BOUNDARY WAS DONE
10527                                                                   ;WITH BAI IN RHCS2 SET
10528                                                                   ;THEN
10529                                                                   ;RHWC SHOULD HAVE 0
10530                                                                   ;BUT CONTAINED WHAT IS
10531                                                                   ;GIVEN IN BAD RHWC
10532   031430                                  71$:
10533
10534
10535
10536                                           ;;*****************************************************************
10537                                           ;*TEST 45       TEST DBL (RHCS3 BIT #10) TEST F
10538
10539                                           ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10540                                           ;*      SET UP FOR A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
10541                                           ;*      DO A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
10542                                           ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
10543                                           ;*      CHECK RHCS3
10544                                           ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
```

```
10545
10546                                          ;;*****************************************************************
10547  031430  000004                         T$T45:  SCOPE
10548  031432  012706  001000                         MOV     #STACK,SP       ;RESET STACK
10549  031436  012737  000045  004656                 MOV     #45,@#T$TNM     ;SAVE TEST NUMBER
10550
10551  031444  004737  040306                         JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10552                                                                         ;SETUP UNIT NUBER
10553                                                                         ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10554
10555                                                  ;SET UP FOR A ELEVEN WORD WRITE FROM AN ODD WORD BOUNDARY
10556  031450  012701  000003                         MOV     #3,R1           ;COUNT OF THREE
10557  031454  012702  003002                         MOV     #BFODD,R2       ;START THREE WORD BUFFER
10558                                                                         ;FROM AN ODD WORD BOUNDARY
10559  031460  012722  052525                 1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10560  031464  005301                                 DEC     R1              ;COUNT
10561  031466  001374                                 BNE     1$              ;BRANCH IF THREE NOT DONE
10562  031470  012777  003002  152366                 MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
10563                                                                         ;FROM AN ODD WORD BOUNDARY
10564  031476  012777  177765  152356                 MOV #-11.,@RHWC ;WORD COUNT ELEVEN
10565  031504  004077  153130                         JSR     R0,@COMND       ;GO TO DO COMMAND
10566  031510  004154                                 WRIDAT                  ;WRITE DATA
10567
10568
10569                                                  ;CHECK THAT RHCS3 HAS DBL
10570  031512  012737  002000  001124                 MOV     #DBL,$GDDAT     ;GET GOOD = 2000
10571
10572  031520  017737  152406  001126                 MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10573  031526  023737  001124  001126                 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10574                                                                         ;DATA WITH DATA READ FROM
10575                                                                         ;RHCS3
10576  031534  001401                                 BEQ     65$             ;BRANCH IF GOOD
10577  031536  104124                                 ERROR   124
10578                                                                         ;AFTER SETTING "CLR" BIT #5
10579                                                                         ;IN RHCS2 TO INIT THE RH
10580                                                                         ;A ELEVEN WORD WRITE FROM AN
10581                                                                         ;ODD WORD BOUNDARY WAS DONE
10582                                                                         ;THEN
10583                                                                         ;RHCS3 SHOULD HAVE DBL
10584                                                                         ;=2000
10585                                                                         ;BUT CONTAINED WHAT IS
10586                                                                         ;GIVEN IN BAD RHCS3
10587  031540                                  65$:
10588  031540  042777  000076  152312                 BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10589                                                  ;CHECK THAT RHCS1 HAS RDY!DVA
10590  031546  012737  004200  001124                 MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10591
10592  031554  017737  152300  001126                 MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10593  031562  023737  001124  001126                 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10594                                                                         ;DATA WITH DATA READ FROM
10595                                                                         ;RHCS1
10596  031570  001401                                 BEQ     67$             ;BRANCH IF GOOD
10597  031572  104125                                 ERROR   125
10598                                                                         ;AFTER SETTING "CLR" BIT #5
10599                                                                         ;IN RHCS2 TO INIT THE RH
10600                                                                         ;A ELEVEN WORD WRITE FROM AN
```

```
10601                                                          ;ODD WORD BOUNDARY WAS DONE
10602                                                          ;THEN
10603                                                          ;RHCS1 SHOULD HAVE RDY!DVA
10604                                                          ;=4200
10605                                                          ;BUT CONTAINED WHAT IS
10606                                                          ;GIVEN IN BAD RHCS1
10607   031574                                 67$:
10608                                          ;CHECK THAT RHCS2 HAS IR
10609   031574  012737  000100  001124         MOV     #IR,$GDDAT      ;GET GOOD = 100
10610   031602  053737  005010  001124         BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10611
10612   031610  017737  152254  001126         MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10613   031616  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10614                                                                  ;DATA WITH DATA READ FROM
10615                                                                  ;RHCS2
10616   031624  001401                         BEQ     69$             ;BRANCH IF GOOD
10617   031626  104126                         ERROR   126
10618                                                                  ;AFTER SETTING "CLR" BIT #5
10619                                                                  ;IN RHCS2 TO INIT THE RH
10620                                                                  ;A ELEVEN WORD WRITE FROM AN
10621                                                                  ;ODD WORD BOUNDARY WAS DONE
10622                                                                  ;THEN
10623                                                                  ;RHCS2 SHOULD HAVE IR
10624                                                                  ;=100
10625                                                                  ;TOGETHER WITH UNIT NUMBER
10626                                                                  ;BUT CONTAINED WHAT IS
10627                                                                  ;GIVEN IN BAD RHCS2
10628   031630                                 69$:
10629                                          ;CHECK THAT RHWC HAS 0
10630   031630  012737  000000  001124         MOV     #0,$GDDAT       ;GET GOOD = 0
10631
10632   031636  017737  152220  001126         MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10633   031644  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10634                                                                  ;DATA WITH DATA READ FROM
10635                                                                  ;RHWC
10636   031652  001401                         BEQ     71$             ;BRANCH IF GOOD
10637   031654  104131                         ERROR   131
10638                                                                  ;AFTER SETTING "CLR" BIT #5
10639                                                                  ;IN RHCS2 TO INIT THE RH
10640                                                                  ;A ELEVEN WORD WRITE FROM AN
10641                                                                  ;ODD WORD BOUNDARY WAS DONE
10642                                                                  ;THEN
10643                                                                  ;RHWC SHOULD HAVE 0
10644                                                                  ;BUT CONTAINED WHAT IS
10645                                                                  ;GIVEN IN BAD RHWC
10646   031656                                 71$:
10647
10648
10649
10650
10651                                          ;;*****************************************************************
10652                                          ;*TEST 46        TEST DBL (RHCS3 BIT #10) TEST G
10653
10654                                          ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10655                                          ;*      SET UP FOR A NINE WORD READ FROM AN EVEN WORD BOUNDARY
10656                                          ;*      DO A NINE WORD READ FROM AN EVEN WORD BOUNDARY
```

```
10657                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10658                                   ;*      CHECK RHCS3
10659                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10660
10661                                   ;;**********************************************************************
10662  031656  000004                  T$T46:  SCOPE
10663  031660  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10664  031664  012737  000046  004656          MOV     #46.,@#T$TNM    ;SAVE TEST NUMBER
10665
10666  031672  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10667                                                                  ;SETUP UNIT NUBER
10668                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10669
10670                                          ;SET UP FOR A NINE WORD READ FROM AN EVEN WORD BOUNDARY
10671  031676  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
10672  031702  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10673                                                                  ;FROM AN EVEN WORD BOUNDARY
10674  031706  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10675  031712  005301                          DEC     R1              ;COUNT
10676  031714  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
10677  031716  012777  003000  152140          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10678                                                                  ;FROM AN EVEN WORD BOUNDARY
10679  031724  012777  177767  152130          MOV #-9.,@RHWC ;WORD COUNT NINE
10680  031732  004077  152702                  JSR     R0,@COMND       ;GO TO DO COMMAND
10681  031736  004160                          READAT                  ;READ DATA
10682
10683
10684                                          ;CHECK THAT RHCS3 HAS 0
10685  031740  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10686
10687  031746  017737  152160  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10688  031754  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10689                                                                  ;DATA WITH DATA READ FROM
10690                                                                  ;RHCS3
10691  031762  001401                          BEQ     65$             ;BRANCH IF GOOD
10692  031764  104124                          ERROR   124
10693                                                                  ;AFTER SETTING "CLR" BIT #5
10694                                                                  ;IN RHCS2 TO INIT THE RH
10695                                                                  ;A NINE WORD READ FROM AN
10696                                                                  ;EVEN WORD BOUNDARY WAS DONE
10697                                                                  ;THEN
10698                                                                  ;RHCS3 SHOULD HAVE 0
10699                                                                  ;BUT CONTAINED WHAT IS
10700                                                                  ;GIVEN IN BAD RHCS3
10701  031766                  65$:
10702  031766  042777  000076  152064          BIC     #76.,@RHCS1     ;CLEAR FUNCTION BITS
10703                                          ;CHECK THAT RHCS1 HAS RDY!DVA
10704  031774  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10705
10706  032002  017737  152052  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10707  032010  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10708                                                                  ;DATA WITH DATA READ FROM
10709                                                                  ;RHCS1
10710  032016  001401                          BEQ     67$             ;BRANCH IF GOOD
10711  032020  104125                          ERROR   125
10712                                                                  ;AFTER SETTING "CLR" BIT #5
```

```
10713                                                                   ;IN RHCS2 TO INIT THE RH
10714                                                                   ;A NINE WORD READ FROM AN
10715                                                                   ;EVEN WORD BOUNDARY WAS DONE
10716                                                                   ;THEN
10717                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
10718                                                                   ;=4200
10719                                                                   ;BUT CONTAINED WHAT IS
10720                                                                   ;GIVEN IN BAD RHCS1
10721   032022                                  67$:
10722                                           ;CHECK THAT RHCS2 HAS IR
10723   032022  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10724   032030  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10725
10726   032036  017737  152026  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10727   032044  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10728                                                                   ;DATA WITH DATA READ FROM
10729                                                                   ;RHCS2
10730   032052  001401                          BEQ     69$             ;BRANCH IF GOOD
10731   032054  104126                          ERROR   126
10732                                                                   ;AFTER SETTING "CLR" BIT #5
10733                                                                   ;IN RHCS2 TO INIT THE RH
10734                                                                   ;A NINE WORD READ FROM AN
10735                                                                   ;EVEN WORD BOUNDARY WAS DONE
10736                                                                   ;THEN
10737                                                                   ;RHCS2 SHOULD HAVE IR
10738                                                                   ;=100
10739                                                                   ;TOGETHER WITH UNIT NUMBER
10740                                                                   ;BUT CONTAINED WHAT IS
10741                                                                   ;GIVEN IN BAD RHCS2
10742   032056                                  69$:
10743                                           ;CHECK THAT RHWC HAS 0
10744   032056  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10745
10746   032064  017737  151772  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10747   032072  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10748                                                                   ;DATA WITH DATA READ FROM
10749                                                                   ;RHWC
10750   032100  001401                          BEQ     71$             ;BRANCH IF GOOD
10751   032102  104131                          ERROR   131
10752                                                                   ;AFTER SETTING "CLR" BIT #5
10753                                                                   ;IN RHCS2 TO INIT THE RH
10754                                                                   ;A NINE WORD READ FROM AN
10755                                                                   ;EVEN WORD BOUNDARY WAS DONE
10756                                                                   ;THEN
10757                                                                   ;RHWC SHOULD HAVE 0
10758                                                                   ;BUT CONTAINED WHAT IS
10759                                                                   ;GIVEN IN BAD RHWC
10760   032104                          71$:
10761
10762
10763
10764                                   ;;*****************************************************************
10765                                   ;*TEST 47        TEST DBL (RHCS3 BIT #10) TEST H
10766
10767                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10768                                   ;*      SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
```

```
10769                                   ;*      DO A TEN WORD READ FROM AN EVEN WORD BOUNDARY
10770                                   ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
10771                                   ;*      CHECK RHCS3
10772                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
10773
10774                                   ;;*****************************************************************
10775  032104  000004                  ↑ST47:  SCOPE
10776  032106  012706  001000                  MOV     #STACK,SP       ;RESET STACK
10777  032112  012737  000047  004656          MOV     #47,@#↑STNM     ;SAVE TEST NUMBER
10778
10779  032120  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
10780                                                                   ;SETUP UNIT NUBER
10781                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
10782
10783                                   ;SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
10784  032124  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
10785  032130  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
10786                                                                   ;FROM AN EVEN WORD BOUNDARY
10787  032134  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
10788  032140  005301                          DEC     R1              ;COUNT
10789  032142  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
10790  032144  012777  003000  151712          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
10791                                                                   ;FROM AN EVEN WORD BOUNDARY
10792  032152  012777  177766  151702          MOV #-10.,@RHWC ;WORD COUNT TEN
10793  032160  004077  152454                  JSR     R0,@COMND       ;GO TO DO COMMAND
10794  032164  004160                          READAT                  ;READ DATA
10795
10796
10797                                   ;CHECK THAT RHCS3 HAS DBL
10798  032166  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
10799
10800  032174  017737  151732  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
10801  032202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10802                                                                   ;DATA WITH DATA READ FROM
10803                                                                   ;RHCS3
10804  032210  001401                          BEQ     65$             ;BRANCH IF GOOD
10805  032212  104124                          ERROR   124
10806                                                                   ;AFTER SETTING "CLR" BIT #5
10807                                                                   ;IN RHCS2 TO INIT THE RH
10808                                                                   ;A TEN WORD READ FROM AN
10809                                                                   ;EVEN WORD BOUNDARY WAS DONE
10810                                                                   ;THEN
10811                                                                   ;RHCS3 SHOULD HAVE DBL
10812                                                                   ;=2000
10813                                                                   ;BUT CONTAINED WHAT IS
10814                                                                   ;GIVEN IN BAD RHCS3
10815  032214                           65$:
10816  032214  042777  000076  151636          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
10817                                  ;CHECK THAT RHCS1 HAS RDY!DVA
10818  032222  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
10819
10820  032230  017737  151624  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
10821  032236  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10822                                                                   ;DATA WITH DATA READ FROM
10823                                                                   ;RHCS1
10824  032244  001401                          BEQ     67$             ;BRANCH IF GOOD
```

```
10825  032246  104125                          ERROR   125
10826                                                       ;AFTER SETTING "CLR" BIT #5
10827                                                       ;IN RHCS2 TO INIT THE RH
10828                                                       ;A TEN WORD READ FROM AN
10829                                                       ;EVEN WORD BOUNDARY WAS DONE
10830                                                       ;THEN
10831                                                       ;RHCS1 SHOULD HAVE RDY!DVA
10832                                                       ;=4200
10833                                                       ;BUT CONTAINED WHAT IS
10834                                                       ;GIVEN IN BAD RHCS1
10835  032250                          67$:
10836                                      ;CHECK THAT RHCS2 HAS IR
10837  032250  012737  000100  001124      MOV     #IR,$GDDAT      ;GET GOOD = 100
10838  032256  053737  005010  001124      BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10839
10840  032264  017737  151600  001126      MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10841  032272  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10842                                                       ;DATA WITH DATA READ FROM
10843                                                       ;RHCS2
10844  032300  001401                      BEQ     69$             ;BRANCH IF GOOD
10845  032302  104126                      ERROR   126
10846                                                       ;AFTER SETTING "CLR" BIT #5
10847                                                       ;IN RHCS2 TO INIT THE RH
10848                                                       ;A TEN WORD READ FROM AN
10849                                                       ;EVEN WORD BOUNDARY WAS DONE
10850                                                       ;THEN
10851                                                       ;RHCS2 SHOULD HAVE IR
10852                                                       ;=100
10853                                                       ;TOGETHER WITH UNIT NUMBER
10854                                                       ;BUT CONTAINED WHAT IS
10855                                                       ;GIVEN IN BAD RHCS2
10856  032304                          69$:
10857                                      ;CHECK THAT RHWC HAS 0
10858  032304  012737  000000  001124      MOV     #0,$GDDAT       ;GET GOOD = 0
10859
10860  032312  017737  151544  001126      MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10861  032320  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10862                                                       ;DATA WITH DATA READ FROM
10863                                                       ;RHWC
10864  032326  001401                      BEQ     71$             ;BRANCH IF GOOD
10865  032330  104131                      ERROR   131
10866                                                       ;AFTER SETTING "CLR" BIT #5
10867                                                       ;IN RHCS2 TO INIT THE RH
10868                                                       ;A TEN WORD READ FROM AN
10869                                                       ;EVEN WORD BOUNDARY WAS DONE
10870                                                       ;THEN
10871                                                       ;RHWC SHOULD HAVE 0
10872                                                       ;BUT CONTAINED WHAT IS
10873                                                       ;GIVEN IN BAD RHWC
10874  032332                          71$:
10875
10876
10877
10878                                  ;;*****************************************************************
10879                                  ;*TEST 50        TEST DBL (RHCS3 BIT #10) TEST I
10880
```

```
                                        TST0     TEST DBL (RHCS3 BIT #10) TEST I                                    SEQ 0204

                                        ;*       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
                                        ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC

                                        ;;*****************************************************************
032332  000004                        TST50:  SCOPE
032334  012706  001000                        MOV     #STACK,SP       ;RESET STACK
032340  012737  000050  004656                MOV     #50,@#TSTNM     ;SAVE TEST NUMBER

032346  004737  040306                        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 ODD WORD BOUNDARY
032352  012701  000003                        MOV     #3,R1           ;COUNT OF THREE
032356  012702  003002                        MOV     #BFODD,R2       ;START THREE WORD BUFFER
                                                                        ;FROM AN ODD WORD BOUNDARY
032362  012722  052525                1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
032366  005301                        DEC     R1              ;COUNT
032370  001374                        BNE     1$              ;BRANCH IF THREE NOT DONE
032372  012777  003002  151464                MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
                                                                        ;FROM AN ODD WORD BOUNDARY
032400  012777  177766  151454                MOV #-10.,@RHWC ;WORD COUNT TEN
032406  004077  152226                        JSR     R0,@COMND       ;GO TO DO COMMAND
032412  004160                        READAT                          ;READ DATA

                                        ;CHECK THAT RHCS3 HAS 0
C32414  012737  000000  001124                MOV     #0,$GDDAT       ;GET GOOD = 0

032422  017737  151504  001126                MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
032430  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS3
032436  001401                        BEQ     65$             ;BRANCH IF GOOD
032440  104124                        ERROR   124
                                                                        ;AFTER SETTING "CLR" BIT #5
                                                                        ;IN RHCS2 TO INIT THE RH
                                                                        ;A TEN WORD READ FROM AN
                                                                        ;ODD WORD BOUNDARY WAS DONE
                                                                        ;THEN
                                                                        ;RHCS3 SHOULD HAVE 0
                                                                        ;BUT CONTAINED WHAT IS
                                                                        ;GIVEN IN BAD RHCS3
032442                          65$:
032442  042777  000076  151410                BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
                                        ;CHECK THAT RHCS1 HAS RDY!DVA
032450  012737  004200  001124                MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200

032456  017737  151376  001126                MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
032464  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                        ;DATA WITH DATA READ FROM
                                                                        ;RHCS1
```

```
10937  032472  001401                          BEQ     67$             ;BRANCH IF GOOD
10938  032474  104125                          ERROR   125
10939                                                                   ;AFTER SETTING "CLR" BIT #5
10940                                                                   ;IN RHCS2 TO INIT THE RH
10941                                                                   ;A TEN WORD READ FROM AN
10942                                                                   ;ODD WORD BOUNDARY WAS DONE
10943                                                                   ;THEN
10944                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
10945                                                                   ;=4200
10946                                                                   ;BUT CONTAINED WHAT IS
10947                                                                   ;GIVEN IN BAD RHCS1
10948  032476                          67$:
10949                                          ;CHECK THAT RHCS2 HAS IR
10950  032476  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
10951  032504  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
10952
10953  032512  017737  151352  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
10954  032520  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10955                                                                  ;DATA WITH DATA READ FROM
10956                                                                  ;RHCS2
10957  032526  001401                          BEQ     69$             ;BRANCH IF GOOD
10958  032530  104126                          ERROR   126
10959                                                                  ;AFTER SETTING "CLR" BIT #5
10960                                                                  ;IN RHCS2 TO INIT THE RH
10961                                                                  ;A TEN WORD READ FROM AN
10962                                                                  ;ODD WORD BOUNDARY WAS DONE
10963                                                                  ;THEN
10964                                                                  ;RHCS2 SHOULD HAVE IR
10965                                                                  ;=100
10966                                                                  ;TOGETHER WITH UNIT NUMBER
10967                                                                  ;BUT CONTAINED WHAT IS
10968                                                                  ;GIVEN IN BAD RHCS2
10969  032532                          69$:
10970                                          ;CHECK THAT RHWC HAS 0
10971  032532  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
10972
10973  032540  017737  151316  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
10974  032546  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
10975                                                                  ;DATA WITH DATA READ FROM
10976                                                                  ;RHWC
10977  032554  001401                          BEQ     71$             ;BRANCH IF GOOD
10978  032556  104131                          ERROR   131
10979                                                                  ;AFTER SETTING "CLR" BIT #5
10980                                                                  ;IN RHCS2 TO INIT THE RH
10981                                                                  ;A TEN WORD READ FROM AN
10982                                                                  ;ODD WORD BOUNDARY WAS DONE
10983                                                                  ;THEN
10984                                                                  ;RHWC SHOULD HAVE 0
10985                                                                  ;BUT CONTAINED WHAT IS
10986                                                                  ;GIVEN IN BAD RHWC
10987  032560                          71$:
10988
10989
10990
10991                                  ;;*****************************************************************
10992                                  ;*TEST 51        TEST DBL (RHCS3 BIT #10) TEST J
```

```
10993
10994                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
10995                                   ;*      SET UP FOR A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
10996                                   ;*      DO A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
10997                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
10998                                   ;*      CHECK RHCS3
10999                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11000
11001                                   ;;*****************************************************************
11002  032560  000004                   TST51:  SCOPE
11003  032562  012706  001000                   MOV     #STACK,SP       ;RESET STACK
11004  032566  012737  000051  004656           MOV     #51,@#TSTNM     ;SAVE TEST NUMBER
11005
11006  032574  004737  040306                   JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11007                                                                   ;SETUP UNIT NUBER
11008                                                                   ;CLEAR RHWC AND FUNCTION JITS IN RHCS1
11009
11010                                   ;SET UP FOR A ELEVEN WORD READ FROM AN EVEN WORD BOUNDARY
11011  032600  012701  000003                   MOV     #3,R1           ;COUNT OF THREE
11012  032604  012702  003000                   MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11013                                                                   ;FROM AN EVEN WORD BOUNDARY
11014  032610  012722  052525           1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11015  032614  005301                           DEC     R1              ;COUNT
11016  032616  001374                           BNE     1$              ;BRANCH IF THREE NOT DONE
11017  032620  012777  003000  151236           MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
11018                                                                   ;FROM AN EVEN WORD BOUNDARY
11019  032626  012777  177765  151226           MOV #-11.,@RHWC ;WORD COUNT ELEVEN
11020  032634  004077  152000                   JSR     R0,@COMND       ;GO TO DO COMMAND
11021  032640  004160                           READAT                  ;READ DATA
11022
11023
11024                                   ;CHECK THAT RHCS3 HAS 0
11025  032642  012737  000000  001124           MOV     #0,$GDDAT       ;GET GOOD = 0
11026
11027  032650  017737  151256  001126           MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11028  032656  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11029                                                                   ;DATA WITH DATA READ FROM
11030                                                                   ;RHCS3
11031  032664  001401                           BEQ     65$             ;BRANCH IF GOOD
11032  032666  104124                           ERROR   124
11033                                                                   ;AFTER SETTING "CLR" BIT #5
11034                                                                   ;IN RHCS2 TO INIT THE RH
11035                                                                   ;A ELEVEN WORD READ FROM AN
11036                                                                   ;EVEN WORD BOUNDARY WAS DONE
11037                                                                   ;THEN
11038                                                                   ;RHCS3 SHOULD HAVE 0
11039                                                                   ;BUT CONTAINED WHAT IS
11040                                                                   ;GIVEN IN BAD RHCS3
11041  032670                           65$:
11042  032670  042777  000076  151162           BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11043                                   ;CHECK THAT RHCS1 HAS RDY!DVA
11044  032676  012737  004200  001124           MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
11045
11046  032704  017737  151150  001126           MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11047  032712  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11048                                                                   ;DATA WITH DATA READ FROM
```

```
11049
11050  032720  001401                          BEQ     67$             ;RHCS1
11051  032722  104125                          ERROR   125             ;BRANCH IF GOOD
11052                                                                   ;AFTER SETTING "CLR" BIT #5
11053                                                                   ;IN RHCS2 TO INIT THE RH
11054                                                                   ;A ELEVEN WORD READ FROM AN
11055                                                                   ;EVEN WORD BOUNDARY WAS DONE
11056                                                                   ;THEN
11057                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
11058                                                                   ;=4200
11059                                                                   ;BUT CONTAINED WHAT IS
11060                                                                   ;GIVEN IN BAD RHCS1
11061  032724                          67$:
11062                                          ;CHECK THAT RHCS2 HAS IR
11063  032724  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
11064  032732  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11065
11066  032740  017737  151124  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11067  032746  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11068                                                                   ;DATA WITH DATA READ FROM
11069                                                                   ;RHCS2
11070  032754  001401                          BEQ     69$             ;BRANCH IF GOOD
11071  032756  104126                          ERROR   126
11072                                                                   ;AFTER SETTING "CLR" BIT #5
11073                                                                   ;IN RHCS2 TO INIT THE RH
11074                                                                   ;A ELEVEN WORD READ FROM AN
11075                                                                   ;EVEN WORD BOUNDARY WAS DONE
11076                                                                   ;THEN
11077                                                                   ;RHCS2 SHOULD HAVE IR
11078                                                                   ;=100
11079                                                                   ;TOGETHER WITH UNIT NUMBER
11080                                                                   ;BUT CONTAINED WHAT IS
11081                                                                   ;GIVEN IN BAD RHCS2
11082  032760                          69$:
11083                                          ;CHECK THAT RHWC HAS 0
11084  032760  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11085
11086  032766  017737  151070  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11087  032774  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11088                                                                   ;DATA WITH DATA READ FROM
11089                                                                   ;RHWC
11090  033002  001401                          BEQ     71$             ;BRANCH IF GOOD
11091  033004  104131                          ERROR   131
11092                                                                   ;AFTER SETTING "CLR" BIT #5
11093                                                                   ;IN RHCS2 TO INIT THE RH
11094                                                                   ;A ELEVEN WORD READ FROM AN
11095                                                                   ;EVEN WORD BOUNDARY WAS DONE
11096                                                                   ;THEN
11097                                                                   ;RHWC SHOULD HAVE 0
11098                                                                   ;BUT CONTAINED WHAT IS
11099                                                                   ;GIVEN IN BAD RHWC
11100  033006                          71$:
11101
11102
11103
11104                                  ;;*****************************************************************
```

```
11105                                   ;*TEST 52       TEST DBL (RHCS3 BIT #10) TEST K
11106
11107                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11108                                   ;*      SET UP FOR A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
11109                                   ;*      DO A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
11110                                   ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
11111                                   ;*      CHECK RHCS3
11112                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11113
11114                                   ;;*****************************************************************
11115  033006  000004                  TST52:  SCOPE
11116  033010  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11117  033014  012737  000052  004656          MOV     #52,@#TSTNM     ;SAVE TEST NUMBER
11118
11119  033022  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11120                                                                   ;SETUP UNIT NUBER
11121                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11122
11123                                   ;SET UP FOR A ELEVEN WORD READ FROM AN ODD WORD BOUNDARY
11124  033026  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11125  033032  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
11126                                                                   ;FROM AN ODD WORD BOUNDARY
11127  033036  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11128  033042  005301                          DEC     R1              ;COUNT
11129  033044  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11130  033046  012777  003002  151010          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
11131                                                                   ;FROM AN ODD WORD BOUNDARY
11132  033054  012777  177765  151000          MOV #-11.,@RHWC ;WORD COUNT ELEVEN
11133  033062  004077  151552                  JSR     R0,@COMND       ;GO TO DO COMMAND
11134  033066  004160                          READAT                  ;READ DATA
11135
11136
11137                                   ;CHECK THAT RHCS3 HAS DBL
11138  033070  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
11139
11140  033076  017737  151030  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11141  033104  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11142                                                                   ;DATA WITH DATA READ FROM
11143                                                                   ;RHCS3
11144  033112  001401                          BEQ     65$             ;BRANCH IF GOOD
11145  033114  104124                          ERROR   124
11146                                                                   ;AFTER SETTING "CLR" BIT #5
11147                                                                   ;IN RHCS2 TO INIT THE RH
11148                                                                   ;A ELEVEN WORD READ FROM AN
11149                                                                   ;ODD WORD BOUNDARY WAS DONE
11150                                                                   ;THEN
11151                                                                   ;RHCS3 SHOULD HAVE DBL
11152                                                                   ;=2000
11153                                                                   ;BUT CONTAINED WHAT IS
11154                                                                   ;GIVEN IN BAD RHCS3
11155  033116                           65$:
11156  033116  042777  000076  150734          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11157                                   ;CHECK THAT RHCS1 HAS RDY!DVA
11158  033124  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
11159
11160  033132  017737  150722  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
```

```
11161  033140  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11162                                                                   ;DATA WITH DATA READ FROM
11163                                                                   ;RHCS1
11164  033146  001401                          BEQ     67$             ;BRANCH IF GOOD
11165  033150  104125                          ERROR   125
11166                                                                   ;AFTER SETTING "CLR" BIT #5
11167                                                                   ;IN RHCS2 TO INIT THE RH
11168                                                                   ;A ELEVEN WORD READ FROM AN
11169                                                                   ;ODD WORD BOUNDARY WAS DONE
11170                                                                   ;THEN
11171                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
11172                                                                   ;=4200
11173                                                                   ;BUT CONTAINED WHAT IS
11174                                                                   ;GIVEN IN BAD RHCS1
11175  033152                          67$:
11176                                          ;CHECK THAT RHCS2 HAS IR
11177  033152  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
11178  033160  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11179
11180  033166  017737  150676  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11181  033174  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11182                                                                   ;DATA WITH DATA READ FROM
11183                                                                   ;RHCS2
11184  033202  001401                          BEQ     69$             ;BRANCH IF GOOD
11185  033204  104126                          ERROR   126
11186                                                                   ;AFTER SETTING "CLR" BIT #5
11187                                                                   ;IN RHCS2 TO INIT THE RH
11188                                                                   ;A ELEVEN WORD READ FROM AN
11189                                                                   ;ODD WORD BOUNDARY WAS DONE
11190                                                                   ;THEN
11191                                                                   ;RHCS2 SHOULD HAVE IR
11192                                                                   ;=100
11193                                                                   ;TOGETHER WITH UNIT NUMBER
11194                                                                   ;BUT CONTAINED WHAT IS
11195                                                                   ;GIVEN IN BAD RHCS2
11196  033206                          69$:
11197                                          ;CHECK THAT RHWC HAS 0
11198  033206  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11199
11200  033214  017737  150642  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11201  033222  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11202                                                                   ;DATA WITH DATA READ FROM
11203                                                                   ;RHWC
11204  033230  001401                          BEQ     71$             ;BRANCH IF GOOD
11205  033232  104131                          ERROR   131
11206                                                                   ;AFTER SETTING "CLR" BIT #5
11207                                                                   ;IN RHCS2 TO INIT THE RH
11208                                                                   ;A ELEVEN WORD READ FROM AN
11209                                                                   ;ODD WORD BOUNDARY WAS DONE
11210                                                                   ;THEN
11211                                                                   ;RHWC SHOULD HAVE 0
11212                                                                   ;BUT CONTAINED WHAT IS
11213                                                                   ;GIVEN IN BAD RHWC
11214  033234                          71$:
11215
11216
```

```
11217
11218                                   ;;*****************************************************************
11219                                   ;*TEST 53        TEST DBL (RHCS3 BIT #10) TEST L
11220
11221                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11222                                   ;*      SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
11223                                   ;*      WITH BAI IN RHCS2 BIT #3 SET
11224                                   ;*      DO A TEN WORD READ FROM AN EVEN WORD BOUNDARY
11225                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
11226                                   ;*      CHECK RHCS3
11227                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11228
11229                                   ;;*****************************************************************
11230  033234  000004                  TST53:  SCOPE
11231  033236  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11232  033242  012737  000053  004656          MOV     #53,@#TSTNM     ;SAVE TEST NUMBER
11233
11234  033250  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11235                                                                  ;SETUP UNIT NUBER
11236                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11237
11238                                   ;SET UP FOR A TEN WORD READ FROM AN EVEN WORD BOUNDARY
11239  033254  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11240  033260  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11241                                                                  ;FROM AN EVEN WORD BOUNDARY
11242  033264  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11243  033270  005301                          DEC     R1              ;COUNT
11244  033272  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11245  033274  012777  003000  150562          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
11246                                                                  ;FROM AN EVEN WORD BOUNDARY
11247  033302  012777  177766  150552          MOV #-10.,@RHWC ;WORD COUNT TEN
11248  033310  052777  000010  150552  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
11249  033316  004077  151316                  JSR     R0,@COMND       ;GO TO DO COMMAND
11250  033322  004160                          READAT                  ;READ DATA
11251
11252
11253                                   ;CHECK THAT RHCS3 HAS 0
11254  033324  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11255
11256  033332  017737  150574  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11257  033340  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11258                                                                  ;DATA WITH DATA READ FROM
11259                                                                  ;RHCS3
11260  033346  001401                          BEQ     65$             ;BRANCH IF GOOD
11261  033350  104124                          ERROR   124
11262                                                                  ;AFTER SETTING "CLR" BIT #5
11263                                                                  ;IN RHCS2 TO INIT THE RH
11264                                                                  ;A TEN WORD READ FROM AN
11265                                                                  ;EVEN WORD BOUNDARY WAS DONE
11266                                                                  ;WITH BAI IN RHCS2 SET
11267                                                                  ;THEN
11268                                                                  ;RHCS3 SHOULD HAVE 0
11269                                                                  ;BUT CONTAINED WHAT IS
11270                                                                  ;GIVEN IN BAD RHCS3
11271  033352                  65$:
11272  033352  042777  000076  150500          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
```

```
11273                                                   ;CHECK THAT RHCS1 HAS RDY!DVA
11274  033360  012737  004200  001124                   MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
11275
11276  033366  017737  150466  001126                   MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11277  033374  023737  001124  001126                   CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11278                                                                           ;DATA WITH DATA READ FROM
11279                                                                           ;RHCS1
11280  033402  001401                                   BEQ     67$             ;BRANCH IF GOOD
11281  033404  104125                                   ERROR   125
11282                                                                           ;AFTER SETTING "CLR" BIT #5
11283                                                                           ;IN RHCS2 TO INIT THE RH
11284                                                                           ;A TEN WORD READ FROM AN
11285                                                                           ;EVEN WORD BOUNDARY WAS DONE
11286                                                                           ;WITH BAI IN RHCS2 SET
11287                                                                           ;THEN
11288                                                                           ;RHCS1 SHOULD HAVE RDY!DVA
11289                                                                           ;=4200
11290                                                                           ;BUT CONTAINED WHAT IS
11291                                                                           ;GIVEN IN BAD RHCS1
11292  033406                                   67$:
11293                                                   ;CHECK THAT RHCS2 HAS IR!BAI
11294  033406  012737  000110  001124                   MOV     #IR!BAI,$GDDAT  ;GET GOOD = 110
11295  033414  053737  005010  001124                   BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11296
11297  033422  017737  150442  001126                   MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11298  033430  023737  001124  001126                   CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11299                                                                           ;DATA WITH DATA READ FROM
11300                                                                           ;RHCS2
11301  033436  001401                                   BEQ     69$             ;BRANCH IF GOOD
11302  033440  104126                                   ERROR   126
11303                                                                           ;AFTER SETTING "CLR" BIT #5
11304                                                                           ;IN RHCS2 TO INIT THE RH
11305                                                                           ;A TEN WORD READ FROM AN
11306                                                                           ;EVEN WORD BOUNDARY WAS DONE
11307                                                                           ;WITH BAI IN RHCS2 SET
11308                                                                           ;THEN
11309                                                                           ;RHCS2 SHOULD HAVE IR!BAI
11310                                                                           ;=110
11311                                                                           ;TOGETHER WITH UNIT NUMBER
11312                                                                           ;BUT CONTAINED WHAT IS
11313                                                                           ;GIVEN IN BAD RHCS2
11314  033442                                   69$:
11315                                                   ;CHECK THAT RHWC HAS 0
11316  033442  012737  000000  001124                   MOV     #0,$GDDAT       ;GET GOOD = 0
11317
11318  033450  017737  150406  001126                   MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11319  033456  023737  001124  001126                   CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11320                                                                           ;DATA WITH DATA READ FROM
11321                                                                           ;RHWC
11322  033464  001401                                   BEQ     71$             ;BRANCH IF GOOD
11323  033466  104131                                   ERROR   131
11324                                                                           ;AFTER SETTING "CLR" BIT #5
11325                                                                           ;IN RHCS2 TO INIT THE RH
11326                                                                           ;A TEN WORD READ FROM AN
11327                                                                           ;EVEN WORD BOUNDARY WAS DONE
11328                                                                           ;WITH BAI IN RHCS2 SET
```

```
                                                                ;THEN
                                                                ;RHWC SHOULD HAVE 0
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHWC
033470                                  71$:
                                        ;;**************************************************************
                                        ;*TEST 54       TEST DBL (RHCS3 BIT #10) TEST M
                                        ;*
                                        ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
                                        ;*      SET UP FOR A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                                        ;*      DO A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
                                        ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
                                        ;*      CHECK RHCS3
                                        ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
                                        ;;**************************************************************
033470  000004                          TST54:  SCOPE
033472  012706  001000                          MOV     #STACK,SP       ;RESET STACK
033476  012737  000054  004656                  MOV     #54,@#TSTNM     ;SAVE TEST NUMBER
033504  004737  040306                          JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
                                                                        ;SETUP UNIT NUBER
                                                                        ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
                                        ;SET UP FOR A ONE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
033510  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
033514  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
                                                                ;FROM AN EVEN WORD BOUNDARY
033520  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
033524  005301                  DEC     R1              ;COUNT
033526  001374                  BNE     1$              ;BRANCH IF THREE NOT DONE
033530  012777  003000  150326          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
                                                                ;FROM AN EVEN WORD BOUNDARY
033536  012777  177777  150316  MOV #-1,@RHWC ;WORD COUNT ONE
033544  004077  151070          JSR     R0,@COMND       ;GO TO DO COMMAND
033550  004204                  REVWRT                  ;REVERSEWRITE DATA
                                ;CHECK THAT RHCS3 HAS 0
033552  012737  000000  001124  MOV     #0,$GDDAT       ;GET GOOD = 0
033560  017737  150346  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
033566  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                        ;DATA WITH DATA READ FROM
                                                        ;RHCS3
033574  001401                  BEQ     65$             ;BRANCH IF GOOD
033576  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
```

```
11385                                                           ;RHCS3 SHOULD HAVE 0
11386                                                           ;BUT CONTAINED WHAT IS
11387                                                           ;GIVEN IN BAD RHCS3
11388  033600                                  65$:
11389  033600  042777  000076  150252          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11390                                          ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11391  033606  012737  104200  001124          MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11392
11393  033614  017737  150240  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11394  033622  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11395                                                          ;DATA WITH DATA READ FROM
11396                                                          ;RHCS1
11397  033630  001401                          BEQ     67$             ;BRANCH IF GOOD
11398  033632  104125                          ERROR   125
11399                                                          ;AFTER SETTING "CLR" BIT #5
11400                                                          ;IN RHCS2 TO INIT THE RH
11401                                                          ;A ONE WORD WRITE REVERSE FROM AN
11402                                                          ;EVEN WORD BOUNDARY WAS DONE
11403                                                          ;THEN
11404                                                          ;RHCS1 SHOULD HAVE SC!RDY!DVA
11405                                                          ;=104200
11406                                                          ;BUT CONTAINED WHAT IS
11407                                                          ;GIVEN IN BAD RHCS1
11408  033634                                  67$:
11409                                          ;CHECK THAT RHCS2 HAS IR!OR
11410  033634  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11411  033642  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11412
11413  033650  017737  150214  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11414  033656  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11415                                                          ;DATA WITH DATA READ FROM
11416                                                          ;RHCS2
11417  033664  001401                          BEQ     69$             ;BRANCH IF GOOD
11418  033666  104126                          ERROR   126
11419                                                          ;AFTER SETTING "CLR" BIT #5
11420                                                          ;IN RHCS2 TO INIT THE RH
11421                                                          ;A ONE WORD WRITE REVERSE FROM AN
11422                                                          ;EVEN WORD BOUNDARY WAS DONE
11423                                                          ;THEN
11424                                                          ;RHCS2 SHOULD HAVE IR!OR
11425                                                          ;=300
11426                                                          ;TOGETHER WITH UNIT NUMBER
11427                                                          ;BUT CONTAINED WHAT IS
11428                                                          ;GIVEN IN BAD RHCS2
11429  033670                                  69$:
11430                                          ;CHECK THAT RHWC HAS 0
11431  033670  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11432
11433  033676  017737  150160  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11434  033704  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11435                                                          ;DATA WITH DATA READ FROM
11436                                                          ;RHWC
11437  033712  001401                          BEQ     71$             ;BRANCH IF GOOD
11438  033714  104131                          ERROR   131
11439                                                          ;AFTER SETTING "CLR" BIT #5
11440                                                          ;IN RHCS2 TO INIT THE RH
```

```
11441                                                                   ;A ONE WORD WRITE REVERSE FROM AN
11442                                                                   ;EVEN WORD BOUNDARY WAS DONE
11443                                                                   ;THEN
11444                                                                   ;RHWC SHOULD HAVE 0
11445                                                                   ;BUT CONTAINED WHAT IS
11446                                                                   ;GIVEN IN BAD RHWC
11447  033716                                 71$:
11448
11449
11450
11451                                 ;;*****************************************************************
11452                                 ;*TEST 55        TEST DBL (RHCS3 BIT #10) TEST N
11453
11454                                 ;*       CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11455                                 ;*       SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11456                                 ;*       DO A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11457                                 ;*       THIS SHOULD NOT SET RHCS3 BIT #10 DBL
11458                                 ;*       CHECK RHCS3
11459                                 ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11460
11461                                 ;;*****************************************************************
11462  033716  000004                 TST55:  SCOPE
11463  033720  012706  001000                 MOV     #STACK,SP       ;RESET STACK
11464  033724  012737  000055  004656         MOV     #55,@#TSTNM     ;SAVE TEST NUMBER
11465
11466  033732  004737  040306                 JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11467                                                                  ;SETUP UNIT NUBER
11468                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11469
11470                                         ;SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11471  033736  012701  000003                 MOV     #3,R1           ;COUNT OF THREE
11472  033742  012702  003000                 MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11473                                                                  ;FROM AN EVEN WORD BOUNDARY
11474  033746  012722  052525         1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11475  033752  005301                         DEC     R1              ;COUNT
11476  033754  001374                         BNE     1$              ;BRANCH IF THREE NOT DONE
11477  033756  012777  003000  150100         MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
11478                                                                  ;FROM AN EVEN WORD BOUNDARY
11479  033764  012777  177776  150070         MOV #-2,@RHWC ;WORD COUNT TWO
11480  033772  004077  150642                 JSR     R0,@COMND       ;GO TO DO COMMAND
11481  033776  004204                         REVWRT                  ;REVERSEWRITE DATA
11482
11483
11484                                         ;CHECK THAT RHCS3 HAS 0
11485  034000  012737  000000  001124         MOV     #0,$GDDAT       ;GET GOOD = 0
11486
11487  034006  017737  150120  001126         MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11488  034014  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11489                                                                  ;DATA WITH DATA READ FROM
11490                                                                  ;RHCS3
11491  034022  001401                         BEQ     65$             ;BRANCH IF GOOD
11492  034024  104124                         ERROR   124
11493                                                                  ;AFTER SETTING "CLR" BIT #5
11494                                                                  ;IN RHCS2 TO INIT THE RH
11495                                                                  ;A TWO WORD WRITE REVERSE FROM AN
11496                                                                  ;EVEN WORD BOUNDARY WAS DONE
```

```
11497                                                                   ;THEN
11498                                                                   ;RHCS3 SHOULD HAVE 0
11499                                                                   ;BUT CONTAINED WHAT IS
11500                                                                   ;GIVEN IN BAD RHCS3
11501   034026                                  65$:
11502   034026  042777  000076  150024            BIC     #76,@RHCS1    ;CLEAR FUNCTION BITS
11503                                           ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11504   034034  012737  104200  001124            MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11505
11506   034042  017737  150012  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11507   034050  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11508                                                                   ;DATA WITH DATA READ FROM
11509                                                                   ;RHCS1
11510   034056  001401                            BEQ     67$           ;BRANCH IF GOOD
11511   034060  104125                            ERROR   125
11512                                                                   ;AFTER SETTING "CLR" BIT #5
11513                                                                   ;IN RHCS2 TO INIT THE RH
11514                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11515                                                                   ;EVEN WORD BOUNDARY WAS DONE
11516                                                                   ;THEN
11517                                                                   ;RHCS1 SHOULD HAVE SC!RDY!DVA
11518                                                                   ;=104200
11519                                                                   ;BUT CONTAINED WHAT IS
11520                                                                   ;GIVEN IN BAD RHCS1
11521   034062                                  67$:
11522                                           ;CHECK THAT RHCS2 HAS IR!OR
11523   034062  012737  000300  001124            MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11524   034070  053737  005010  001124            BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11525
11526   034076  017737  147766  001126            MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11527   034104  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11528                                                                   ;DATA WITH DATA READ FROM
11529                                                                   ;RHCS2
11530   034112  001401                            BEQ     69$           ;BRANCH IF GOOD
11531   034114  104126                            ERROR   126
11532                                                                   ;AFTER SETTING "CLR" BIT #5
11533                                                                   ;IN RHCS2 TO INIT THE RH
11534                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11535                                                                   ;EVEN WORD BOUNDARY WAS DONE
11536                                                                   ;THEN
11537                                                                   ;RHCS2 SHOULD HAVE IR!OR
11538                                                                   ;=300
11539                                                                   ;TOGETHER WITH UNIT NUMBER
11540                                                                   ;BUT CONTAINED WHAT IS
11541                                                                   ;GIVEN IN BAD RHCS2
11542   034116                                  69$:
11543                                           ;CHECK THAT RHWC HAS 0
11544   034116  012737  000000  001124            MOV     #0,$GDDAT       ;GET GOOD = 0
11545
11546   034124  017737  147732  001126            MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11547   034132  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11548                                                                   ;DATA WITH DATA READ FROM
11549                                                                   ;RHWC
11550   034140  001401                            BEQ     71$           ;BRANCH IF GOOD
11551   034142  104131                            ERROR   131
11552                                                                   ;AFTER SETTING "CLR" BIT #5
```

```
11553                                                                   ;IN RHCS2 TO INIT THE RH
11554                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11555                                                                   ;EVEN WORD BOUNDARY WAS DONE
11556                                                                   ;THEN
11557                                                                   ;RHWC SHOULD HAVE 0
11558                                                                   ;BUT CONTAINED WHAT IS
11559                                                                   ;GIVEN IN BAD RHWC
11560  034144                                  71$:
11561
11562
11563
11564                                  ;;*****************************************************************
11565                                  ;*TEST 56        TEST DBL (RHCS3 BIT #10) TEST 0
11566
11567                                  ;*       CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11568                                  ;*       SET UP FOR A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11569                                  ;*       DO A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11570                                  ;*       THIS SHOULD SET RHCS3 BIT #10 DBL
11571                                  ;*       CHECK RHCS3
11572                                  ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11573
11574                                  ;;*****************************************************************
11575  034144  000004                  ↑ST56:  SCOPE
11576  034146  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11577  034152  012737  000056  004656          MOV     #56,@#↑STNM     ;SAVE TEST NUMBER
11578
11579  034160  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11580                                                                   ;SETUP UNIT NUBER
11581                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11582
11583                                          ;SET UP FOR A TWO WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11584  034164  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11585  034170  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
11586                                                                   ;FROM AN ODD WORD BOUNDARY
11587  034174  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11588  034200  005301                          DEC     R1              ;COUNT
11589  034202  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11590  034204  012777  003002  147652          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
11591                                                                   ;FROM AN ODD WORD BOUNDARY
11592  034212  012777  177776  147642          MOV #-2,@RHWC ;WORD COUNT TWO
11593  034220  004077  150414                  JSR     R0,@COMND       ;GO TO DO COMMAND
11594  034224  004204                          REVWRT                  ;REVERSEWRITE DATA
11595
11596
11597                                          ;CHECK THAT RHCS3 HAS DBL
11598  034226  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
11599
11600  034234  017737  147672  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11601  034242  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11602                                                                   ;DATA WITH DATA READ FROM
11603                                                                   ;RHCS3
11604  034250  001401                          BEQ     65$             ;BRANCH IF GOOD
11605  034252  104124                          ERROR   124
11606                                                                   ;AFTER SETTING "CLR" BIT #5
11607                                                                   ;IN RHCS2 TO INIT THE RH
11608                                                                   ;A TWO WORD WRITE REVERSE FROM AN
```

```
                                                                ;ODD WORD BOUNDARY WAS DONE
                                                                ;THEN
                                                                ;RHCS3 SHOULD HAVE DBL
                                                                ;=2000
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS3
034254                                          65$:
034254  042777  000076  147576                  BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
                                                ;CHECK THAT RHCS1 HAS SC!RDY!DVA
034262  012737  104200  001124                  MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200

034270  017737  147564  001126                  MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
034276  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS1
034304  001401                                  BEQ     67$             ;BRANCH IF GOOD
034306  104125                                  ERROR   125
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;A TWO WORD WRITE REVERSE FROM AN
                                                                ;ODD WORD BOUNDARY WAS DONE
                                                                ;THEN
                                                                ;RHCS1 SHOULD HAVE SC!RDY!DVA
                                                                ;=104200
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS1
034310                                          67$:
                                                ;CHECK THAT RHCS2 HAS IR!OR
034310  012737  000300  001124                  MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
034316  053737  005010  001124                  BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER

034324  017737  147540  001126                  MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
034332  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHCS2
034340  001401                                  BEQ     69$             ;BRANCH IF GOOD
034342  104126                                  ERROR   126
                                                                ;AFTER SETTING "CLR" BIT #5
                                                                ;IN RHCS2 TO INIT THE RH
                                                                ;A TWO WORD WRITE REVERSE FROM AN
                                                                ;ODD WORD BOUNDARY WAS DONE
                                                                ;THEN
                                                                ;RHCS2 SHOULD HAVE IR!OR
                                                                ;=300
                                                                ;TOGETHER WITH UNIT NUMBER
                                                                ;BUT CONTAINED WHAT IS
                                                                ;GIVEN IN BAD RHCS2
034344                                          69$:
                                                ;CHECK THAT RHWC HAS 0
034344  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0

034352  017737  147504  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
034360  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
                                                                ;DATA WITH DATA READ FROM
                                                                ;RHWC
034366  001401                                  BEQ     71$             ;BRANCH IF GOOD
```

```
11665  034370  104131                          ERROR   131
11666                                                       ;AFTER SETTING "CLR" BIT #5
11667                                                       ;IN RHCS2 TO INIT THE RH
11668                                                       ;A TWO WORD WRITE REVERSE FROM AN
11669                                                       ;ODD WORD BOUNDARY WAS DONE
11670                                                       ;THEN
11671                                                       ;RHWC SHOULD HAVE 0
11672                                                       ;BUT CONTAINED WHAT IS
11673                                                       ;GIVEN IN BAD RHWC
11674  034372                          71$:
11675
11676
11677
11678                                   ;;*****************************************************************
11679                                   ;*TEST 57       TEST DBL (RHCS3 BIT #10) TEST P
11680
11681                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11682                                   ;*      SET UP FOR A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11683                                   ;*      DO A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11684                                   ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
11685                                   ;*      CHECK RHCS3
11686                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11687
11688                                   ;;*****************************************************************
11689  034372  000004                  T$T57:  SCOPE
11690  034374  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11691  034400  012737  000057  004656          MOV     #57,@#T$TNM     ;SAVE TEST NUMBER
11692
11693  034406  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11694                                                                  ;SETUP UNIT NUBER
11695                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11696
11697                                   ;SET UP FOR A THREE WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11698  034412  012701  000003          MOV     #3,R1           ;COUNT OF THREE
11699  034416  012702  003000          MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11700                                                          ;FROM AN EVEN WORD BOUNDARY
11701  034422  012722  052525  1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11702  034426  005301                  DEC     R1              ;COUNT
11703  034430  001374                  BNE     1$              ;BRANCH IF THREE NOT DONE
11704  034432  012777  003000  147424  MOV     #BFEVE .,@RHBA  ;SET BUS ADDRESS TO START
11705                                                          ;FROM AN EVEN WORD BOUNDARY
11706  034440  012777  177775  147414  MOV #-3,@RHWC ;WORD COUNT THREE
11707  034446  004077  150166          JSR     R0,@COMND       ;GO TO DO COMMAND
11708  034452  004204                  REVWRT                  ;REVERSEWRITE DATA
11709
11710
11711                                   ;CHECK THAT RHCS3 HAS DBL
11712  034454  012737  002000  001124  MOV     #DBL,$GDDAT     ;GET GOOD = 2000
11713
11714  034462  017737  147444  001126  MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11715  034470  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11716                                                          ;DATA WITH DATA READ FROM
11717                                                          ;RHCS3
11718  034476  001401                  BEQ     65$             ;BRANCH IF GOOD
11719  034500  104124                  ERROR   124
11720                                                          ;AFTER SETTING "CLR" BIT #5
```

```
11721                                                           ;IN RHCS2 TO INIT THE RH
11722                                                           ;A THREE WORD WRITE REVERSE FROM AN
11723                                                           ;EVEN WORD BOUNDARY WAS DONE
11724                                                           ;THEN
11725                                                           ;RHCS3 SHOULD HAVE DBL
11726                                                           ;=2000
11727                                                           ;BUT CONTAINED WHAT IS
11728                                                           ;GIVEN IN BAD RHCS3
11729  034502                                65$:
11730  034502  042777  000076  147350          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
11731                                        ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11732  034510  012737  104200  001124          MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11733
11734  034516  017737  147336  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
11735  034524  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11736                                                          ;DATA WITH DATA READ FROM
11737                                                          ;RHCS1
11738  034532  001401                          BEQ     67$             ;BRANCH IF GOOD
11739  034534  104125                          ERROR   125
11740                                                          ;AFTER SETTING "CLR" BIT #5
11741                                                          ;IN RHCS2 TO INIT THE RH
11742                                                          ;A THREE WORD WRITE REVERSE FROM AN
11743                                                          ;EVEN WORD BOUNDARY WAS DONE
11744                                                          ;THEN
11745                                                          ;RHCS1 SHOULD HAVE SC!RDY!DVA
11746                                                          ;=104200
11747                                                          ;BUT CONTAINED WHAT IS
11748                                                          ;GIVEN IN BAD RHCS1
11749  034536                                67$:
11750                                        ;CHECK THAT RHCS2 HAS IR!OR
11751  034536  012737  000300  001124          MOV     #IR!OR,$GDDAT   ;GET GOOD = 300
11752  034544  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
11753
11754  034552  017737  147312  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
11755  034560  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11756                                                          ;DATA WITH DATA READ FROM
11757                                                          ;RHCS2
11758  034566  001401                          BEQ     69$             ;BRANCH IF GOOD
11759  034570  104126                          ERROR   126
11760                                                          ;AFTER SETTING "CLR" BIT #5
11761                                                          ;IN RHCS2 TO INIT THE RH
11762                                                          ;A THREE WORD WRITE REVERSE FROM AN
11763                                                          ;EVEN WORD BOUNDARY WAS DONE
11764                                                          ;THEN
11765                                                          ;RHCS2 SHOULD HAVE IR!OR
11766                                                          ;=300
11767                                                          ;TOGETHER WITH UNIT NUMBER
11768                                                          ;BUT CONTAINED WHAT IS
11769                                                          ;GIVEN IN BAD RHCS2
11770  034572                                69$:
11771                                        ;CHECK THAT RHWC HAS 0
11772  034572  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11773
11774  034600  017737  147256  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
11775  034606  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11776                                                          ;DATA WITH DATA READ FROM
```

```
11777
11778  034614  001401                          BEQ     71$             ;RHWC
11779  034616  104131                          ERROR   131             ;BRANCH IF GOOD
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                                                                   ;GIVEN IN BAD RHWC
11788  034620                          71$:
11789
11790
11791
11792                                  ;;*************************************************************
11793                                  ;*TEST 60        TEST DBL (RHCS3 BIT #10) TEST Q
11794
11795                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11796                                  ;*      SET UP FOR A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11797                                  ;*      DO A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11798                                  ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
11799                                  ;*      CHECK RHCS3
11800                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11801
11802                                  ;;*************************************************************
11803  034620  000004                  TST60:  SCOPE
11804  034622  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11805  034626  012737  000060  004656          MOV     #60,@#TSTNM     ;SAVE TEST NUMBER
11806
11807  034634  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11808                                                                   ;SETUP UNIT NUBER
11809                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11810
11811                                  ;SET UP FOR A THREE WORD WRITE REVERSE FROM AN ODD WORD BOUNDARY
11812  034640  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11813  034644  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
11814                                                                   ;FROM AN ODD WORD BOUNDARY
11815  034650  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11816  034654  005301                          DEC     R1              ;COUNT
11817  034656  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11818  034660  012777  003002  147176          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
11819                                                                   ;FROM AN ODD WORD BOUNDARY
11820  034666  012777  177775  147166  MOV #-3,@RHWC ;WORD COUNT THREE
11821  034674  004077  147740          JSR     R0,@COMND               ;GO TO DO COMMAND
11822  034700  004204                  REVWRT                          ;REVERSEWRITE DATA
11823
11824
11825                                  ;CHECK THAT RHCS3 HAS 0
11826  034702  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11827
11828  034710  017737  147216  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11829  034716  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
11830                                                                   ;DATA WITH DATA READ FROM
11831                                                                   ;RHCS3
11832  034724  001401                          BEQ     65$             ;BRANCH IF GOOD
```

```
11833  034726  104124                          ERROR   124
11834                                                       ;AFTER SETTING "CLR" BIT #5
11835                                                       ;IN RHCS2 TO INIT THE RH
11836                                                       ;A THREE WORD WRITE REVERSE FROM AN
11837                                                       ;ODD WORD BOUNDARY WAS DONE
11838                                                       ;THEN
11839                                                       ;RHCS3 SHOULD HAVE 0
11840                                                       ;BUT CONTAINED WHAT IS
11841                                                       ;GIVEN IN BAD RHCS3
11842  034730                                65$:
11843  034730  042777  000076  147122        BIC     #76,@RHCS1          ;CLEAR FUNCTION BITS
11844                                        ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11845  034736  012737  104200  001124        MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11846
11847  034744  017737  147110  001126        MOV     @RHCS1,$BDDAT       ;READ RHCS1 FOR COMPARISON
11848  034752  023737  001124  001126        CMP     $GDDAT,$BDDAT       ;COMPARE EXPECTED
11849                                                       ;DATA WITH DATA READ FROM
11850                                                       ;RHCS1
11851  034760  001401                        BEQ     67$                 ;BRANCH IF GOOD
11852  034762  104125                        ERROR   125
11853                                                       ;AFTER SETTING "CLR" BIT #5
11854                                                       ;IN RHCS2 TO INIT THE RH
11855                                                       ;A THREE WORD WRITE REVERSE FROM AN
11856                                                       ;ODD WORD BOUNDARY WAS DONE
11857                                                       ;THEN
11858                                                       ;RHCS1 SHOULD HAVE SC!RDY!DVA
11859                                                       ;=104200
11860                                                       ;BUT CONTAINED WHAT IS
11861                                                       ;GIVEN IN BAD RHCS1
11862  034764                                67$:
11863                                        ;CHECK THAT RHCS2 HAS IR!OR
11864  034764  012737  000300  001124        MOV     #IR!OR,$GDDAT       ;GET GOOD = 300
11865  034772  053737  005010  001124        BIS     UNIT,$GDDAT         ;INCLUDE UNIT NUMBER
11866
11867  035000  017737  147064  001126        MOV     @RHCS2,$BDDAT       ;READ RHCS2 FOR COMPARISON
11868  035006  023737  001124  001126        CMP     $GDDAT,$BDDAT       ;COMPARE EXPECTED
11869                                                       ;DATA WITH DATA READ FROM
11870                                                       ;RHCS2
11871  035014  001401                        BEQ     69$                 ;BRANCH IF GOOD
11872  035016  104126                        ERROR   126
11873                                                       ;AFTER SETTING "CLR" BIT #5
11874                                                       ;IN RHCS2 TO INIT THE RH
11875                                                       ;A THREE WORD WRITE REVERSE FROM AN
11876                                                       ;ODD WORD BOUNDARY WAS DONE
11877                                                       ;THEN
11878                                                       ;RHCS2 SHOULD HAVE IR!OR
11879                                                       ;=300
11880                                                       ;TOGETHER WITH UNIT NUMBER
11881                                                       ;BUT CONTAINED WHAT IS
11882                                                       ;GIVEN IN BAD RHCS2
11883  035020                                69$:
11884                                        ;CHECK THAT RHWC HAS 0
11885  035020  012737  000000  001124        MOV     #0,$GDDAT           ;GET GOOD = 0
11886
11887  035026  017737  147030  001126        MOV     @RHWC,$BDDAT        ;READ RHWC FOR COMPARISON
11888  035034  023737  001124  001126        CMP     $GDDAT,$BDDAT       ;COMPARE EXPECTED
```

```
11889                                                                   ;DATA WITH DATA READ FROM
11890                                                                   ;RHWC
11891   035042  001401                        BEQ     71$             ;BRANCH IF GOOD
11892   035044  104131                        ERROR   131
11893                                                                   ;AFTER SETTING "CLR" BIT #5
11894                                                                   ;IN RHCS2 TO INIT THE RH
11895                                                                   ;A THREE WORD WRITE REVERSE FROM AN
11896                                                                   ;ODD WORD BOUNDARY WAS DONE
11897                                                                   ;THEN
11898                                                                   ;RHWC SHOULD HAVE 0
11899                                                                   ;BUT CONTAINED WHAT IS
11900                                                                   ;GIVEN IN BAD RHWC
11901   035046                          71$:
11902
11903
11904
11905                                   ;;*****************************************************************
11906                                   ;*TEST 61        TEST DBL (RHCS3 BIT #10) TEST R
11907
11908                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
11909                                   ;*      SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11910                                   ;*      WITH BAI IN RHCS2 BIT #3 SET
11911                                   ;*      DO A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11912                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
11913                                   ;*      CHECK RHCS3
11914                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
11915
11916                                   ;;*****************************************************************
11917   035046  000004                  T%T61:  SCOPE
11918   035050  012706  001000                  MOV     #STACK,SP       ;RESET STACK
11919   035054  012737  000061  004656          MOV     #61,@#T%TSTNM   ;SAVE TEST NUMBER
11920
11921   035062  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
11922                                                                   ;SETUP UNIT NUBER
11923                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
11924
11925                                           ;SET UP FOR A TWO WORD WRITE REVERSE FROM AN EVEN WORD BOUNDARY
11926   035066  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
11927   035072  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
11928                                                                   ;FROM AN EVEN WORD BOUNDARY
11929   035076  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
11930   035102  005301                          DEC     R1              ;COUNT
11931   035104  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
11932   035106  012777  003000  146750          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
11933                                                                   ;FROM AN EVEN WORD BOUNDARY
11934   035114  012777  177776  146740          MOV #-2,@RHWC ;WORD COUNT TWO
11935   035122  052777  000010  146740  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
11936   035130  004077  147504                  JSR     R0,@COMND       ;GO TO DO COMMAND
11937   035134  004204                          REVWRT                  ;REVERSEWRITE DATA
11938
11939
11940                                           ;CHECK THAT RHCS3 HAS 0
11941   035136  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
11942
11943   035144  017737  146762  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
11944   035152  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
```

```
11945                                                                   ;DATA WITH DATA READ FROM
11946                                                                   ;RHCS3
11947  035160  001401                        BEQ     65$                ;BRANCH IF GOOD
11948  035162  104124                        ERROR   124
11949                                                                   ;AFTER SETTING "CLR" BIT #5
11950                                                                   ;IN RHCS2 TO INIT THE RH
11951                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11952                                                                   ;EVEN WORD BOUNDARY WAS DONE
11953                                                                   ;WITH BAI IN RHCS2 SET
11954                                                                   ;THEN
11955                                                                   ;RHCS3 SHOULD HAVE 0
11956                                                                   ;BUT CONTAINED WHAT IS
11957                                                                   ;GIVEN IN BAD RHCS3
11958  035164                           65$:
11959  035164  042777  000076  146666        BIC     #76,@RHCS1         ;CLEAR FUNCTION BITS
11960                                        ;CHECK THAT RHCS1 HAS SC!RDY!DVA
11961  035172  012737  104200  001124        MOV     #SC!RDY!DVA,$GDDAT      ;GET GOOD = 104200
11962
11963  035200  017737  146654  001126        MOV     @RHCS1,$BDDAT      ;READ RHCS1 FOR COMPARISON
11964  035206  023737  001124  001126        CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
11965                                                                   ;DATA WITH DATA READ FROM
11966                                                                   ;RHCS1
11967  035214  001401                        BEQ     67$                ;BRANCH IF GOOD
11968  035216  104125                        ERROR   125
11969                                                                   ;AFTER SETTING "CLR" BIT #5
11970                                                                   ;IN RHCS2 TO INIT THE RH
11971                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11972                                                                   ;EVEN WORD BOUNDARY WAS DONE
11973                                                                   ;WITH BAI IN RHCS2 SET
11974                                                                   ;THEN
11975                                                                   ;RHCS1 SHOULD HAVE SC!RDY!DVA
11976                                                                   ;=104200
11977                                                                   ;BUT CONTAINED WHAT IS
11978                                                                   ;GIVEN IN BAD RHCS1
11979  035220                           67$:
11980                                        ;CHECK THAT RHCS2 HAS IR!OR!BAI
11981  035220  012737  000310  001124        MOV     #IR!OR!BAI,$GDDAT       ;GET GOOD = 310
11982  035226  053737  005010  001124        BIS     UNIT,$GDDAT        ;INCLUDE UNIT NUMBER
11983
11984  035234  017737  146630  001126        MOV     @RHCS2,$BDDAT      ;READ RHCS2 FOR COMPARISON
11985  035242  023737  001124  001126        CMP     $GDDAT,$BDDAT      ;COMPARE EXPECTED
11986                                                                   ;DATA WITH DATA READ FROM
11987                                                                   ;RHCS2
11988  035250  001401                        BEQ     69$                ;BRANCH IF GOOD
11989  035252  104126                        ERROR   126
11990                                                                   ;AFTER SETTING "CLR" BIT #5
11991                                                                   ;IN RHCS2 TO INIT THE RH
11992                                                                   ;A TWO WORD WRITE REVERSE FROM AN
11993                                                                   ;EVEN WORD BOUNDARY WAS DONE
11994                                                                   ;WITH BAI IN RHCS2 SET
11995                                                                   ;THEN
11996                                                                   ;RHCS2 SHOULD HAVE IR!OR!BAI
11997                                                                   ;=310
11998                                                                   ;TOGETHER WITH UNIT NUMBER
11999                                                                   ;BUT CONTAINED WHAT IS
12000                                                                   ;GIVEN IN BAD RHCS2
```

```
                                        T61     TEST DBL (RHCS3 BIT #10) TEST R                                 SEQ 0224

 12001  035254                                 69$:
 12002                                                  ;CHECK THAT RHWC HAS 0
 12003  035254  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
 12004
 12005  035262  017737  146574  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
 12006  035270  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
 12007                                                                          ;DATA WITH DATA READ FROM
 12008                                                                          ;RHWC
 12009  035276  001401                                  BEQ     71$             ;BRANCH IF GOOD
 12010  035300  104131                                  ERROR   131
 12011                                                                          ;AFTER SETTING "CLR" BIT #5
 12012                                                                          ;IN RHCS2 TO INIT THE RH
 12013                                                                          ;A TWO WORD WRITE REVERSE FROM AN
 12014                                                                          ;EVEN WORD BOUNDARY WAS DONE
 12015                                                                          ;WITH BAI IN RHCS2 SET
 12016                                                                          ;THEN
 12017                                                                          ;RHWC SHOULD HAVE 0
 12018                                                                          ;BUT CONTAINED WHAT IS
 12019                                                                          ;GIVEN IN BAD RHWC
 12020  035302                                 71$:
 12021
 12022
 12023
 12024
 12025                                  ;;*****************************************************************
 12026                                  ;*TEST 62        TEST DBL (RHCS3 BIT #10) TEST S
 12027
 12028                                  ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
 12029                                  ;*      SET UP FOR A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
 12030                                  ;*      DO A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
 12031                                  ;*      THIS SHOULD SET RHCS3 BIT #10 DBL
 12032                                  ;*      CHECK RHCS3
 12033                                  ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
 12034                                  ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
 12035
 12036                                  ;;*****************************************************************
 12037  035302  000004                  TST62:  SCOPE
 12038  035304  012706  001000                  MOV     #STACK,SP       ;RESET STACK
 12039  035310  012737  000062  004656          MOV     #62,@#TSTNM     ;SAVE TEST NUMBER
 12040
 12041  035316  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
 12042                                                                  ;SETUP UNIT NUBER
 12043                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
 12044  035322  022737  041674  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
 12045
 12046  035330  001103                          BNE     TST63   ;BRANCH IF TU NOT THERE
 12047
 12048
 12049                                          ;SET UP FOR A NINE WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
 12050  035332  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
 12051  035336  012702  003000                  MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
 12052                                                                  ;FROM AN EVEN WORD BOUNDARY
 12053  035342  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
 12054  035346  005301                          DEC     R1              ;COUNT
 12055  035350  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
 12056  035352  012777  003000  146504          MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
```

```
12057                                                   ;FROM AN EVEN WORD BOUNDARY
12058  035360  012777  177767  146474          MOV #-9.,@RHWC ;WORD COUNT NINE
12059
12060  035366  004077  147246                  JSR     R0,@COMND       ;GO TO DO COMMAND
12061  035372  004200                          REVRED                  ;REVERSE READ
12062
12063                                           ;CHECK THAT RHCS3 HAS DBL
12064  035374  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
12065
12066  035402  017737  146524  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12067  035410  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12068                                                                  ;DATA WITH DATA READ FROM
12069                                                                  ;RHCS3
12070  035416  001401                          BEQ     65$             ;BRANCH IF GOOD
12071  035420  104124                          ERROR   124
12072                                                                  ;AFTER SETTING "CLR" BIT #5
12073                                                                  ;IN RHCS2 TO INIT THE RH
12074                                                                  ;A NINE WORD READ REVERSE FROM AN
12075                                                                  ;EVEN WORD BOUNDARY WAS DONE
12076                                                                  ;THEN
12077                                                                  ;RHCS3 SHOULD HAVE DBL
12078                                                                  ;=2000
12079                                                                  ;BUT CONTAINED WHAT IS
12080                                                                  ;GIVEN IN BAD RHCS3
12081  035422                          65$:
12082  035422  042777  000076  146430          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12083                                           ;CHECK THAT RHCS1 HAS RDY!DVA
12084  035430  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12085
12086  035436  017737  146416  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12087  035444  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12088                                                                  ;DATA WITH DATA READ FROM
12089                                                                  ;RHCS1
12090  035452  001401                          BEQ     67$             ;BRANCH IF GOOD
12091  035454  104125                          ERROR   125
12092                                                                  ;AFTER SETTING "CLR" BIT #5
12093                                                                  ;IN RHCS2 TO INIT THE RH
12094                                                                  ;A NINE WORD READ REVERSE FROM AN
12095                                                                  ;EVEN WORD BOUNDARY WAS DONE
12096                                                                  ;THEN
12097                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
12098                                                                  ;=4200
12099                                                                  ;BUT CONTAINED WHAT IS
12100                                                                  ;GIVEN IN BAD RHCS1
12101  035456                          67$:
12102                                           ;CHECK THAT RHCS2 HAS IR
12103  035456  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12104  035464  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12105
12106  035472  017737  146372  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12107  035500  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12108                                                                  ;DATA WITH DATA READ FROM
12109                                                                  ;RHCS2
12110  035506  001401                          BEQ     69$             ;BRANCH IF GOOD
12111  035510  104126                          ERROR   126
12112                                                                  ;AFTER SETTING "CLR" BIT #5
```

```
12113                                                                   ;IN RHCS2 TO INIT THE RH
12114                                                                   ;A NINE WORD READ REVERSE FROM AN
12115                                                                   ;EVEN WORD BOUNDARY WAS DONE
12116                                                                   ;THEN
12117                                                                   ;RHCS2 SHOULD HAVE IR
12118                                                                   ;=100
12119                                                                   ;TOGETHER WITH UNIT NUMBER
12120                                                                   ;BUT CONTAINED WHAT IS
12121                                                                   ;GIVEN IN BAD RHCS2
12122  035512                                 69$:
12123                                                  ;CHECK THAT RHWC HAS 0
12124  035512  012737  000000  001124                  MOV     #0,$GDDAT       ;GET GOOD = 0
12125
12126  035520  017737  146336  001126                  MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12127  035526  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12128                                                                   ;DATA WITH DATA READ FROM
12129                                                                   ;RHWC
12130  035534  001401                                  BEQ     71$             ;BRANCH IF GOOD
12131  035536  104131                                  ERROR   131
12132                                                                   ;AFTER SETTING "CLR" BIT #5
12133                                                                   ;IN RHCS2 TO INIT THE RH
12134                                                                   ;A NINE WORD READ REVERSE FROM AN
12135                                                                   ;EVEN WORD BOUNDARY WAS DONE
12136                                                                   ;THEN
12137                                                                   ;RHWC SHOULD HAVE 0
12138                                                                   ;BUT CONTAINED WHAT IS
12139                                                                   ;GIVEN IN BAD RHWC
12140  035540                                 71$:
12141
12142
12143
12144                                          ;;*************************************************************
12145                                          ;*TEST 63       TEST DBL (RHCS3 BIT #10) TEST T
12146
12147                                          ;*       CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12148                                          ;*       SET UP FOR A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12149                                          ;*       DO A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12150                                          ;*       THIS SHOULD NOT SET RHCS3 BIT #10 DBL
12151                                          ;*       CHECK RHCS3
12152                                          ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12153                                          ;*       IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12154
12155                                          ;;*************************************************************
12156  035540  000004                         TST63:  SCOPE
12157  035542  012706  001000                         MOV     #STACK,SP       ;RESET STACK
12158  035546  012737  000063  004656                 MOV     #63,@#TSTNM     ;SAVE TEST NUMBER
12159
12160  035554  004737  004306                         JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12161                                                                   ;SETUP UNIT NUBER
12162                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12163  035560  022737  041674  004640                 CMP     #CMNDTM,COMND   ;IS TU THERE
12164
12165  035566  001103                                 BNE     TST64   ;BRANCH IF TU NOT THERE
12166
12167
12168                                                  ;SET UP FOR A TEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
```

```
12168  035570  012701  000003                 MOV     #3,R1           ;COUNT OF THREE
12170  035574  012702  003000                 MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
12171                                                                  ;FROM AN EVEN WORD BOUNDARY
12172  035600  012722  052525          1$:    MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12173  035604  005301                 DEC     R1              ;COUNT
12174  035606  001374                 BNE     1$              ;BRANCH IF THREE NOT DONE
12175  035610  012777  003000  146246 MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
12176                                                                  ;FROM AN EVEN WORD BOUNDARY
12177  035616  012777  177766  146236 MOV #-10.,@RHWC ;WORD COUNT TEN
12178
12179  035624  004077  147010         JSR     R0,@COMND       ;GO TO DO COMMAND
12180  035630  004200                 REVRED                  ;REVERSE READ
12181
12182                                 ;CHECK THAT RHCS3 HAS 0
12183  035632  012737  000000  001124 MOV     #0,$GDDAT       ;GET GOOD = 0
12184
12185  035640  017737  146266  001126 MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12186  035646  023737  001124  001126 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12187                                                                  ;DATA WITH DATA READ FROM
12188                                                                  ;RHCS3
12189  035654  001401                 BEQ     65$             ;BRANCH IF GOOD
12190  035656  104124                 ERROR   124
12191                                                                  ;AFTER SETTING "CLR" BIT #5
12192                                                                  ;IN RHCS2 TO INIT THE RH
12193                                                                  ;A TEN WORD READ REVERSE FROM AN
12194                                                                  ;EVEN WORD BOUNDARY WAS DONE
12195                                                                  ;THEN
12196                                                                  ;RHCS3 SHOULD HAVE 0
12197                                                                  ;BUT CONTAINED WHAT IS
12198                                                                  ;GIVEN IN BAD RHCS3
12199  035660                         65$:
12200  035660  042777  000076  146172 BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12201                                 ;CHECK THAT RHCS1 HAS RDY!DVA
12202  035666  012737  004200  001124 MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12203
12204  035674  017737  146160  001126 MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12205  035702  023737  001124  001126 CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12206                                                                  ;DATA WITH DATA READ FROM
12207                                                                  ;RHCS1
12208  035710  001401                 BEQ     67$             ;BRANCH IF GOOD
12209  035712  104125                 ERROR   125
12210                                                                  ;AFTER SETTING "CLR" BIT #5
12211                                                                  ;IN RHCS2 TO INIT THE RH
12212                                                                  ;A TEN WORD READ REVERSE FROM AN
12213                                                                  ;EVEN WORD BOUNDARY WAS DONE
12214                                                                  ;THEN
12215                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
12216                                                                  ;=4200
12217                                                                  ;BUT CONTAINED WHAT IS
12218                                                                  ;GIVEN IN BAD RHCS1
12219  035714                         67$:
12220                                 ;CHECK THAT RHCS2 HAS IR
12221  035714  012737  000100  001124 MOV     #IR,$GDDAT      ;GET GOOD = 100
12222  035722  053737  005010  001124 BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12223
12224  035730  017737  146134  001126 MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
```

```
12225  035736  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12226                                                                   ;DATA WITH DATA READ FROM
12227                                                                   ;RHCS2
12228  035744  001401                          BEQ     69$             ;BRANCH IF GOOD
12229  035746  104126                          ERROR   126
12230                                                                   ;AFTER SETTING "CLR" BIT #5
12231                                                                   ;IN RHCS2 TO INIT THE RH
12232                                                                   ;A TEN WORD READ REVERSE FROM AN
12233                                                                   ;EVEN WORD BOUNDARY WAS DONE
12234                                                                   ;THEN
12235                                                                   ;RHCS2 SHOULD HAVE IR
12236                                                                   ;=100
12237                                                                   ;TOGETHER WITH UNIT NUMBER
12238                                                                   ;BUT CONTAINED WHAT IS
12239                                                                   ;GIVEN IN BAD RHCS2
12240  035750                          69$:
12241                                          ;CHECK THAT RHWC HAS 0
12242  035750  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12243
12244  035756  017737  146100  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12245  035764  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12246                                                                   ;DATA WITH DATA READ FROM
12247                                                                   ;RHWC
12248  035772  001401                          BEQ     71$             ;BRANCH IF GOOD
12249  035774  104131                          ERROR   131
12250                                                                   ;AFTER SETTING "CLR" BIT #5
12251                                                                   ;IN RHCS2 TO INIT THE RH
12252                                                                   ;A TEN WORD READ REVERSE FROM AN
12253                                                                   ;EVEN WORD BOUNDARY WAS DONE
12254                                                                   ;THEN
12255                                                                   ;RHWC SHOULD HAVE 0
12256                                                                   ;BUT CONTAINED WHAT IS
12257                                                                   ;GIVEN IN BAD RHWC
12258  035776                          71$:
12259
12260
12261
12262                                  ;;*****************************************************************
12263                                  ;*TEST 64        TEST DBL (RHCS3 BIT #10) TEST U
12264
12265                                  ;*       CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12266                                  ;*       SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12267                                  ;*       DO A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12268                                  ;*       THIS SHOULD SET RHCS3 BIT #10 DBL
12269                                  ;*       CHECK RHCS3
12270                                  ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12271                                  ;*       IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12272
12273                                  ;;*****************************************************************
12274  035776  000004                  ↑ST64:  SCOPE
12275  036000  012706  001000                  MOV     #STACK,SP       ;RESET STACK
12276  036004  012737  000064  004656          MOV     #64,@#↑STNM     ;SAVE TEST NUMBER
12277
12278  036012  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12279                                                                   ;SETUP UNIT NUBER
12280                                                                   ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
```

```
12281  036016  022737  041674  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
12282
12283  036024  001103                          BNE     TST65   ;BRANCH IF TU NOT THERE
12284
12285
12286                                          ;SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12287  036026  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12288  036032  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
12289                                                                  ;FROM AN ODD WORD BOUNDARY
12290  036036  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12291  036042  005301                          DEC     R1              ;COUNT
12292  036044  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12293  036046  012777  003002  146010          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
12294                                                                  ;FROM AN ODD WORD BOUNDARY
12295  036054  012777  177766  146000          MOV #-10.,@RHWC ;WORD COUNT TEN
12296
12297  036062  004077  146552                  JSR     R0,@COMND       ;GO TO DO COMMAND
12298  036066  004200                          REVRED                  ;REVERSE READ
12299
12300                                          ;CHECK THAT RHCS3 HAS DBL
12301  036070  012737  002000  001124          MOV     #DBL,$GDDAT     ;GET GOOD = 2000
12302
12303  036076  017737  146030  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12304  036104  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12305                                                                  ;DATA WITH DATA READ FROM
12306                                                                  ;RHCS3
12307  036112  001401                          BEQ     65$             ;BRANCH IF GOOD
12308  036114  104124                          ERROR   124
12309                                                                  ;AFTER SETTING "CLR" BIT #5
12310                                                                  ;IN RHCS2 TO INIT THE RH
12311                                                                  ;A TEN WORD READ REVERSE FROM AN
12312                                                                  ;ODD WORD BOUNDARY WAS DONE
12313                                                                  ;THEN
12314                                                                  ;RHCS3 SHOULD HAVE DBL
12315                                                                  ;=2000
12316                                                                  ;BUT CONTAINED WHAT IS
12317                                                                  ;GIVEN IN BAD RHCS3
12318  036116                          65$:
12319  036116  042777  000076  145734          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12320                                          ;CHECK THAT RHCS1 HAS RDY!DVA
12321  036124  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12322
12323  036132  017737  145722  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12324  036140  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12325                                                                  ;DATA WITH DATA READ FROM
12326                                                                  ;RHCS1
12327  036146  001401                          BEQ     67$             ;BRANCH IF GOOD
12328  036150  104125                          ERROR   125
12329                                                                  ;AFTER SETTING "CLR" BIT #5
12330                                                                  ;IN RHCS2 TO INIT THE RH
12331                                                                  ;A TEN WORD READ REVERSE FROM AN
12332                                                                  ;ODD WORD BOUNDARY WAS DONE
12333                                                                  ;THEN
12334                                                                  ;RHCS1 SHOULD HAVE RDY!DVA
12335                                                                  ;=4200
12336                                                                  ;BUT CONTAINED WHAT IS
```

```
12337                                                                   ;GIVEN IN BAD RHCS1
12338  036152                                  67$:
12339                                          ;CHECK THAT RHCS2 HAS IR
12340  036152  012737  000100  001124          MOV     #IR,$GDDAT       ;GET GOOD = 100
12341  036160  053737  005010  001124          BIS     UNIT,$GDDAT      ;INCLUDE UNIT NUMBER
12342
12343  036166  017737  145676  001126          MOV     @RHCS2,$BDDAT    ;READ RHCS2 FOR COMPARISON
12344  036174  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;COMPARE EXPECTED
12345                                                                   ;DATA WITH DATA READ FROM
12346                                                                   ;RHCS2
12347  036202  001401                          BEQ     69$              ;BRANCH IF GOOD
12348  036204  104126                          ERROR   126
12349                                                                   ;AFTER SETTING "CLR" BIT #5
12350                                                                   ;IN RHCS2 TO INIT THE RH
12351                                                                   ;A TEN WORD READ REVERSE FROM AN
12352                                                                   ;ODD WORD BOUNDARY WAS DONE
12353                                                                   ;THEN
12354                                                                   ;RHCS2 SHOULD HAVE IR
12355                                                                   ;=100
12356                                                                   ;TOGETHER WITH UNIT NUMBER
12357                                                                   ;BUT CONTAINED WHAT IS
12358                                                                   ;GIVEN IN BAD RHCS2
12359  036206                                  69$:
12360                                          ;CHECK THAT RHWC HAS 0
12361  036206  012737  000000  001124          MOV     #0,$GDDAT        ;GET GOOD = 0
12362
12363  036214  017737  145642  001126          MOV     @RHWC,$BDDAT     ;READ RHWC FOR COMPARISON
12364  036222  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;COMPARE EXPECTED
12365                                                                   ;DATA WITH DATA READ FROM
12366                                                                   ;RHWC
12367  036230  001401                          BEQ     71$              ;BRANCH IF GOOD
12368  036232  104131                          ERROR   131
12369                                                                   ;AFTER SETTING "CLR" BIT #5
12370                                                                   ;IN RHCS2 TO INIT THE RH
12371                                                                   ;A TEN WORD READ REVERSE FROM AN
12372                                                                   ;ODD WORD BOUNDARY WAS DONE
12373                                                                   ;THEN
12374                                                                   ;RHWC SHOULD HAVE 0
12375                                                                   ;BUT CONTAINED WHAT IS
12376                                                                   ;GIVEN IN BAD RHWC
12377  036234                                  71$:
12378
12379
12380
12381                                  ;;*****************************************************************
12382                                  ;*TEST 65       TEST DBL (RHCS3 BIT #10) TEST V
12383
12384                                  ;*       CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12385                                  ;*       SET UP FOR A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12386                                  ;*       DO A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12387                                  ;*       THIS SHOULD SET RHCS3 BIT #10 DBL
12388                                  ;*       CHECK RHCS3
12389                                  ;*       CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12390                                  ;*       IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12391
12392                                  ;;*****************************************************************
```

```
12393  036234  000004                    TST65:  SCOPE
12394  036236  012706  001000                    MOV     #STACK,SP       ;RESET STACK
12395  036242  012737  000065  004656            MOV     #65,@#TSTNM     ;SAVE TEST NUMBER
12396
12397  036250  004737  040306                    JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12398                                                                    ;SETUP UNIT NUBER
12399                                                                    ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12400  036254  022737  041674  004640            CMP     #CMNDTM,COMND   ;IS TU THERE
12401
12402  036262  001103                            BNE     TST66   ;BRANCH IF TU NOT THERE
12403
12404
12405                                            ;SET UP FOR A ELEVEN WORD READ REVERSE FROM AN EVEN WORD BOUNDARY
12406  036264  012701  000003                    MOV     #3,R1           ;COUNT OF THREE
12407  036270  012702  003000                    MOV     #BFEVEN,R2      ;START THREE WORD BUFFER
12408                                                                    ;FROM AN EVEN WORD BOUNDARY
12409  036274  012722  052525            1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12410  036300  005301                            DEC     R1              ;COUNT
12411  036302  001374                            BNE     1$              ;BRANCH IF THREE NOT DONE
12412  036304  012777  003000  145552            MOV     #BFEVEN,@RHBA   ;SET BUS ADDRESS TO START
12413                                                                    ;FROM AN EVEN WORD BOUNDARY
12414  036312  012777  177765  145542            MOV #-11.,@RHWC ;WORD COUNT ELEVEN
12415
12416  036320  004077  146314                    JSR     R0,@COMND       ;GO TO DO COMMAND
12417  036324  004200                            REVRED                  ;REVERSE READ
12418
12419                                            ;CHECK THAT RHCS3 HAS DBL
12420  036326  012737  002000  001124            MOV     #DBL,$GDDAT     ;GET GOOD = 2000
12421
12422  036334  017737  145572  001126            MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12423  036342  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12424                                                                    ;DATA WITH DATA READ FROM
12425                                                                    ;RHCS3
12426  036350  001401                            BEQ     65$             ;BRANCH IF GOOD
12427  036352  104124                            ERROR   124
12428                                                                    ;AFTER SETTING "CLR" BIT #5
12429                                                                    ;IN RHCS2 TO INIT THE RH
12430                                                                    ;A ELEVEN WORD READ REVERSE FROM AN
12431                                                                    ;EVEN WORD BOUNDARY WAS DONE
12432                                                                    ;THEN
12433                                                                    ;RHCS3 SHOULD HAVE DBL
12434                                                                    ;=2000
12435                                                                    ;BUT CONTAINED WHAT IS
12436                                                                    ;GIVEN IN BAD RHCS3
12437  036354                            65$:
12438  036354  042777  000076  145476            BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12439                                            ;CHECK THAT RHCS1 HAS RDY!DVA
12440  036362  012737  004200  001124            MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12441
12442  036370  017737  145464  001126            MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12443  036376  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12444                                                                    ;DATA WITH DATA READ FROM
12445                                                                    ;RHCS1
12446  036404  001401                            BEQ     67$             ;BRANCH IF GOOD
12447  036406  104125                            ERROR   125
12448                                                                    ;AFTER SETTING "CLR" BIT #5
```

```
12449                                                           ;IN RHCS2 TO INIT THE RH
12450                                                           ;A ELEVEN WORD READ REVERSE FROM AN
12451                                                           ;EVEN WORD BOUNDARY WAS DONE
12452                                                           ;THEN
12453                                                           ;RHCS1 SHOULD HAVE RDY!DVA
12454                                                           ;=4200
12455                                                           ;BUT CONTAINED WHAT IS
12456                                                           ;GIVEN IN BAD RHCS1
12457  036410                           67$:
12458                                   ;CHECK THAT RHCS2 HAS IR
12459  036410  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12460  036416  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12461
12462  036424  017737  145440  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12463  036432  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12464                                                           ;DATA WITH DATA READ FRO
12465                                                           ;RHCS2
12466  036440  001401                          BEQ     69$             ;BRANCH IF GOOD
12467  036442  104126                          ERROR   126
12468                                                           ;AFTER SETTING "CLR" BIT #5
12469                                                           ;IN RHCS2 TO INIT THE RH
12470                                                           ;A ELEVEN WORD READ REVERSE FROM AN
12471                                                           ;EVEN WORD BOUNDARY WAS DONE
12472                                                           ;THEN
12473                                                           ;RHCS2 SHOULD HAVE IR
12474                                                           ;=100
12475                                                           ;TOGETHER WITH UNIT NUMBER
12476                                                           ;BUT CONTAINED WHAT IS
12477                                                           ;GIVEN IN BAD RHCS2
12478  036444                           69$:
12479                                   ;CHECK THAT RHWC HAS 0
12480  036444  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12481
12482  036452  017737  145404  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12483  036460  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12484                                                           ;DATA WITH DATA READ FROM
12485                                                           ;RHWC
12486  036466  001401                          BEQ     71$             ;BRANCH IF GOOD
12487  036470  104131                          ERROR   131
12488                                                           ;AFTER SETTING "CLR" BIT #5
12489                                                           ;IN RHCS2 TO INIT THE RH
12490                                                           ;A ELEVEN WORD READ REVERSE FROM AN
12491                                                           ;EVEN WORD BOUNDARY WAS DONE
12492                                                           ;THEN
12493                                                           ;RHWC SHOULD HAVE 0
12494                                                           ;BUT CONTAINED WHAT IS
12495                                                           ;GIVEN IN BAD RHWC
12496  036472                           71$:
12497
12498
12499
12500                                   ;;*****************************************************************
12501                                   ;*TEST 66        TEST DBL (RHCS3 BIT #10) TEST W
12502
12503                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12504                                   ;*      SET UP FOR A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
```

```
12505                                   ;*      DO A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12506                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
12507                                   ;*      CHECK RHCS3
12508                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12509                                   ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12510
12511                                   ;;*****************************************************************
12512  036472  000004                  T$T66:  SCOPE
12513  036474  012706  001000                  MOV     #STACK,SP       ;RESET STACK
12514  036500  012737  000066  004656          MOV     #66,@#T$TNM     ;SAVE TEST NUMBER
12515
12516  036506  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12517                                                                  ;SETUP UNIT NUBER
12518                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12519  036512  022737  041674  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
12520
12521  036520  001103                          BNE     TST67   ;BRANCH IF TU NOT THERE
12522
12523
12524                                          ;SET UP FOR A ELEVEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12525  036522  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12526  036526  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
12527                                                                  ;FROM AN ODD WORD BOUNDARY
12528  036532  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12529  036536  005301                          DEC     R1              ;COUNT
12530  036540  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12531  036542  012777  003002  145314          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
12532                                                                  ;FROM AN ODD WORD BOUNDARY
12533  036550  012777  177765  145304          MOV #-11.,@RHWC ;WORD COUNT ELEVEN
12534
12535  036556  004077  146056                  JSR     R0,@COMND       ;GO TO DO COMMAND
12536  036562  004200                          REVRED                  ;REVERSE READ
12537
12538                                          ;CHECK THAT RHCS3 HAS 0
12539  036564  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12540
12541  036572  017737  145334  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12542  036600  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12543                                                                  ;DATA WITH DATA READ FROM
12544                                                                  ;RHCS3
12545  036606  001401                          BEQ     65$             ;BRANCH IF GOOD
12546  036610  104124                          ERROR   124
12547                                                                  ;AFTER SETTING "CLR" BIT #5
12548                                                                  ;IN RHCS2 TO INIT THE RH
12549                                                                  ;A ELEVEN WORD READ REVERSE FROM AN
12550                                                                  ;ODD WORD BOUNDARY WAS DONE
12551                                                                  ;THEN
12552                                                                  ;RHCS3 SHOULD HAVE 0
12553                                                                  ;BUT CONTAINED WHAT IS
12554                                                                  ;GIVEN IN BAD RHCS3
12555  036612                  65$:
12556  036612  042777  000076  145240          BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12557                                          ;CHECK THAT RHCS1 HAS RDY!DVA
12558  036620  012737  004200  001124          MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12559
12560  036626  017737  145226  001126          MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
```

```
12561  036634  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12562                                                                   ;DATA WITH DATA READ FROM
12563                                                                   ;RHCS1
12564  036642  001401                          BEQ     67$             ;BRANCH IF GOOD
12565  036644  104125                          ERROR   125
12566                                                                   ;AFTER SETTING "CLR" BIT #5
12567                                                                   ;IN RHCS2 TO INIT THE RH
12568                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12569                                                                   ;ODD WORD BOUNDARY WAS DONE
12570                                                                   ;THEN
12571                                                                   ;RHCS1 SHOULD HAVE RDY!DVA
12572                                                                   ;=4200
12573                                                                   ;BUT CONTAINED WHAT IS
12574                                                                   ;GIVEN IN BAD RHCS1
12575  036646                          67$:
12576                                          ;CHECK THAT RHCS2 HAS IR
12577  036646  012737  000100  001124          MOV     #IR,$GDDAT      ;GET GOOD = 100
12578  036654  053737  005010  001124          BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12579
12580  036662  017737  145202  001126          MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12581  036670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12582                                                                   ;DATA WITH DATA READ FROM
12583                                                                   ;RHCS2
12584  036676  001401                          BEQ     69$             ;BRANCH IF GOOD
12585  036700  104126                          ERROR   126
12586                                                                   ;AFTER SETTING "CLR" BIT #5
12587                                                                   ;IN RHCS2 TO INIT THE RH
12588                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12589                                                                   ;ODD WORD BOUNDARY WAS DONE
12590                                                                   ;THEN
12591                                                                   ;RHCS2 SHOULD HAVE IR
12592                                                                   ;=100
12593                                                                   ;TOGETHER WITH UNIT NUMBER
12594                                                                   ;BUT CONTAINED WHAT IS
12595                                                                   ;GIVEN IN BAD RHCS2
12596  036702                          69$:
12597                                          ;CHECK THAT RHWC HAS 0
12598  036702  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12599
12600  036710  017737  145146  001126          MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12601  036716  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12602                                                                   ;DATA WITH DATA READ FROM
12603                                                                   ;RHWC
12604  036724  001401                          BEQ     71$             ;BRANCH IF GOOD
12605  036726  104131                          ERROR   131
12606                                                                   ;AFTER SETTING "CLR" BIT #5
12607                                                                   ;IN RHCS2 TO INIT THE RH
12608                                                                   ;A ELEVEN WORD READ REVERSE FROM AN
12609                                                                   ;ODD WORD BOUNDARY WAS DONE
12610                                                                   ;THEN
12611                                                                   ;RHWC SHOULD HAVE 0
12612                                                                   ;BUT CONTAINED WHAT IS
12613                                                                   ;GIVEN IN BAD RHWC
12614  036730                          71$:
12615
12616
```

```
12617
12618                                   ;;*****************************************************************
12619                                   ;*TEST 67       TEST DBL (RHCS3 BIT #10) TEST X
12620
12621                                   ;*      CLEAR THE SUBSYSTEM BY A CONTROLLER CLEAR (RHCS2 BIT #5)
12622                                   ;*      SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12623                                   ;*      WITH BAI IN RHCS2 BIT #3 SET
12624                                   ;*      DO A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12625                                   ;*      THIS SHOULD NOT SET RHCS3 BIT #10 DBL
12626                                   ;*      CHECK RHCS3
12627                                   ;*      CHECK RHCS1,RHCS2,RHBA,RHBAE,RHWC
12628                                   ;*      IF MAG. TAPE NOT USED.THIS TEST IS NOT DONE
12629
12630                                   ;;*****************************************************************
12631  036730  000004                  TST67:  SCOPE
12632  036732  012706  001000                  MOV     #STACK,SP       ;RESET STACK
12633  036736  012737  000067  004656          MOV     #67,@#TSTNM     ;SAVE TEST NUMBER
12634
12635  036744  004737  040306                  JSR     PC,@#CLDISK     ;GIVE RH INITIALIZE
12636                                                                  ;SETUP UNIT NUBER
12637                                                                  ;CLEAR RHWC AND FUNCTION BITS IN RHCS1
12638  036750  022737  041674  004640          CMP     #CMNDTM,COMND   ;IS TU THERE
12639
12640  036756  001106                          BNE     TST70   ;BRANCH IF TU NOT THERE
12641
12642
12643                                           ;SET UP FOR A TEN WORD READ REVERSE FROM AN ODD WORD BOUNDARY
12644  036760  012701  000003                  MOV     #3,R1           ;COUNT OF THREE
12645  036764  012702  003002                  MOV     #BFODD,R2       ;START THREE WORD BUFFER
12646                                                                  ;FROM AN ODD WORD BOUNDARY
12647  036770  012722  052525          1$:     MOV     #52525,(R2)+    ;MOVE THREE 52525 INTO BUFFER
12648  036774  005301                          DEC     R1              ;COUNT
12649  036776  001374                          BNE     1$              ;BRANCH IF THREE NOT DONE
12650  037000  012777  003002  145056          MOV     #BFODD,@RHBA    ;SET BUS ADDRESS TO START
12651                                                                  ;FROM AN ODD WORD BOUNDARY
12652  037006  012777  177766  145046          MOV #-10.,@RHWC ;WORD COUNT TEN
12653  037014  052777  000010  145046  BIS #BAI,@RHCS2 ;SET BAI IN RHCS2
12654
12655  037022  004077  145612                  JSR     R0,@COMND       ;GO TO DO COMMAND
12656  037026  004200                          REVRED                  ;REVERSE READ
12657
12658                                           ;CHECK THAT RHCS3 HAS 0
12659  037030  012737  000000  001124          MOV     #0,$GDDAT       ;GET GOOD = 0
12660
12661  037036  017737  145070  001126          MOV     @RHCS3,$BDDAT   ;READ RHCS3 FOR COMPARISON
12662  037044  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12663                                                                  ;DATA WITH DATA READ FROM
12664                                                                  ;RHCS3
12665  037052  001401                          BEQ     65$             ;BRANCH IF GOOD
12666  037054  104124                          ERROR   124
12667                                                                  ;AFTER SETTING "CLR" BIT #5
12668                                                                  ;IN RHCS2 TO INIT THE RH
12669                                                                  ;A TEN WORD READ REVERSE FROM AN
12670                                                                  ;ODD WORD BOUNDARY WAS DONE
12671                                                                  ;WITH BAI IN RHCS2 SET
12672                                                                  ;THEN
```

```
12673                                                           ;RHCS3 SHOULD HAVE 0
12674                                                           ;BUT CONTAINED WHAT IS
12675                                                           ;GIVEN IN BAD RHCS3
12676  037056                                65$:
12677  037056  042777  000076  144774        BIC     #76,@RHCS1      ;CLEAR FUNCTION BITS
12678                                        ;CHECK THAT RHCS1 HAS RDY!DVA
12679  037064  012737  004200  001124        MOV     #RDY!DVA,$GDDAT ;GET GOOD = 4200
12680
12681  037072  017737  144762  001126        MOV     @RHCS1,$BDDAT   ;READ RHCS1 FOR COMPARISON
12682  037100  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12683                                                           ;DATA WITH DATA READ FROM
12684                                                           ;RHCS1
12685  037106  001401                        BEQ     67$             ;BRANCH IF GOOD
12686  037110  104125                        ERROR   125
12687                                                           ;AFTER SETTING "CLR" BIT #5
12688                                                           ;IN RHCS2 TO INIT THE RH
12689                                                           ;A TEN WORD READ REVERSE FROM AN
12690                                                           ;ODD WORD BOUNDARY WAS DONE
12691                                                           ;WITH BAI IN RHCS2 SET
12692                                                           ;THEN
12693                                                           ;RHCS1 SHOULD HAVE RDY!DVA
12694                                                           ;=4200
12695                                                           ;BUT CONTAINED WHAT IS
12696                                                           ;GIVEN IN BAD RHCS1
12697  037112                                67$:
12698                                        ;CHECK THAT RHCS2 HAS IR!BAI
12699  037112  012737  000110  001124        MOV     #IR!BAI,$GDDAT  ;GET GOOD = 110
12700  037120  053737  005010  001124        BIS     UNIT,$GDDAT     ;INCLUDE UNIT NUMBER
12701
12702  037126  017737  144736  001126        MOV     @RHCS2,$BDDAT   ;READ RHCS2 FOR COMPARISON
12703  037134  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12704                                                           ;DATA WITH DATA READ FROM
12705                                                           ;RHCS2
12706  037142  001401                        BEQ     69$             ;BRANCH IF GOOD
12707  037144  104126                        ERROR   126
12708                                                           ;AFTER SETTING "CLR" BIT #5
12709                                                           ;IN RHCS2 TO INIT THE RH
12710                                                           ;A TEN WORD READ REVERSE FROM AN
12711                                                           ;ODD WORD BOUNDARY WAS DONE
12712                                                           ;WITH BAI IN RHCS2 SET
12713                                                           ;THEN
12714                                                           ;RHCS2 SHOULD HAVE IR!BAI
12715                                                           ;=110
12716                                                           ;TOGETHER WITH UNIT NUMBER
12717                                                           ;BUT CONTAINED WHAT IS
12718                                                           ;GIVEN IN BAD RHCS2
12719  037146                                69$:
12720                                        ;CHECK THAT RHWC HAS 0
12721  037146  012737  000000  001124        MOV     #0,$GDDAT       ;GET GOOD = 0
12722
12723  037154  017737  144702  001126        MOV     @RHWC,$BDDAT    ;READ RHWC FOR COMPARISON
12724  037162  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
12725                                                           ;DATA WITH DATA READ FROM
12726                                                           ;RHWC
12727  037170  001401                        BEQ     71$             ;BRANCH IF GOOD
12728  037172  104131                        ERROR   131
```

```
12729                                                                   ;AFTER SETTING "CLR" BIT #5
12730                                                                   ;IN RHCS2 TO INIT THE RH
12731                                                                   ;A TEN WORD READ REVERSE FROM AN
12732                                                                   ;ODD WORD BOUNDARY WAS DONE
12733                                                                   ;WITH BAI IN RHCS2 SET
12734                                                                   ;THEN
12735                                                                   ;RHWC SHOULD HAVE 0
12736                                                                   ;BUT CONTAINED WHAT IS
12737                                                                   ;GIVEN IN BAD RHWC
12738  037174                           71$:
12739
12740
12741
12742
12743
12744
12745                                   ;;*********************************************************************
12746                                   ;*TEST 70       END OF ONE RH
12747                                   ;*       THIS IS THE END OF TEST FOR ONE RH
12748                                   ;*       IF THERE ARE MORE RH THEN THE PROGRAM
12749                                   ;*       JUMPS TO TEST 2 FOR NEXT RH TEST
12750                                   ;*       END PASS IS REACHED ONLY AFTER ALL RH ARE COMPLETE
12751                                   ;;*********************************************************************
12752  037174  000004                  T$T70:  SCOPE
12753  037176  012737  000001  001212          MOV     #1,$TIMES       ;;DO 1 ITERATION
12754  037204  005037  177776                  CLR     PS              ;REINSTATE PS TO 0
12755  037210  104401  037216                  TYPE    ,65$            ;;TYPE ASCIZ STRING
12756  037214  000414                          BR      64$             ;;GET OVER THE ASCIZ
12757                                   ;;65$:  .ASCIZ  <15><12>/END OF TEST FOR RH NO/
12758  037246                           64$:
12759  037246  013746  005510                  MOV     WORUNT,-(SP)    ;GET WORKING RH NUMBER LEFT
12760  037252  013702  005506                  MOV     TSTUNT,R2       ;GET TOTAL NO OF RH FOUND
12761  037256  162602                          SUB     (SP)+,R2        ;NO OF RH TESTED
12762  037260  006302                          ASL     R2              ;INDEX FOR RH TESTED
12763  037262  016246  005436                  MOV     FRHNM(R2),-(SP) ;TYPE OF RH NO.
12764  037266  104405                          TYPDS
12765  037270  104401  037276                  TYPE    ,67$            ;;TYPE ASCIZ STRING
12766  037274  000410                          BR      66$             ;;GET OVER THE ASCIZ
12767                                   ;;67$:  .ASCIZ  /           BASE /
12768  037316                           66$:
12769  037316  013746  004060                  MOV     RHCS1,-(SP)     ;TYPE BASE ADDRESS
12770  037322  104402                          TYPOC
12771  037324  104401  037332                  TYPE    ,69$            ;;TYPE ASCIZ STRING
12772  037330  000421                          BR      68$             ;;GET OVER THE ASCIZ
12773                                   ;;69$:  .ASCIZ  <15><12>/TOTAL NO OF ERRORS ON THIS PASS/
12774  037374                           68$:
12775  037374  013746  001112                  MOV     $ERTTL,-(SP)    ;TYPE TOTAL NO OF ERRORS
12776  037400  104405                          TYPDS
12777  037402  104401  037410                  TYPE    ,71$            ;;TYPE ASCIZ STRING
12778  037406  000402                          BR      70$             ;;GET OVER THE ASCIZ
12779                                   ;;71$:  .ASCIZ  <15><12>/ /
12780  037414                           70$:
12781  037414  104401  037422                  TYPE    ,73$            ;;TYPE ASCIZ STRING
12782  037420  000402                          BR      72$             ;;GET OVER THE ASCIZ
12783                                   ;;73$:  .ASCIZ  <15><12>/ /
12784  037426                           72$:
```

F03

END OF ONE RH


```
12785  037426  005037  001112                  CLR     $ERTTL          ;CLEAR TOTAL NO OF ERRORS
12786  037432  005337  005510                  DEC     WORUNT          ;DECREASE NO. OF RH TESTED
12787  037436  001402                          BEQ     1$
12788  037440  000137  010610                  JMP     TST2
12789  037444  013737  005506  005510  1$:     MOV     TSTUNT,WORUNT   ;MAKE WORKING RH SAME
12790                                                                  ;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

037452                                  $EOP:
037452  000004                                  SCOPE
037454  005037  001102                         CLR     $TSTNM          ;;ZERO THE TEST NUMBER
037460  005037  001212                         CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
037464  005237  001100                         INC     $PASS           ;;INCREMENT THE PASS NUMBER
037470  042737  100000  001100                BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
037476  005327                                  DEC     (PC)+           ;;LOOP?
037500  000001                          $EOPCT: .WORD   1
037502  003022                                  BGT     $DOAGN          ;;YES
037504  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
037506  000001                          $ENDCT: .WORD   1
037510  037500                                  $EOPCT
037512  104401  037557                         TYPE    ,$ENDMG         ;;TYPE "END PASS #"
037516  013746  001100                         MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
037522  104405                                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
037524  104401  037554                         TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
037530  013700  000042                 $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
037534  001405                                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
037536  000005                                  RESET                   ;;CLEAR THE WORLD
037540  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
037542  000240                                  NOP                     ;;SAVE ROOM
037544  000240                                  NOP                     ;;FOR
037546  000240                                  NOP                     ;;ACT11
037550                                  $DOAGN:
037550  000137                                  JMP     @(PC)+          ;;RETURN
037552  010610                          $RTNAD: .WORD   TST2
037554      377       377       000     $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
037557      015   042412    042116    $ENDMG: .ASCIZ  <15><12>/END PASS #/
037564  050040  051501    020123
037572  000043

                                        .SBTTL  SUBROUTINES

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

037574                                  TIEOUT:
037574  104401  037602                         TYPE    ,65$            ;;TYPE ASCIZ STRING
037600  000433                                  BR      64$             ;;GET OVER THE ASCIZ
                                        ;;65$:  .ASCIZ  <15><12>/TIMEOUT OCCURED THROUGH LOCATION 4 FROM PROGRAM PC/
```

```
12847   037670                          64$:
12848   037670  162716  000002                  SUB     #2,(SP)         ;TYPE PC
12849   037674  104402                          TYPOC
12850   037676  104401  037704                  TYPE    ,67$            ;;TYPE ASCIZ STRING
12851   037702  000410                          BR      66$             ;;GET OVER THE ASCIZ
12852                                   ;;67$:  .ASCIZ  <15><12>/PSW WAS    /
12853   037724                          66$:
12854   037724  104402                          TYPOC
12855   037726  000137  043104                  JMP     $SCOPE  ;RETURN TO TEST
12856
12857
12858
12859
12860
12861
12862
12863
12864                                   ;;*****************************************************************
12865                                   ;*      THIS ROUTINE HANDLES PARITY ERRORS
12866                                   ;;*****************************************************************
12867   037732                          PARITY:
12868   037732  104401  037740                  TYPE    ,65$            ;;TYPE ASCIZ STRING
12869   037736  000417                          BR      64$             ;;GET OVER THE ASCIZ
12870                                   ;;65$:  .ASCIZ  <15><12>/PARITY TRAP THRU VECTOR 114/
12871   037776                          64$:
12872   037776  104401  040004                  TYPE    ,67$            ;;TYPE ASCIZ STRING
12873   040002  000413                          BR      66$             ;;GET OVER THE ASCIZ
12874                                   ;;67$:  .ASCIZ  <15><12>/FROM PROGRAM PC    /
12875   040032                          66$:
12876   040032  162716  000002                  SUB     #2,(SP)                 ;TYPE PC
12877   040036  104402                          TYPOC
12878   040040  104401  040046                  TYPE    ,69$            ;;TYPE ASCIZ STRING
12879   040044  000407                          BR      68$             ;;GET OVER THE ASCIZ
12880                                   ;;69$:  .ASCIZ  <15><12>/PSW WAS    /
12881   040064                          68$:
12882   040064  104402                          TYPOC
12883   040066  104401  040074                  TYPE    ,71$            ;;TYPE ASCIZ STRING
12884   040072  000421                          BR      70$             ;;GET OVER THE ASCIZ
12885                                   ;;71$:  .ASCIZ  <15><1>/HIGH ERROR ADDRESS REGISTER    /
12886   040136                          70$:
12887   040136  017746  145354                  MOV     @HERADD,-(SP)           ;TYPE HIGH ERROR
12888   040142  104402                          TYPOC
```

```
12889  040144  104401  040152                  TYPE    ,73$            ;;TYPE ASCIZ STRING
12890  040150  000421                          BR      72$             ;;GET OVER THE ASCIZ
12891                                  ;;73$:  .ASCIZ  <15><12>/LOW ERROR ADDRESS REGISTER    /
12892  040214                          72$:
12893  040214  017746  145274                MOV     @LERADD,-(SP)           ;TYPE LOW ERROR
12894  040220  104402                          TYPOC
```

```
12895   040222  104401  040230                  TYPE    #75$            ;;TYPE ASCIZ STRING
12896   040226  000422                          BR      #74$            ;;GET OVER THE ASCIZ
12897                                   ;;75$:  .ASCIZ  <15><12>/MEMORY SYSTEM ERROR REGISTER/
12898   040274                          #74$:
12899   040274  017746  145220                  MOV     @MEMERR,-(SP)
12900   040300  104402                          TYPOC
12901   040302  000137  043104                  JMP     $SCOPE
12902
12903
12904
12905
12906
12907
12908
12909   040306  012777  000040  143554  CLDISK: MOV     #CLR,@RHCS2             ;CLEAR ALL REG.
12910   040314  013777  005010  143546          MOV     @#UNIT,@RHCS2   ;REINSTATE UNIT NO.
12911   040322  042777  000077  143530          BIC     #77,@RHCS1                      ;CLEAR FUNCTION BITS
12912   040330  005077  143526                  CLR     @RHWC                   ;CLEAR RHWC
12913   040334  000207                          RTS     PC
```

```
12914
12915
12916
12917
12918                                   ;THIS CHECKS DEVICE AVAILABLE (DVA) AND READY (RDY) IN RHCS1
12919                                   ;AND CHECKS MEDIUM ON LINE (MOL), DEVICE PRESENT (DPR), DEVICE READY (DRY) IN RHDS1
12920  040336  000000                  PCJSR:  0                       ;PC OF JSR
12921
12922
12923
12924
12925  040340  011637  040336          CHECK:  MOV     (SP),@#PCJSR                    ;SAVE PC OF JSR+4
12926  040344  162737  000004  040336          SUB     #4,@#PCJSR      ;GET PC OF JSR
12927  040352  011346                  MOV     @R3,-(SP)       ;GET RHDS1
12928  040354  052716  000100                  BIS     #VV,(SP)        ;DONT CHECK VV BIT
12929  040360  000406                  BR      CHECKC          ;GOTO COMMON CHECK ROUTIN
12930  040362  011637  040336          CHECKT: MOV     (SP),@#PCJSR                    ;SAVE PC OF JSR+4
12931  040366  162737  000004  040336          SUB     #4,@#PCJSR      ;GET PC OF JSR
12932  040374  011346                  MOV     @R3,-(SP)       ;GET RHDS1 & DO VV CHECK AT 3$
12933  040376  011146                  CHECKC: MOV     @R1,-(SP)       ;GET CS1
12934  040400  042716  173577                  BIC     #173577,(SP)                    ;CLEAR UNWANTED BITS
12935  040404  022726  004200                  CMP     #DVA!RDY,(SP)+                  ;RHCS1 SHOULD HAVE DEVICE AVAILABLE
12936                                                                  ;AND BE READY
12937  040410  001403                  BEQ     3$                      ;BRANCH IF GOOD
12938  040412  011137  001122                  MOV     @R1,@#$BDADR    ;BAD DATA REGISTER (RHCS1)
12939  040416  104062                  ERROR   62                      ;RHCS1 DID NOT HAVE DEVICE
12940                                                                  ;AVAILABLE RIGHT AT THE START
12941                                                                  ;ALL OTHER BITS SHOULD BE 0
12942  040420  042716  102000          3$:     BIC     #ATA!LBT,(SP)                   ;CLEAR UNWANTED BITS
12943  040424  022726  010700                  CMP     #MOL!DPR!DRY!VV,(SP)+;RHDS1 SHOULD HAVE THESE SET
12944  040430  001403                  BEQ     7$                      ;BRANCH IF GOOD
12945  040432  011337  001122                  MOV     @R3,@#$BDADR    ;BAD DATA IN  REGISTER (RHDS1)
12946  040436  104061                  ERROR   61                      ;RHDS1 HAS SOME BITS OTHER
12947                                                                  ;THAN MOL, DRY, DPR,VV SET
12948                                                                  ;ALL OTHER BITS SHOULD BE 0
12949  040440  000207                  7$:     RTS     PC                              ;RETURN TO TEST NO.
12950
```

```
 12951
 12952
 12953
 12954
 12955
 12956
 12957
 12958
 12959
 12960
 12961                                  ;THIS IS A WAIT LOOP WHEN NO P-CLOCK IS AVAILABLE
 12962                                  ;NO TIMING IS DONE
 12963                                  ;CALL IS
 12964                                  ;       WAT
 12965                                  ;       A       ;POINTER TO REGISTER ADDRESS
 12966                                  ;       B       ;BIT WAITED FOR
 12967                                  ;R3-IS A TEMPORARY COUNTER
 12968  040442  177777                 TIMCNT: 177777                  ;COUNT FOR WAIT LOOP
 12969
 12970  040444                         WAIT.T:
 12971  040444  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 12972  040446  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 12973  040450  016600  000004                 MOV     4(SP),R0        ;R0 HAS ADDRESS OF NEXT LOCATION
 12974  040454  010037  004666                 MOV     R0,@#WAITPC     ;WAT PC +2 IS IN WAITPC
 12975  040460  162737  000002  004666         SUB     #2,@#WAITPC     ;WAT PC IS IN WAITPC
 12976  040466  013037  004670                 MOV     @(R0)+,@#WAITRE  ;WAIT ON REGISTER ADDRESS
 12977  040472  012037  004672                 MOV     (R0)+,@#WAITBT  ;WAIT ON BIT
 12978  040476  010066  000004                 MOV     R0,4(SP)        ;RESTORE RETURN ON STACK
 12979
 12980                                          ;THIS HAS THE TWO COUNT DOWNS FROM 177777
 12981                                          ;FOR WAITING FOR A BIT TO SET
 12982  040502  005737  004646                 TST     WAT0            ;WAITING FOR 1?
 12983  040506  001025                         BNE     4$              ;BRANCH IF WAITING FOR 0
 12984  040510  013703  040442                 MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
 12985  040514  033777  004672  144146  1$:    BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
 12986  040522  001042                         BNE     3$              ;BRANCH IF YES
 12987  040524  005303                         DEC     R3              ;COUNT IF REQUIRED BIT NOT THERE
 12988  040526  001372                         BNE     1$
 12989  040530  013703  040442                 MOV     @#TIMCNT,R3     ;SECOND COUNT DOWN FROM 177777
 12990  040534  033777  004672  144126  2$:    BIT     @#WAITBT,@WAITRE ;IS REQUIRED BIT THERE
 12991  040542  001032                         BNE     3$              ;BRANCH IF YES
 12992  040544  005303                         DEC     R3              ;COUNT IF REQUIRED BIT NOOT THERE
 12993  040546  001372                         BNE     2$
 12994  040550  017737  144114  001126         MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
 12995  040556  104135                         ERROR   135             ;WAS WAITING FOR A BIT TO SET
 12996                                                                  ;BIT IN QUESTION IS IN "BIT WAITED"
 12997                                                                  ;REGISTER IN QUESTION IS IN "REG ADDR"
 12998  040560  000423                         BR      3$              ;BRANCH OUT
 12999
 13000
 13001
 13002                                          ;THIS HAS THE TWO COUNT DOWNS FROM 177777
 13003                                          ;FOR WAITING FOR A BIT TO RESET
 13004  040562  013703  040442          4$:    MOV     @#TIMCNT,R3     ;R3 HAS TEMPORARY COUNT
 13005  040566  012700  000020                 MOV     #20,R0          ;R0 HAS TEMPORARY COUNT
 13006  040572  033777  004672  144070  5$:    BIT     @#WAITBT,@WAITRE  ;IS REQUIRED BIT RESET?
```

```
13007   040600  001413                          BEQ     3$              ;BRANCH IF YES
13008   040602  005303                          DEC     R3              ;COUNT IF REQUIRED BIT HASN'T RESET YET
13009   040604  001372                          BNE     5$              ;TRY AGAIN
13010   040606  005300                          DEC     R0              ;DECREMENT R0 COUNT
13011   040610  001370                          BNE     5$              ;START OVER IF COUNT ISN'T ZERO YET
13012   040612  017737  144052  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
13013   040620  104136                          ERROR   136             ;WAS WAITING FOR A BIT TO RESET
13014                                                                   ;BIT IN QUESTION IS IN "BIT WAITED"
13015                                                                   ;REGISTER IN QUESTION IS IN "REG ADDR"
13016   040622  017737  144042  001126          MOV     @WAITRE,@#$BDDAT ;REGISTER CONTENTS FOR TYPEOUT
13017   040630                          3$:
13018   040630  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
13019   040632  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
13020   040634  000002                          RTI                     ;RETURN TO MAIN TEST
```

```
13021
13022
13023
13024                                   ;HERE IS A DETAILED EXPLAINATION OF HOW THE LOOP ON ERROR WORKS.
13025                                   ;ON HITTING AN ERROR IF THE LOOP ON ERROR SWITCH IS SET, THE
13026                                   ;PROGRAM GOES BACK - USUALLY BACK TO THE BEGINNING OF THE TEST.
13027
13028                                   ;WHEN THIS OPERATOR SELECTABLE SCOPE LOOP IS USED THEN THE POINT
13029                                   ;THE PROGRAM GOES BACK TO CAN BE CHANGED.
13030                                   ;THE RESTRICTIONS TO THE POINT WHERE THE PROGRAM CAN GO ARE: -
13031                                   ;1.  IT MUST BE WITHIN THE TEST UNDER CONSIDERATION
13032                                   ;2.  LOOP ON ERROR SWITCH MUST BE SET
13033                                   ;3.  THE ERROR MUST OCCUR WITHIN THE TEST UNDER CONSIDERATION
13034                                   ;IF THE ERROR DOES NOT OCCUR WITHIN THE TEST UNDER CONSIDERATION
13035                                   ;THE PROGRAM WILL REVERT TO NORMAL OPERATION.  HOWEVER, IF LOOP ON
13036                                   ;TEST SWITCH IS SET AND THIS OPERATOR SELECTABLE SCOPE LOOP IS USED
13037                                   ;THEN THE PROGRAM WILL LOOP BACK TO THE SELECTED POINT WHEN IT
13038                                   ;COMES TO THE END OF THE TEST UNDER CONSIDERATION.
13039                                   ;
13040                                   ;AFTER LOOPING FOR SOME TIME IF THE LOOP SWITCH IS PUT DOWN THEN
13041                                   ;NORMAL OPERATION WILL CONTINUE.
13042
13043  040636  000000                   TESTAD: 0                       ;FIRST ADDRESS OF TEST
13044  040640                          OPERSEL:
13045  040640  005037  177776                   CLR     PS              ;MAKE PROCESSOR STATUS ZERO
13046  040644  012737  177777  045064           MOV     #-1,@#PRITEM    ;CLEAR PREVIOUS ITEM NUMBER
13047  040652  104401  040660                   TYPE    ,65$            ;;TYPE ASCIZ STRING
13048  040656  000421                           BR      64$             ;;GET OVER THE ASCIZ
13049                                  ;;65$:   .ASCIZ  <15><12>/THE PROGRAM WAS IN TEST NUMBER /
13050  040722                          64$:
13051  040722  013746  004656                   MOV     @#TSTNM,-(SP)   ;GET READY TO TYPE TEST
13052  040726  104402                           TYPOC                   ;NUMBER
13053  040730  104401  040736                   TYPE    ,67$            ;;TYPE ASCIZ STRING
13054  040734  000414                           BR      66$             ;;GET OVER THE ASCIZ
13055                                  ;;67$:   .ASCIZ  <15><12>/THE LOOP BACK PC WAS /
13056  040766                          66$:
13057  040766  013746  001110                   MOV     @#$LPERR,-(SP)  ;GET READY TO TYPE LOOP BACK PC
13058  040772  104402                           TYPOC
13059  040774  104401  001223                   TYPE    ,$CRLF
13060  041000  104401  041006                   TYPE    ,69$            ;;TYPE ASCIZ STRING
13061  041004  000430                           BR      68$             ;;GET OVER THE ASCIZ
13062                                  ;;69$:   .ASCIZ  <15><12>/SET SWITCH FOR LOOP ON ERROR OR LOOP ON TEST/
13063  041066                          68$:
13064  041066  104401  041074                   TYPE    ,71$            ;;TYPE ASCIZ STRING
13065  041072  000430                           BR      70$             ;;GET OVER THE ASCIZ
13066                                  ;;71$:   .ASCIZ  <15><12>/TYPE THE FIRST PC OF THE TEST TO BE LOOPED ON/
13067  041154                          70$:
13068  041154  104401  041162                   TYPE    ,73$            ;;TYPE ASCIZ STRING
13069  041160  000422                           BR      72$             ;;GET OVER THE ASCIZ
13070                                  ;;73$:   .ASCIZ  <15><12>/ FOLLOWED BY A CARRIAGE RETURN /<15><12>
13071  041226                          72$:
13072  041226  104410                           RDOCT
13073  041230  062716  000002                   ADD     #2,(SP)         ;GET LPADR
13074  041234  012637  001106                   MOV     (SP)+,@#$LPADR
13075  041240  104401  041246                   TYPE    ,75$            ;;TYPE ASCIZ STRING
13076  041244  000417                           BR      74$             ;;GET OVER THE ASCIZ
```

```
13077
13078  041304
13079  041304  104401  041312          ;;75$:  .ASCIZ  <15><12>/TYPE THE PC WHERE YOU WANT/
13080  041310  000440                  74$:
13081                                          TYPE    ,77$            ;;TYPE ASCIZ STRING
13082  041312                                  BR      76$             ;;GET OVER THE ASCIZ
13083  041312  104410                  ;;77$:  .ASCIZ  <15><12>/ THE PROGRAM TO LOOP BACK TO FOLLOWED BY A CARRIAGE RETURN /<15
13084  041314  012637  001110          76$:
13085  041320  013746  001106                  RDOCT
13086  041324  000002                          MOV     (SP)+,@#$LPERR  ;GET LPERR
13087                                          MOV     @#$LPADR,-(SP)
13088                                          RTI
13089
13090
13091                                  ;THIS SAVES THE CONTENTS OF ALL HARDWARE REGISTERS
13092                                  ;IN MEMORY LOCATIONS TAGED FROM "WC" TO "EC2"
13093                                  ;THIS IS DONE SO THAT COMPARES ARE DONE WITH SAVED LOCATIONS
13094                                  ;AND NOT THE REGISTERS THEMSELVES.   THIS WILL MAKE
13095                                  ;ERROR PRINTOUTS FOR GOOD AND BAD DATA ALWAYS DIFFRENT
13096
13097
13098
13099
13100  041426                  PUTREG:
13101  041426  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
13102  041430  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
13103  041432  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
13104  041434  012700  004062          MOV     #RHWC,R0        ;STARTING ADDRESS OF REG
13105  041440  012701  000002          MOV     #WC,R1          ;STARTING ADDRESS OF WERE SAVED
13106  041444  012702  000010          MOV     #RHCC-RHWC+2/2,R2 ;NUMBER OF REG. INTO R2
13107  041450  013021          10$:    MOV     @(R0)+,(R1)+    ;SAVE HARDWARE REG.
13108  041452  005302                  DEC     R2
13109  041454  001375                  BNE     10$
13110  041456  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
13111  041460  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
13112  041462  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
13113  041464  000207                  RTS     PC
```

```
13114
13115                                   .SBTTL  RS DATA TRANSFER SUBROUTINE
13116
13117                                   ;*      THIS SUBROUTINE WRITES OR READS OR DOES A
13118                                   ;*      WRITE CHECK ON CYLINDER 0 TRACK 0 SECTOR 0
13119                                   ;*      IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY
13120                                   ;*      FILLED
13121                                   ;*      WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
13122                                   ;*      MAIN PROGRAM IT MEANS THE TRANSFER IS COMPLETE
13123                                   ;*
13124                                   ;*      THE CALL IS BY A JSR R0,@COMND COMAND
13125
13126  041466  012037  004642          CMNDRS: MOV     (R0)+,COMAND    ;GET COMMAND
13127
13128                                          ;CHECK THAT RHDS1 HAS DRY!DPR!MOL
13129  041472  012737  010600  001124          MOV     #DRY!DPR!MOL,$GDDAT     ;GET GOOD = 10300
13130
13131  041500  017737  142366  001126          MOV     @RHDS1,$BDDAT   ;READ RHDS1 FOR COMPARISON
13132  041506  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
13133                                                                  ;DATA WITH DATA READ FROM
13134                                                                  ;RHDS1
13135  041514  001401                          BEQ     65$             ;BRANCH IF GOOD
13136  041516  104133                          ERROR   133
13137                                                                  ;BEFORE ANY COMMAND IS GIVEN
13138                                                                  ;RHDS1 SHOULD HAVE DRY!DPR!MOL
13139                                                                  ;=10300
13140                                                                  ;BUT CONTAINED WHAT IS
13141                                                                  ;GIVEN IN BAD RHDS1
13142  041520                          65$:
13143
13144  041520  005077  142342                  CLR     @RHDA           ;TRANSFER ON CYLINDER 0
13145                                                                  ;TRANSFER ON SECTOR 0
13146                                                                  ;TRACK 0
13147  041524  017777  143112  142326          MOV     @COMAND,@RHCS1  ;SET UP COMMAND
13148  041532  005737  004644                  TST     NOGO            ;IS GO TO BE GIVEN
13149  041536  001006                          BNE     1$              ;BRANCH IF NO
13150  041540  052777  000001  142312          BIS     #GO,@RHCS1      ;SET GO
13151
13152                                          ;WAIT FOR RDY BIT IN RHCS1 REGISTER TO SET
13153  041546  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13154  041550  004060                          RHCS1                   ;AND WAIT FOR RDY BIT IN
13155  041552  000200                          RDY                     ;RHCS1 REGISTER
13156                                                                  ;IF ERROR OCCURS HERE
13157                                                                  ;IT MEANS "RDY" DID NOT
13158                                                                  ;SET FOR THE FULL COUNT
13159                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13160                                                                  ;THIS TIME IS APPROXIMATELY
13161                                                                  ;LARGER THAN 500 MILLISECONDS
13162  041554  000200                  1$:     RTS     R0
13163
13164                                   .SBTTL  RP04 DATA TRANSFER SUBROUTINE
13165
13166                                   ;*      THIS SUBROUTINE WRITE/READ/WRITE CHECK ON THE RP04 CYLINDER 0,
13167                                   ;*      TRACK 0, SECTOR 0.
13168                                   ;*      IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
13169                                   ;*      WHEN IT RETURNS FROM THIS SUBROUTINE TO THE MAIN
```

```
13170                                  ;*      PROGRAM IT MEANS THE COMMAND IS COMPLETE
13171                                  ;*
13172                                  ;*      THE CALL IS BY A JSR R0,@COMND COMMAND.
13173
13174  041556  012037  004642        CMNDRP: MOV     (R0)+,COMAND    ;GET COMMAND
13175
13176  041562  013746  004172                MOV     PKACK,-(SP)     ;GET READY TO SET VV
13177  041566  052716  000001                BIS     #GO,(SP)        ;GET PACK ACKNOWLEDGE COMMAND
13178  041572  012677  142262                MOV     (SP)+,@RHCS1    ;SET VV
13179
13180                                        ;CHECK THAT RHDS1 HAS VV!DRY!DPR!MOL
13181  041576  012737  010700  001124        MOV     #VV!DRY!DPR!MOL,$GDDAT  ;GET GOOD = 10700
13182
13183  041604  017737  142262  001126        MOV     @RHDS1,$BDDAT   ;READ RHDS1 FOR COMPARISON
13184  041612  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED
13185                                                                ;DATA WITH DATA READ FROM
13186                                                                ;RHDS1
13187  041620  001401                        BEQ     65$             ;BRANCH IF GOOD
13188  041622  104132                        ERROR   132
13189                                                                ;BEFORE ANY COMMAND IS GIVEN
13190                                                                ;RHDS1 SHOULD HAVE VV!DRY!DPR!MOL
13191                                                                ;=10700
13192                                                                ;BUT CONTAINED WHAT IS
13193                                                                ;GIVEN IN BAD RHDS1
13194  041624                          65$:
13195
13196  041624  005077  142264                CLR     @RHCA           ;TRANSFER ON CYLINDER 0
13197  041630  005077  142232                CLR     @RHDST          ;TRANSFER ON SECTOR 0
13198                                                                ;TRACK 0
13199  041634  012777  014000  142250        MOV     #ECI!FMT22,@RHOF;INHIBIT ECC
13200                                                                ;FORMAT 16 BIT PER WORD
13201  041642  017777  142774  142210        MOV     @COMAND,@RHCS1  ;SET UP COMMAND
13202  041650  005737  004644                TST     NOGO            ;IS GO TO BE GIVEN
13203  041654  001006                        BNE     1$              ;BRANCH IF NO
13204  041656  052777  000001  142174        BIS     #GO,@RHCS1      ;SET GO
13205
13206                                        ;WAIT FOR RDY BIT IN RHCS1 REGISTER TO SET
13207  041664  104411                        WAT                     ;TRAP TO WAIT.T SUBROUTINE
13208  041666  004060                        RHCS1                   ;AND WAIT FOR RDY BIT IN
13209  041670  000200                        RDY                     ;RHCS1 REGISTER
13210                                                                ;IF ERROR OCCURS HERE
13211                                                                ;IT MEANS "RDY" DID NOT
13212                                                                ;SET FOR THE FULL COUNT
13213                                                                ;DOWN OF THE WAIT.T SUBROUTINE
13214                                                                ;THIS TIME IS APPROXIMATELY
13215                                                                ;LARGER THAN 500 MILLISECONDS
13216  041672  000200                  1$:   RTS     R0
13217
13218                                  .SBTTL  TM02 DATA TRANSFER SUBROUTINE
13219
13220                                  ;*      THIS SUBROUTINE WRITES/READS/WRITE CHECKS ON THE TM02-TU16
13221                                  ;*      ON THE FIRST BLOCK FROM BEGINNING OF TAPE (BOT)
13222                                  ;*      IT ASSUMES THE RH REGISTERS ARE APPROPRIATELY FILLED.
13223                                  ;*      WHEN IT RETURNS FROM THIS SUBROUTINE TO THE
13224                                  ;*      MAIN PROGRAM IT MEANS THE COMMAND IS COMPLETE.
13225                                  ;*
```

```
13226                                          ;*      THE CALL IS
13227                                          ;*      JSR     R0,@COMND
13228
13229  041674  012037  004642          CMNDTM: MOV     (R0)+,COMAND
13230  041700  013777  004660  142204          MOV     SLAVE,@RHTC     ;GET SLAVE NUMBER
13231  041706  012737  010600  001124          MOV     #MOL!DPR!DRY,$GDDAT ;GET GOOD DATA
13232  041714  017737  142152  001126          MOV     @RHDS1,$BDDAT   ;GET RHDS1
13233  041722  042737  000062  001126          BIC     #BOT!BIT05!BIT04,$BDDAT ;DISREGARD BOT
13234  041730  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE EXPECTED DATA WITH
13235                                                                 ;DATA READ FROM RHDS1
13236  041736  001401                          BEQ     1$              ;BRANCH IF GOOD
13237  041740  104134                          ERROR   134             ;BEFORE ANY COMMAND
13238                                                                 ;IS GIVEN DRIVE STATUS REGISTER
13239                                                                 ;DID NOT CONTAIN WHAT IS IN
13240                                                                 ;GOOD
13241  041742  032777  000002  142122  1$:     BIT     #BOT,@RHDS1     ;IS IT ALREADY AT BOT
13242  041750  001035                          BNE     2$              ;BRANCH IF YES
13243                                          ;IF NOT AT BOT GIVE A REWIND COMMAND
13244  041752  013777  004176  142100          MOV     REWIND,@RHCS1   ;LOAD RHCS1 WITH REWIND COMMAND
13245  041760  052777  000001  142072          BIS     #GO,@RHCS1
13246                                          ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
13247  041766  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13248  041770  004072                          RHDS1                   ;AND WAIT FOR PIP BIT IN
13249  041772  020000                          PIP                     ;RHDS1 REGISTER
13250                                                                 ;IF ERROR OCCURS HERE
13251                                                                 ;IT MEANS "PIP" DID NOT
13252                                                                 ;SET FOR THE FULL COUNT
13253                                                                 ;DOWN OF THE WAIT.T SUBROUTINE
13254                                                                 ;THIS TIME IS APPROXIMATELY
13255                                                                 ;LARGER THAN 500 MILLISECONDS
13256                                          ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
13257  041774  012737  177777  004646                  MOV     #-1,WATO        ;WAIT FOR PIP TO GO FROM 1 TO 0
13258  042002  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13259  042004  004072                          RHDS1                   ;AND WAIT FOR PIP BIT IN
13260  042006  020000                          PIP                     ;RHDS1 REGISTER
13261                                                                 ;IF ERROR OCCURS HERE
13262                                                                 ;IT MEANS "PIP" DID NOT
13263                                                                 ;SET FOR THE FULL COUNT
13264                                                                 ;DOWN OF THE WAIT.T SUBROUTINE
13265                                                                 ;THIS TIME IS APPROXIMATELY
13266                                                                 ;LARGER THAN 500 MILLISECONDS
13267  042010  005037  004646          CLR     WATO
13268                                          ;WAIT FOR BOT BIT IN RHDS1 REGISTER TO SET
13269  042014  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13270  042016  004072                          RHDS1                   ;AND WAIT FOR BOT BIT IN
13271  042020  000002                          BOT                     ;RHDS1 REGISTER
13272                                                                 ;IF ERROR OCCURS HERE
13273                                                                 ;IT MEANS "BOT" DID NOT
13274                                                                 ;SET FOR THE FULL COUNT
13275                                                                 ;DOWN OF THE WAIT.T SUBROUTINE
13276                                                                 ;THIS TIME IS APPROXIMATELY
13277                                                                 ;LARGER THAN 500 MILLISECONDS
13278
13279                                          ;CLEAR ATTENTIONS
13280  042022  013746  004142          MOV     DCLEAR,-(SP)
13281  042026  052716  000001          BIS     #GO,(SP)
```

```
13282  042032  012677  142022                MOV     (SP)+,@RHCS1
13283
13284                                        ;WAIT FOR DRY BIT IN RHDS1 REGISTER TO SET
13285  042036  104411                       WAT                     ;TRAP TO WAIT.T SUBROUTINE
13286  042040  004072                       RHDS1                   ;AND WAIT FOR DRY BIT IN
13287  042042  000200                       DRY                     ;RHDS1 REGISTER
13288                                                                ;IF ERROR OCCURS HERE
13289                                                                ;IT MEANS "DRY" DID NOT
13290                                                                ;SET FOR THE FULL COUNT
13291                                                                ;DOWN OF THE WAIT.T SUBROUTINE
13292                                                                ;THIS TIME IS APPROXIMATELY
13293                                                                ;LARGER THAN 500 MILLISECONDS
13294  042044                        2$:     ;CHECK IF COMMAND IS REVERSE READ
13295                                        ;IF IT IS REVERSE READ, SPACE FORWARD ONE BLOCK
13296  042044  023777  004200  142570        CMP     REVRED,@COMAND  ;IS IT REVERSE READ
13297  042052  001035                       BNE     3$              ;BRANCH IF NOT
13298  042054  052777  001700  142030        BIS     #BPI8!NML,@RHTC ;800 BPI, NORMAL
13299  042062  042777  000010  142022        BIC     #EPAR, @RHTC    ;ODD PARITY
13300
13301  042070  012777  177777  141770        MOV     #-1,@RHFC       ;FRAME COUNT 1
13302                                                                ;FOR SPACE FORWARD ONE BLOCK
13303  042076  013777  004202  141754        MOV     SPACFD,@RHCS1   ;SPACE FORWARD
13304  042104  052777  000001  141746        BIS     #GO,@RHCS1      ;GO
13305                                        ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
13306  042112  104411                       WAT                     ;TRAP TO WAIT.T SUBROUTINE
13307  042114  004072                       RHDS1                   ;AND WAIT FOR PIP BIT IN
13308  042116  020000                       PIP                     ;RHDS1 REGISTER
13309                                                                ;IF ERROR OCCURS HERE
13310                                                                ;IT MEANS "PIP" DID NOT
13311                                                                ;SET FOR THE FULL COUNT
13312                                                                ;DOWN OF THE WAIT.T SUBROUTINE
13313                                                                ;THIS TIME IS APPROXIMATELY
13314                                                                ;LARGER THAN 500 MILLISECONDS
13315                                        ;WAIT FOR PIP BIT IN RHDS1 REGISTER TO SET
13316  042120  012737  177777  004646                MOV     #-1,WATO        ;WAIT FOR PIP TO GO FROM 1 TO 0
13317  042126  104411                       WAT                     ;TRAP TO WAIT.T SUBROUTINE
13318  042130  004072                       RHDS1                   ;AND WAIT FOR PIP BIT IN
13319  042132  020000                       PIP                     ;RHDS1 REGISTER
13320                                                                ;IF ERROR OCCURS HERE
13321                                                                ;IT MEANS "PIP" DID NOT
13322                                                                ;SET FOR THE FULL COUNT
13323                                                                ;DOWN OF THE WAIT.T SUBROUTINE
13324                                                                ;THIS TIME IS APPROXIMATELY
13325                                                                ;LARGER THAN 500 MILLISECONDS
13326  042134  005037  004646                CLR     WATO
13327                                        ;WAIT FOR DRY BIT IN RHDS1 REGISTER TO SET
13328  042140  104411                       WAT                     ;TRAP TO WAIT.T SUBROUTINE
13329  042142  004072                       RHDS1                   ;AND WAIT FOR DRY BIT IN
13330  042144  000200                       DRY                     ;RHDS1 REGISTER
13331                                                                ;IF ERROR OCCURS HERE
13332                                                                ;IT MEANS "DRY" DID NOT
13333                                                                ;SET FOR THE FULL COUNT
13334                                                                ;DOWN OF THE WAIT.T SUBROUTINE
13335                                                                ;THIS TIME IS APPROXIMATELY
13336                                                                ;LARGER THAN 500 MILLISECONDS
13337  042146                        3$:     ;NOW GIVE COMMAND
```

```
13338  042146  052777  001700  141736          BIS     #BPI8!NML,@RHTC ;800 BPI, NORMAL
13339  042154  042777  000010  141730          BIC     #EPAR,@RHTC     ;ODD PARITY
13340  042162  017746  141674                  MOV     @RHWC,-(SP)     ;GET WORD COUNT
13341  042166  005416                          NEG     (SP)            ;GET POSITIVE NUMBER
13342  042170  061616                          ADD     (SP),(SP)       ;DOUBLE WORD COUNT
13343  042172  005416                          NEG     (SP)            ;GET NEGATIVE NUMBER FOR
13344                                                                  ;FRAME COUNT REGISTER
13345  042174  012677  141666                  MOV     (SP)+,@RHFC     ;FILL FRAME COUNT
13346  042200  017777  142436  141652          MOV     @COMAND,@RHCS1  ;GET COMMAND
13347
13348  042206  005737  004644                  TST     NOGO            ;IS GO TO BE GIVEN
13349  042212  001006                          BNE     4$              ;BRANCH IF NO
13350  042214  052777  000001  141636          BIS     #GO,@RHCS1      ;GO
13351
13352                                          ;WAIT FOR DRY BIT IN RHDS1 REGISTER TO SET
13353  042222  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTIN_
13354  042224  004072                          RHDS1                   ;AND WAIT FOR DRY BIT IN
13355  042226  000200                          DRY                     ;RHDS1 REGISTER
13356                                                                  ;IF ERROR OCCURS HERE
13357                                                                  ;IT MEANS "DRY" DID NOT
13358                                                                  ;SET FOR THE FULL COUNT
13359                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13360                                                                  ;THIS TIME IS APPROXIMATELY
13361                                                                  ;LARGER THAN 500 MILLISECONDS
13362  042230                          4$:
13363                                          ;WAIT FOR RDY BIT IN RHCS1 REGISTER TO SET
13364  042230  104411                          WAT                     ;TRAP TO WAIT.T SUBROUTINE
13365  042232  004060                          RHCS1                   ;AND WAIT FOR RDY BIT IN
13366  042234  000200                          RDY                     ;RHCS1 REGISTER
13367                                                                  ;IF ERROR OCCURS HERE
13368                                                                  ;IT MEANS "RDY" DID NOT
13369                                                                  ;SET FOR THE FULL COUNT
13370                                                                  ;DOWN OF THE WAIT.T SUBROUTINE
13371                                                                  ;THIS TIME IS APPROXIMATELY
13372                                                                  ;LARGER THAN 500 MILLISECONDS
13373  042236  000200                          RTS     R0
```

```
13374
13375                                   ;*      THIS ROUTINE WILL ALLOW THE CHANGE OF THE BASE
13376                                   ;*      ADDRESS FROM 176700 TO ANY TYPED VALUE
13377
13378  042240                           BASECH:
13379  042240  104401  042246                   TYPE    ,65$                ;;TYPE ASCIZ STRING
13380  042244  000424                           BR      64$                 ;;GET OVER THE ASCIZ
13381                                   ;;65$:  .ASCIZ  <15><12>/PRESENT BASE ADDRESS OF REGISTERS IS /
13382  042316                           64$:
13383  042316  013746  004060                   MOV     @#RHCS1,-(SP)       ;GET READY TO TYPE OLD BASE
13384  042322  104402                           TYPOC
13385  042324  104401  042332                   TYPE    ,67$                ;;TYPE ASCIZ STRING
13386  042330  000425                           BR      66$                 ;;GET OVER THE ASCIZ
13387                                   ;;67$:  .ASCIZ  <15><12>/TYPE NEW BASE ADDRESS FOLLOWED BY 'CR'/
13388  042404                           66$:
13389  042404  104410                           RDOCT
13390  042406  012700  004102                   MOV     #RHDB,R0            ;GET STARTING ADDRESS OF RGISTERS
13391  042412  012701  000024                   MOV     #20.,R1             ;NUMBER OF REGISTERS
13392  042416  042710  177700           1$:     BIC     #^C77,(R0)          ;CLEAR OLD BASE
13393  042422  051620                           BIS     (SP),(R0)+          ;SET NEW BASE
13394  042424  005301                           DEC     R1                  ;COUNT
13395  042426  001373                           BNE     1$                  ;BRANCH IF 20 NOT DONE
13396  042430  104401  042436                   TYPE    ,69$                ;;TYPE ASCIZ STRING
13397  042434  000417                           BR      68$                 ;;GET OVER THE ASCIZ
13398                                   ;;69$:  .ASCIZ  <15><12>/PRESENT VECTOR ADDRESS IS /
13399  042474                           68$:
13400  042474  013746  002716                   MOV     @#RPVEC,-(SP)       ;GET READY TO TYPE OLD VECTOR ADDRESS
13401  042500  104402                           TYPOC
13402  042502  104401  042510                   TYPE    ,71$                ;;TYPE ASCIZ STRING
13403  042506  000437                           BR      70$                 ;;GET OVER THE ASCIZ
13404                                   ;;71$:  .ASCIZ  <15><12>/TYPE NEW VECTOR ADDRESS OR RETYPE OLD ONE FOLLOWED BY "CR"/
13405  042606                           70$:
13406  042606  104410                           RDOCT
13407  042610  012637  002716                   MOV     (SP)+,@#RPVEC       ;SETUP VECTOR ADDRESS
13408  042614  104401  042622                   TYPE    ,73$                ;;TYPE ASCIZ STRING
13409  042620  000421                           BR      72$                 ;;GET OVER THE ASCIZ
13410                                   ;;73$:  .ASCIZ  <15><12>/RESTART PROGRAM FROM 200 OR 210/
13411  042664                           72$:
13412  042664  104401  042672                   TYPE    ,75$                ;;TYPE ASCIZ STRING
13413  042670  000414                           BR      74$                 ;;GET OVER THE ASCIZ
13414                                   ;;75$:  .ASCIZ  <15><12>/NEW BASE WILL REMAIN/
13415  042722                           74$:
13416  042722  013746  004060                   MOV     @#RHCS1,-(SP)
13417  042726  104402                           TYPOC
13418  042730  104401  042736                   TYPE    ,77$                ;;TYPE ASCIZ STRING
13419  042734  000415                           BR      76$                 ;;GET OVER THE ASCIZ
13420                                   ;;77$:  .ASCIZ  <15><12>/NEW VECTOR WILL REMAIN /
13421  042770                           76$:
13422  042770  013746  002716                   MOV     @#RPVEC,-(SP)
13423  042774  104402                           TYPOC
13424  042776  104401  043004                   TYPE    ,79$                ;;TYPE ASCIZ STRING
13425  043002  000416                           BR      78$                 ;;GET OVER THE ASCIZ
13426                                   ;;79$:  .ASCIZ  <15><12>/UNTIL PROGRAM IS RELOADED/
13427  043040                           78$:
13428  043040  000000                           HALT
13429
```

```
 13430   043042                         RPVECT:
 13431   043042  104401  043050                 TYPE    ,65$                    ;;TYPE ASCIZ STRING
 13432   043046  000411                 BR      64$                     ;;GET OVER THE ASCIZ
 13433                                  ;;65$:  .ASCIZ  /TRAPED FROM PC = /
 13434   043072                         64$:
 13435   043072  104402                 TYPOC                                   ;TYPE FROM PC
 13436   043074  012777  043042  137614         MOV     #RPVECT,@RPVEC          ;RESTORE TRAP RP04 VECTOR
 13437   043102  000000                 HALT                                    ;CHANGE TO CONTINUE
 13438
 13439
 13440
 13441
```

```
                                                                        ;****************************
                                       .SBTTL  SCOPE HANDLER ROUTINE

                                       ;;*************************************************************
                                       ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                       ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                       ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                       ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                       ;*SW14=1         LOOP ON TEST
                                       ;*SW11=1         INHIBIT ITERATIONS
                                       ;*SW09=1         LOOP ON ERROR
                                       ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                                       ;*CALL
                                       ;*       SCOPE                 ;;SCOPE=IOT

043104                                 $SCOPE:
043104  005037  004646                         CLR     WAT0            ;WAIT FOR BIT TO SET
                                                                       ;USED IN "WAT" WAIT ROUTINE
043110  005037  004644                         CLR NOGO                ;DO GIVE GO IN COMMAND ROUTINE
043114  032777  040000  136016         1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
043122  001111                                 BNE     $OVER           ;;YES IF SW14=1
                                       ;#####START OF CODE FOR THE XOR TESTER#####
043124  000416                         $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                       ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
043126  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
043132  012737  043152  000004                 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
043140  005737  177060                         TST     @#177060        ;;TIME OUT ON XOR?
043144  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
043150  000463                                 BR      $SVLAD          ;;GO TO THE NEXT TEST
043152  022626                         5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
043154  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
043160  000423                                 BR      7$              ;;LOOP ON THE PRESENT TEST
043162                                 6$:;#####END OF CODE FOR THE XOR TESTER#####
043162  032777  000400  135750                 BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
043170  001404                                 BEQ     2$              ;;BR IF NO
043172  127737  135742  001102                 CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?     SWR<7:0>
043200  001462                                 BEQ     $OVER           ;;BR IF YES
043202  105737  001103                 2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
043206  001421                                 BEQ     3$              ;;BR IF NO
043210  123737  001115  001103                 CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
043216  101015                                 BHI     3$              ;;BR IF NO
043220  032777  001000  135712                 BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
043226  001404                                 BEQ     4$              ;;BR IF NO
043230  013737  001110  001106         7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
043236  000443                                 BR      $OVER
043240  105037  001103                 4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
043244  005037  001212                         CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
043250  000415                                 BR      1$              ;;ESCAPE TO THE NEXT TEST
043252  032777  004000  135660         3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
043260  001011                                 BNE     1$              ;;BR IF YES
043262  005737  001100                         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
043266  001406                                 BEQ     1$              ;;      INHIBIT ITERATIONS
043270  005237  001104                         INC     $ICNT           ;;INCREMENT ITERATION COUNT
043274  023737  001212  001104                 CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
043302  002021                                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
043304  012737  000001  001104         1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
```

```
 13498   043312   013737   043362   001212            MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
 13499   043320   105237   001102             $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
 13500   043324   011637   001106                     MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 13501   043330   011637   001110                     MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 13502   043334   005037   001214                     CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 13503   043340   112737   000001   001115            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 13504   043346   013777   001102   135566   $OVER:   MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 13505   043354   013716   001106                     MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 13506   043360   000002                              RTI                     ;;FIXES PS
 13507   043362   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

043364                          $TYPDS:
043364  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
043366  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
043370  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
043372  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
043374  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
043376  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
043402  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
043406  100004                        BPL     1$              ;;BR IF INPUT IS POS.
043410  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
043412  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
043420  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
043422  012703  043600                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
043426  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
043432  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
043434  016001  043570                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
043440  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
043442  002402                        BLT     4$              ;;BR IF DONE
043444  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
043446  000774                        BR      3$
043450  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
043452  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
043454  001002                        BNE     5$              ;;FALL THROUGH IF 0
043456  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
043460  100407                        BMI     7$              ;;BR IF YES
043462  106316                5$:     ASLB    (SP)            ;;MSD?
043464  103003                        BCC     6$              ;;BR IF NO
043466  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
043474  052702  000060        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
043500  052702  000040        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
043504  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
043506  005720                        TST     (R0)+           ;;JUST INCREMENTING
043510  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
043514  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
043516  003002                        BGT     8$              ;;GO TO EXIT
043520  010502                        MOV     R5,R2           ;;GET THE LSD
043522  000764                        BR      6$              ;;GO CHANGE TO ASCII
043524  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
043526  100003                        BPL     9$              ;;BR IF NO
043530  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
043536  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
043540  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
043542  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
043544  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
13564  043546  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
13565  043550  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
13566  043552  104401  043600                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
13567  043556  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
13568  043564  012616                        MOV     (SP)+,(SP)
13569  043566  000002                        RTI                     ;;RETURN TO USER
13570  043570  023420                $DTBL:  10000.
13571  043572  001750                        1000.
13572  043574  000144                        100.
13573  043576  000012                        10.
13574  043600  000004                $DBLK:  .BLKW   4
13575                                 .SBTTL  TYPE ROUTINE
13576
13577                                 ;;*****************************************************************
13578                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
13579                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
13580                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
13581                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
13582                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
13583                                 ;*
13584                                 ;*CALL:
13585                                 ;*1) USING A TRAP INSTRUCTION
13586                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
13587                                 ;*OR
13588                                 ;*      TYPE
13589                                 ;*      MESADR
13590                                 ;*
13591
13592  043610  105737  001157        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
13593  043614  100002                        BPL     1$              ;;BR IF YES
13594  043616  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
13595  043620  000407                        BR      3$              ;;LEAVE
13596  043622  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
13597  043624  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
13598  043630  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
13599  043632  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
13600  043634  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
13601  043636  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
13602  043640  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
13603  043644  000002                        RTI                     ;;RETURN
13604  043646  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
13605  043652  001430                        BEQ     8$
13606  043654  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
13607  043660  001006                        BNE     5$
13608  043662  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
13609  043664  104401                        TYPE                    ;;TYPE A CR AND LF
13610  043666  001223                        $CRLF
13611  043670  105037  044024                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
13612  043674  000755                        BR      2$              ;;GET NEXT CHARACTER
13613  043676  004737  043760        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
13614  043702  123726  001156        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
13615  043706  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
13616  043710  013746  001154                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
13617                                                                 ;;AND THE NULL CHAR.
13618  043714  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
13619  043720  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
13620  043722  004737  043760                JSR     PC,$TYPEC       ;;GO TYPE A NULL
13621  043726  105337  044024                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
13622  043732  000770                        BR      7$              ;;LOOP
13623
13624                                ;HORIZONTAL TAB PROCESSOR
13625
13626  043734  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
13627  043740  004737  043760        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
13628  043744  132737  000007  044024        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
13629  043752  001372                        BNE     9$              ;;TAB STOP
13630  043754  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
13631  043756  000724                        BR      2$              ;;GET NEXT CHARACTER
13632  043760  105777  135164        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
13633  043764  100375                        BPL     $TYPEC
13634  043766  116677  000002  135156        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
13635  043774  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
13636  044002  001003                        BNE     1$              ;;BRANCH IF NO
13637  044004  105037  044024                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
13638  044010  000406                        BR      $TYPEX          ;;EXIT
13639  044012  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
13640  044020  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
13641  044022  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
13642  044024  000000        $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
13643  044026  000207        $TYPEX: RTS     PC
13644
13645                                .SBTTL  TTY INPUT ROUTINE
13646
13647                                ;;**************************************************************
13648                                .ENABL  LSB
13649  044030  000000        $TKCNT: .WORD   0                       ;;NUMBER OF ITEMS IN QUEUE
13650  044032  000000        $TKQIN: .WORD   0                       ;;INPUT POINTER
13651  044034  000000        $TKQOUT: .WORD  0                       ;;OUTPUT POINTER
13652  044036  000011        $TKQSRT: .BLKB  9.                      ;;TTY KEYBOARD QUEUE
13653          044047        $TKQEND=.
13654          044050                        .EVEN
13655
13656                                ;*TK INITIALIZE ROUTINE
13657                                ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
13658                                ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
13659                                ;
13660                                ;*CALL:
13661                                ;*      JSR     PC,$TKINT
13662                                ;*      RETURN
13663                                ;
13664  044050  005037  044030        $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
13665  044054  012737  044036  044032        MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
13666  044062  013737  044032  044034        MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
13667  044070  012737  044120  000060        MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
13668  044076  012737  000200  000062        MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
13669  044104  005777  135036              TST     @$TKB           ;;CLEAR DONE FLAG
13670  044110  012777  000100  135026        MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
13671  044116  000207                        RTS     PC              ;;RETURN TO CALLER
13672
13673                                ;*TK SERVICE ROUTINE
13674                                ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
13675                                ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
```

```
13676                                   ;*IT IN THE QUEUE.
13677                                   ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) $TKINT IS CALLED AND
13678                                   ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (OPERSEL)
13679                                   ;
13680  044120  117746  135022          $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
13681  044124  042716  177600                  BIC     #↑C177,(SP)     ;;STRIP THE JUNK
13682  044130  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
13683  044134  001007                          BNE     1$              ;;BRANCH IF NO
13684  044136  104401  044527                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (↑C)
13685  044142  004737  044050                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
13686  044146  005726                          TST     (SP)+           ;;CLEAN UP STACK
13687  044150  000137  040640                  JMP     OPERSEL         ;;CONTROL C RESTART
13688
13689  044154                          1$:
13690  044154  022737  000011  044030          CMP     #9.,$TKCNT      ;;IS THE QUEUE FULL?
13691  044162  001004                          BNE     3$              ;;BRANCH IF NO
13692  044164  104401  001216                  TYPE    ,$BELL          ;;RING THE TTY BELL
13693  044170  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
13694  044172  000451                          BR      5$              ;;EXIT
13695  044174  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
13696  044200  001021                          BNE     32$             ;;BRANCH IF NO
13697  044202  005077  134736                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
13698  044206  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
13699  044210  105777  134730          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
13700  044214  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
13701  044216  117746  134724                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
13702  044222  042716  177600                  BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
13703  044226  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
13704  044232  001366                          BNE     31$             ;;BRANCH IF NO
13705  044234  012777  000100  134702          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
13706  044242  000002                          RTI                     ;;RETURN
13707  044244  005237  044030          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
13708  044250  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
13709  044254  002405                          BLT     4$              ;;BRANCH IF YES
13710  044256  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
13711  044262  003002                          BGT     4$              ;;BRANCH IF YES
13712  044264  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
13713  044270  112677  177536          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
13714  044274  005237  044032                  INC     $TKQIN          ;;UPDATE THE POINTER
13715  044300  023727  044032  044047          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
13716  044306  001003                          BNE     5$              ;;BRANCH IF NO
13717  044310  012737  044036  044032          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
13718  044316  000002                  5$:     RTI                     ;;RETURN
13719
13720                                   .DSABL  LSB
13721
13722
13723                                   ;;*****************************************************************
13724                                   ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
13725                                   ;*CALL:
13726                                   ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
13727                                   ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
13728                                   ;*                              ;;WITH PARITY BIT STRIPPED OFF
13729                                   ;
13730
13731  044320  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
```

```
13732  044322  016666  000004  000002          MOV     4(SP),2(SP)     ;;THE PS
13733  044330  005066  000004                  CLR     4(SP)           ;;GET READY FOR A CHARACTER
13734  044334  005046                          CLR     -(SP)           ;;PUT NEW PS ON STACK
13735  044336  012746  044344                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
13736  044342  000002                          RTI                     ;;POP NEW PC AND PS
13737  044344                          64$:
13738  044344  005737  044030          1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
13739  044350  001775                          BEQ     1$
13740  044352  005337  044030                  DEC     $TKCNT          ;;DECREMENT THE COUNTER
13741  044356  117766  177452  000004          MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
13742  044364  005237  044034                  INC     $TKQOUT         ;;UPDATE THE POINTER
13743  044370  023727  044034  044047          CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
13744  044376  001003                          BNE     2$              ;;BRANCH IF NO
13745  044400  012737  044036  044034          MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
13746  044406  000002          2$:             RTI                     ;;RETURN
13747                          ;;*******************************************************
13748                          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
13749                          ;*CALL:
13750                          ;*      RDLIN                           ;;INPUT A STRING FROM THE TTY
13751                          ;*      RETURN HERE                     ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
13752                          ;*                                      ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
13753
13754  044410  010346          $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
13755  044412  012703  044516  1$:     MOV     #$TTYIN,R3              ;;GET ADDRESS
13756  044416  022703  044527  2$:     CMP     #$TTYIN+9.,R3           ;;BUFFER FULL?
13757  044422  101405                  BLOS    4$                      ;;BR IF YES
13758  044424  104406                  RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
13759  044426  112613                  MOVB    (SP)+,(R3)              ;;GET CHARACTER
13760  044430  122713  000177  10$:     CMPB    #177,(R3)               ;;IS IT A RUBOUT
13761  044434  001003                  BNE     3$                      ;;SKIP IF NOT
13762  044436  104401  001222  4$:     TYPE    ,$QUES                  ;;TYPE A '?'
13763  044442  000763                  BR      1$                      ;;CLEAR THE BUFFER AND LOOP
13764  044444  111337  044514  3$:     MOVB    (R3),9$                 ;;ECHO THE CHARACTER
13765  044450  104401  044514          TYPE    ,9$
13766  044454  122723  000015          CMPB    #15,(R3)+               ;;CHECK FOR RETURN
13767  044460  001356                  BNE     2$                      ;;LOOP IF NOT RETURN
13768  044462  105063  177777          CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
13769  044466  104401  001224          TYPE    ,$LF                    ;;TYPE A LINE FEED
13770  044472  012603                  MOV     (SP)+,R3                ;;RESTORE R3
13771  044474  011646                  MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
13772  044476  016666  000004  000002  MOV     4(SP),2(SP)             ;;      FIRST ASCII CHARACTER ON IT
13773  044504  012766  044516  000004  MOV     #$TTYIN,4(SP)
13774  044512  000002                  RTI                             ;;RETURN
13775  044514     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
13776  044515     000                  .BYTE   0                       ;;TERMINATOR
13777  044516  000011          $TTYIN: .BLKB   9.                      ;;RESERVE 9. BYTES FOR TTY INPUT
13778  044527     136  006503  000012  $CNTLC: .ASCIZ  /↑C/<15><12>    ;;CONTROL "C"
13779  044534  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
13780  044541     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
13781  044546  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
13782  044554  020075     000
13783  044557     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
13784  044564  036440  000040
13785                                                           ;FROM THE TTY
13786
13787
```

```
13788
13789                                   .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
13790
13791                                   ;;**********************************************************************
13792                                   ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
13793                                   ;*CHANGE IT TO BINARY.
13794                                   ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
13795                                   ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
13796                                   ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
13797                                   ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
13798                                   ;*CALL:
13799                                   ;*      RDOCT                   ;;READ AN OCTAL NUMBER
13800                                   ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
13801                                   ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
13802
13803  044570  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
13804  044572  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
13805  044600  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
13806  044602  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
13807  044604  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
13808  044606  104407                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
13809  044610  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
13810  044612  010037  044716                  MOV     R0,5$           ;;AND SAVE IT
13811  044616  005001                          CLR     R1              ;;CLEAR DATA WORD
13812  044620  005002                          CLR     R2
13813  044622  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
13814  044624  001420                          BEQ     3$              ;;IF ZERO GET OUT
13815  044626  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
13816  044632  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
13817  044634  122716  000067                  CMPB    #'7,(SP)
13818  044640  002423                          BLT     4$
13819  044642  006301                          ASL     R1              ;;*2
13820  044644  006102                          ROL     R2
13821  044646  006301                          ASL     R1              ;;*4
13822  044650  006102                          ROL     R2
13823  044652  006301                          ASL     R1              ;;*8
13824  044654  006102                          ROL     R2
13825  044656  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
13826  044662  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
13827  044664  000756                          BR      2$              ;;LOOP
13828  044666  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
13829  044670  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
13830  044674  010237  044726                  MOV     R2,$HIOCT
13831  044700  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
13832  044702  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
13833  044704  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
13834  044706  000002                          RTI                     ;;RETURN
13835  044710  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
13836  044712  105010                          CLRB    (R0)            ;;SET A TERMINATOR
13837  044714  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
13838  044716  000000                  5$:     .WORD   0
13839  044720  104401  001222                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
13840  044724  000730                          BR      1$              ;;TRY AGAIN
13841  044726  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
13842                                  .SBTTL  ERROR HANDLER ROUTINE
13843
13844                                  ;;**********************************************************************
13845                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
13846                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
13847                                  ;*AND GO TO $ERRTYP ON ERROR
13848                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
13849                                  ;*SW15=1          HALT ON ERROR
13850                                  ;*SW13=1          INHIBIT ERROR TYPEOUTS
13851                                  ;*SW10=1          BELL ON ERROR
13852                                  ;*SW09=1          LOOP ON ERROR
13853                                  ;*CALL
13854                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
13855
13856  044730                          $ERROR:
13857  044730  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
13858  044734  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
13859  044736  013777  001102  134176          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
13860  044744  032777  002000  134166          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
13861  044752  001402                          BEQ     1$              ;;NO - SKIP
13862  044754  104401  001216                  TYPE    ,$BELL          ;;RING BELL
13863  044760  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
13864  044764  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
13865  044770  162737  000002  001116          SUB     #2,$ERRPC
13866  044776  117737  134114  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
13867  045004  032777  020000  134126          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
13868  045012  001004                          BNE     20$             ;;SKIP TYPEOUTS
13869  045014  004737  045066                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
13870  045020  104401  001223                  TYPE    ,$CRLF
13871  045024                          20$:
13872  045024  [illegible]777  134110  2$:     TST     @SWR            ;;HALT ON ERROR
13873  045030  100001                          BPL     3$              ;;SKIP IF CONTINUE
13874  045032  000000                          HALT                    ;;HALT ON ERROR!
13875  045034  032777  001000  134076  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
13876  045042  001402                          BEQ     4$              ;;BR IF NO
13877  045044  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
13878  045050  005737  001214          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
13879  045054  001402                          BEQ     5$              ;;BR IF NONE
13880  045056  013716  001214                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
13881  045062                          5$:
13882  045062  000002                          RTI                     ;;RETURN
13883                                  ;;**********************************************************
13884                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
13885                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
13886                                  ;*ERROR IS TO BE REPORTED.  IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
13887                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
13888                                  ;*IT IS A COPY OF THE $ERRTYP SUBROUTINE FROM SYSMAC.
13889                                  ;*WITH ONLY MINOR CHANGES
13890                                  ;*FIRST IF SWITCH 6 IS SET AND SWITCH 8 RESET THEN
13891                                  ;*ALL REGISTER CONTENTS WILL BE TYPED BEFOR REPORTING THE ERROR
13892                                  ;*SECOND IF THE CURRENT ERROR HAS THE SAME ITEM NUMBER
13893                                  ;*AS THE PREVIOUS ERROR THEN ONLY THE DATA WILL BE TYPED
13894                                  ;*AND NOT THE ERROR MESSAGE AND HEADER.
13895  045064  000000                  PRITEM: 0                       ;PREVIOUS ITEM NO. LOCATION
13896  045066  013746  001140          $ERRTYP: MOV    @#SWR,-(SP)     ;GET SWITCH SETTING
13897  045072  042716  177277                   BIC    #^C500,(SP)     ;KEEP ONLY SWITCH 8 AND 6
```

```
13898   045076  022726  000100          CMP     #SW06,(SP)+     ;IS 6 SET AND 8 RESET
13899   045102  001001                  BNE     1$              ;IF NOT BRANCH
13900   045104  000402                  BR      2$              ;BRANCH IF SW 6 IS SET AND 8 RESET
13901   045106  000137  046006  1$:     JMP     @#TYPERR        ;JUMP IF SW 8 IS SET
13902                                                           ;OR IF SW 8 IS RESET AND SW 6 IS RESET
13903   045112  104401  000200  2$:     TYPE    ,CRLF
13904
13905   045116  104401  045124          TYPE    ,65$            ;;TYPE ASCIZ STRING
13906   045122  000405                  BR      64$             ;;GET OVER THE ASCIZ
13907                           ;;65$:  .ASCIZ  /   RHWC= /
13908   045136                  64$:
13909   045136  017746  136720          MOV     @RHWC,-(SP)     ;GET READY TO TYPE RHWC CONTENTS
13910   045142  104402                  TYPOC
13911
13912
13913   045144  104401  045152          TYPE    ,67$            ;;TYPE ASCIZ STRING
13914   045150  000405                  BR      66$             ;;GET OVER THE ASCIZ
13915                           ;;67$:  .ASCIZ  /   RHBA= /
13916   045164                  66$:
13917   045164  017746  136674          MOV     @RHBA,-(SP)     ;GET READY TO TYPE RHBA CONTENTS
13918   045170  104402                  TYPOC
13919
13920
13921   045172  104401  045200          TYPE    ,69$            ;;TYPE ASCIZ STRING
13922   045176  000406                  BR      68$             ;;GET OVER THE ASCIZ
13923                           ;;69$:  .ASCIZ  /   RHCS2= /
13924   045214                  68$:
13925   045214  017746  136650          MOV     @RHCS2,-(SP)    ;GET READY TO TYPE RHCS2 CONTENTS
13926   045220  104402                  TYPOC
13927
13928
13929   045222  104401  045230          TYPE    ,71$            ;;TYPE ASCIZ STRING
13930   045226  000406                  BR      70$             ;;GET OVER THE ASCIZ
13931                           ;;71$:  .ASCIZ  /   RHCS1= /
13932   045244                  70$:
13933   045244  017746  136610          MOV     @RHCS1,-(SP)    ;GET READY TO TYPE RHCS1 CONTENTS
13934   045250  104402                  TYPOC
13935
13936
13937   045252  104401  045260          TYPE    ,73$            ;;TYPE ASCIZ STRING
13938   045256  000406                  BR      72$             ;;GET OVER THE ASCIZ
13939                           ;;73$:  .ASCIZ  /   RHDS1= /
13940   045274                  72$:
13941   045274  017746  136572          MOV     @RHDS1,-(SP)    ;GET READY TO TYPE RHDS1 CONTENTS
13942   045300  104402                  TYPOC
13943
13944
13945   045302  104401  045310          TYPE    ,75$            ;;TYPE ASCIZ STRING
13946   045306  000406                  BR      74$             ;;GET OVER THE ASCIZ
13947                           ;;75$:  .ASCIZ  /   RHER1= /
13948   045324                  74$:
13949   045324  017746  136544          MOV     @RHER1,-(SP)    ;GET READY TO TYPE RHER1 CONTENTS
13950   045330  104402                  TYPOC
13951
13952
13953   045332  104401  045340          TYPE    ,77$            ;;TYPE ASCIZ STRING
```

```
13954  045336  000406                        BR      76$             ;;GET OVER THE ASCIZ
13955                                ;;77$:    .ASCIZ  /   RHER2= /
13956  045354                        76$:
13957  045354  017746  136540                MOV     @RHER2,-(SP)    ;GET READY TO TYPE RHER2 CONTENTS
13958  045360  104402                        TYPOC
13959
13960
13961  045362  104401  045370                TYPE    ,79$            ;;TYPE ASCIZ STRING
13962  045366  000406                        BR      78$             ;;GET OVER THE ASCIZ
13963                                ;;79$:    .ASCIZ  /   RHER3= /
13964  045404                        78$:
13965  045404  017746  136512                MOV     @RHER3,-(SP)    ;GET READY TO TYPE RHER3 CONTENTS
13966  045410  104402                        TYPOC
13967
13968
13969  045412  104401  045420                TYPE    ,81$            ;;TYPE ASCIZ STRING
13970  045416  000406                        BR      80$             ;;GET OVER THE ASCIZ
13971                                ;;81$:    .ASCIZ  /   RHDST= /
13972  045434                        80$:
13973  045434  017746  136426                MOV     @RHDST,-(SP)    ;GET READY TO TYPE RHDST CONTENTS
13974  045440  104402                        TYPOC
13975
13976
13977  045442  104401  045450                TYPE    ,83$            ;;TYPE ASCIZ STRING
13978  045446  000405                        BR      82$             ;;GET OVER THE ASCIZ
13979                                ;;83$:    .ASCIZ  /   RHCA= /
13980  045462                        82$:
13981  045462  017746  136426                MOV     @RHCA,-(SP)     ;GET READY TO TYPE RHCA CONTENTS
13982  045466  104402                        TYPOC
13983
13984
13985  045470  104401  045476                TYPE    ,85$            ;;TYPE ASCIZ STRING
13986  045474  000405                        BR      84$             ;;GET OVER THE ASCIZ
13987                                ;;85$:    .ASCIZ  /   RHAS= /
13988  045510                        84$:
13989  045510  017746  136362                MOV     @RHAS,-(SP)     ;GET READY TO TYPE RHAS CONTENTS
13990  045514  104402                        TYPOC
13991
13992
13993  045516  104401  045524                TYPE    ,87$            ;;TYPE ASCIZ STRING
13994  045522  000405                        BR      86$             ;;GET OVER THE ASCIZ
13995                                ;;87$:    .ASCIZ  /   RHOF= /
13996  045536                        86$:
13997  045536  017746  136350                MOV     @RHOF,-(SP)     ;GET READY TO TYPE RHOF CONTENTS
13998  045542  104402                        TYPOC
13999
14000
14001  045544  104401  045552                TYPE    ,89$            ;;TYPE ASCIZ STRING
14002  045550  000405                        BR      88$             ;;GET OVER THE ASCIZ
14003                                ;;89$:    .ASCIZ  /   RHMR= /
14004  045564                        88$:
14005  045564  017746  136314                MOV     @RHMR,-(SP)     ;GET READY TO TYPE RHMR CONTENTS
14006  045570  104402                        TYPOC
14007
14008
14009  045572  104401  045600                TYPE    ,91$            ;;TYPE ASCIZ STRING
```

```
14010  045576  000405                        BR      90$             ;;GET OVER THE ASCIZ
14011                                  ;;91$:  .ASCIZ  /   RHLA= /
14012  045612                          90$:
14013  045612  017746  136262                MOV     @RHLA,-(SP)     ;GET READY TO TYPE RHLA CONTENTS
14014  045616  104402                        TYPOC
14015
14016
14017  045620  104401  045626                TYPF    ,93$            ;;TYPE ASCIZ STRING
14018  045624  000405                        BR      92$             ;;GET OVER THE ASCIZ
14019                                  ;;93$:  .ASCIZ  /   RHCC= /
14020  045640                          92$:
14021  045640  017746  136234                MOV     @RHCC,-(SP)     ;GET READY TO TYPE RHCC CONTENTS
14022  045644  104402                        TYPOC
14023
14024
14025  045646  104401  045654                TYPE    ,95$            ;;TYPE ASCIZ STRING
14026  045652  000406                        BR      94$             ;;GET OVER THE ASCIZ
14027                                  ;;95$:  .ASCIZ  /   RHEC1= /
14028  045670                          94$:
14029  045670  017746  136230                MOV     @RHEC1,-(SP)    ;GET READY TO TYPE RHEC1 CONTENTS
14030  045674  104402                        TYPOC
14031
14032
14033  045676  104401  045704                TYPE    ,97$            ;;TYPE ASCIZ STRING
14034  045702  000406                        BR      96$             ;;GET OVER THE ASCIZ
14035                                  ;;97$:  .ASCIZ  /   RHEC2= /
14036  045720                          96$:
14037  045720  017746  136202                MOV     @RHEC2,-(SP)    ;GET READY TO TYPE RHEC2 CONTENTS
14038  045724  104402                        TYPOC
14039
14040
14041  045726  104401  045734                TYPE    ,99$            ;;TYPE ASCIZ STRING
14042  045732  000405                        BR      98$             ;;GET OVER THE ASCIZ
14043                                  ;;99$:  .ASCIZ  /   RHDT= /
14044  045746                          98$:
14045  045746  017746  136134                MOV     @RHDT,-(SP)     ;GET READY TO TYPE RHDT CONTENTS
14046  045752  104402                        TYPOC
14047
14048
14049  045754  104401  045762                TYPE    ,101$           ;;TYPE ASCIZ STRING
14050  045760  000405                        BR      100$            ;;GET OVER THE ASCIZ
14051                                  ;;101$: .ASCIZ  /   RHSN= /
14052  045774                          100$:
14053  045774  017746  136110                MOV     @RHSN,-(SP)     ;GET READY TO TYPE RHSN CONTENTS
14054  046000  104402                        TYPOC
14055
14056  046002  005037  045064                CLR     @#PRITEM        ;CLEAR PREVIOUS ERROR ITEM
14057  046006                          TYPERR:
14058  046006  104401  001223                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
14059  046012  010046                        MOV     R0,-(SP)        ;SAVE R0
14060  046014  005000                        CLR     R0              ;PICKUP THE ITEM INDEX
14061  046016  153700  001114                BISB    @#$ITEMB,R0
14062  046022  001004                        BNE     1$              ;IF ITEM NUMBER IS ZERO, JUST
14063                                                                 ;TYPE THE PC OF THE ERROR
14064  046024  013746  001116                MOV     $ERRPC,-(SP)    ;SAVE $ERRPC FOR TYPEOUT
14065                                                                 ;ERROR ADDRESS
```

```
14066  046030  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
14067  046032  000454                        BR      10$             ;GET OUT
14068  046034  005300                1$:     DEC     R0              ;ADJUST THE INDEX SO THAT IT WILL
14069  046036  006300                        ASL     R0              ;       WORK FOR THE ERROR TABLE
14070  046040  006300                        ASL     R0
14071  046042  006300                        ASL     R0
14072  046044  062700  001226                ADD     #$ERRTB,R0      ;FORM TABLE POINTER
14073  046050  020037  045064                CMP     R0,@#PRITEM     ;WAS PREVIOUS ERROR SAME
14074  046054  001002                        BNE     13$             ;BRANCH IF NOT
14075  046056  022020                        CMP     (R0)+,(R0)+     ;POP R0 OVER EM AND DH
14076  046060  000420                        BR      5$
14077  046062  010037  045064        13$:    MOV     R0,@#PRITEM     ;SAVE NEW ERROR ITEM
14078  046066  012037  046076                MOV     (R0)+,2$        ;PICKUP "ERROR MESSAGE" POINTER
14079  046072  001404                        BEQ     3$              ;SKIP TYPEOUT IF NO POINTER
14080  046074  104401                        TYPE                    ;TYPE THE "ERROR MESSAGE"
14081  046076  000000                2$:     .WORD   0               ;"ERROR MESSAGE" POINTER GOES HERE
14082  046100  104401  001223                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
14083  046104  012037  046114        3$:     MOV     (R0)+,4$        ;PICKUP "DATA HEADER" POINTER
14084  046110  001404                        BEQ     5$              ;SKIP TYPEOUT IF 0
14085  046112  104401                        TYPE                    ;TYPE THE "DATA HEADER"
14086  046114  000000                4$:     .WORD   0               ;"DATA HEADER" POINTER GOES HERE
14087  046116  104401  001223                TYPE    ,$CRLF          ;"CARRIAGE RETURN" & "LINE FEED"
14088  046122  010146                5$:     MOV     R1,-(SP)        ;SAVE R1
14089  046124  012001                        MOV     (R0)+,R1        ;PICKUP "DATA TABLE" POINTER
14090  046126  001415                        BEQ     9$              ;BR IF NO DATA TO BE TYPED
14091  046130  012000                        MOV     (R0)+,R0        ;PICKUP "DATA FORMAT" POINTER
14092  046132  105720                6$:     TSTB    (R0)+           ;"OCTAL" OR "DECIMAL"
14093  046134  001003                        BNE     7$              ;BR IF DECIMAL
14094  046136  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
14095  046140  104402                        TYPOC                   ;GO TYPE--OCTAL ASCII(ALL DIGITS)
14096  046142  000402                        BR      8$
14097  046144                         7$:
14098  046144  013146                        MOV     @(R1)+,-(SP)    ;SAVE @(R1)+ FOR TYPEOUT
14099  046146  104405                        TYPDS                   ;GO TYPE--DECIMAL ASCII WITH SIGN
14100  046150  005711                8$:     TST     (R1)            ;IS THERE ANOTHER NUMBER?
14101  046152  001403                        BEQ     9$              ;BR IF NO
14102  046154  104401  046202                TYPE    ,11$            ;TYPE TWO(2) SPACES
14103  046160  000764                        BR      6$              ;LOOP
14104
14105  046162  012601                9$:     MOV     (SP)+,R1        ;RESTORE R1
14106  046164  012600                10$:    MOV     (SP)+,R0        ;"CARRIAGE RETURN" & "LINE FEED"
14107  046166  023737  000042  000046        CMP     @#42,@#46       ;ARE WE IN QV OR AUTO ACCEPT MODE?
14108  046174  001001                        BNE     .+4             ;BRANCH OVER HALT IF NOT
14109  046176  000000                        HALT                    ;HALT AFTER ERROR MESSAGE
14110  046200  000207                        RTS     PC              ;RETURN
14111  046202  020040      000       11$:    .ASCIZ  /  /            ;TWO(2) SPACES
14112          046206                        .EVEN
14113                                 ;*****************************************************************
14114                                 ;*****************************************************************
14115                                 .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
14116
14117                                 ;;*****************************************************************
14118                                 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
14119                                 ;*OCTAL (ASCII) NUMBER AND TYPE IT.
14120                                 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
14121                                 ;*CALL:
```

```
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

046206  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
046212  116637  000001  046431          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
046220  112637  046433          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
046224  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
046230  000406                  BR      $TYPON
046232  112737  000001  046431  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
046240  112737  000006  046433          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
046246  112737  000005  046430  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
046254  010346                  MOV     R3,-(SP)        ;;SAVE R3
046256  010446                  MOV     R4,-(SP)        ;;SAVE R4
046260  010546                  MOV     R5,-(SP)        ;;SAVE R5
046262  113704  046433          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
046266  005404                  NEG     R4
046270  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
046274  110437  046432          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
046300  113704  046431          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
046304  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
046310  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
046312  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
046314  000404                  BR      3$              ;;GO DO MSB
046316  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
046320  006105                  ROL     R5
046322  006105                  ROL     R5
046324  010503                  MOV     R5,R3
046326  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
046330  105337  046432          DECB    $OMODE          ;;TYPE THIS DIGIT?
046334  100016                  BPL     7$              ;;BR IF NO
046336  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
046342  001002                  BNE     4$              ;;TEST FOR 0
046344  005704                  TST     R4              ;;SUPPRESS THIS 0?
046346  001403                  BEQ     5$              ;;BR IF YES
046350  005204          4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
046352  052703  000060          BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
046356  052703  000040  5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
046362  110337  046426          MOVB    R3,8$           ;;SAVE FOR TYPING
046366  104401  046426          TYPE    ,8$             ;;GO TYPE THIS DIGIT
046372  105337  046430  7$:     DECB    $OCNT           ;;COUNT BY 1
046376  003347                  BGT     2$              ;;BR IF MORE TO DO
```

```
14178   046400   002402                          BLT     6$              ;;BR IF DONE
14179   046402   005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
14180   046404   000744                          BR      2$              ;;GO DO THE LAST DIGIT
14181   046406   012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
14182   046410   012604                          MOV     (SP)+,R4        ;;RESTORE R4
14183   046412   012603                          MOV     (SP)+,R3        ;;RESTORE R3
14184   046414   016666   000002  000004         MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
14185   046422   012616                          MOV     (SP)+,(SP)
14186   046424   000002                          RTI                     ;;RETURN
14187   046426      000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
14188   046427      000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
14189   046430      000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
14190   046431      000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
14191   046432   000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
14192                                   .SBTTL  TRAP DECODER
14193
14194                                   ;;*************************************************************
14195                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
14196                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
14197                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
14198                                   ;*GO TO THAT ROUTINE.
14199
14200  046434  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
14201  046436  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
14202  046442  005740                           TST     -(R0)           ;;BACKUP BY 2
14203  046444  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
14204  046446  006300                           ASL     R0              ;;POSITION FOR INDEXING
14205  046450  016000  046470                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
14206  046454  000200                           RTS     R0              ;;GO TO ROUTINE
14207
14208
14209                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
14210
14211  046456  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
14212  046460  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
14213  046466  000002                           RTI                     ;;RESTORE THE PSW
14214
14215                                   .SBTTL  TRAP TABLE
14216
14217                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
14218                                   ;*BY THE "TRAP" INSTRUCTION.
14219
14220                                   ;       ROUTINE
14221                                   ;       -------
14222  046470  046456                   $TRPAD: .WORD   $TRAP2
14223  046472  043610                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
14224  046474  046232                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
14225  046476  046206                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
14226  046500  046246                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
14227  046502  043364                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
14228
14229
14230  046504  044320                           $RDCHR  ;;CALL=RDCHR    TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
14231  046506  044410                           $RDLIN  ;;CALL=RDLIN    TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
14232  046510  044570                           $RDOCT  ;;CALL=RDOCT    TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
14233  046512  040444                           WAIT.T  ;;CALL=WAT      TRAP+11(104411) DONT ADD ABOVE THIS TRAP
14234
14235
14236
```

```
14237                                   .SBTTL  POWER DOWN AND UP ROUTINES
14238
14239                                   ;;*****************************************************************
14240                                   ;POWER DOWN ROUTINE
14241  046514  012737  046660  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
14242  046522  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
14243  046530  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
14244  046532  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
14245  046534  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
14246  046536  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
14247  046540  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
14248  046542  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
14249  046544  017746  132370                MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
14250  046550  010637  046664                MOV     SP,$SAVR6       ;;SAVE SP
14251  046554  012737  046566  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
14252  046562  000000                        HALT
14253  046564  000776                        BR      .-2             ;;HANG UP
14254
14255                                   ;;*****************************************************************
14256                                   ;POWER UP ROUTINE
14257  046566  012737  046660  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
14258  046574  013706  046664                MOV     $SAVR6,SP       ;;GET SP
14259  046600  005037  046664                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
14260  046604  005237  046664          1$:   INC     $SAVR6          ;;WAIT FOR THE INC
14261  046610  001375                        BNE     1$              ;;OF  WORD
14262  046612  012677  132322                MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
14263  046616  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
14264  046620  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
14265  046622  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
14266  046624  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
14267  046626  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
14268  046630  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
14269  046632  012737  046514  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
14270  046640  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
14271  046646  104401                        TYPE                    ;REPORT THE POWER FAILURE
14272  046650  046666          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
14273  046652  012716                        MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
14274  046654  005522          $PWRAD: .WORD   BEGIN                   ;;RESTART ADDRESS
14275  046656  000002                  RTI
14276  046660  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
14277  046662  000776                  BR      .-2                     ;;  BEFORE THE POWER DOWN WAS COMPLETE
14278  046664  000000          $SAVR6: 0                               ;;PUT THE SP HERE
14279  046666  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
14280  046674  000122
14281                                   .EVEN
```

```
14282                                   ;*****************************************************************************
14283                                   ;
14284                                   ;ERROR AND MESSAGE TABLE CONDIMENTS
14285                                   ;
14286                                   ;*****************************************************************************
14287
14288  046676  044124  020105  044122  EM1:    .ASCII  /THE RH BASE ADDRESS WAS 172040 INDICATING AN RS BUT THE DRIVE TYPE/<15>
14289  046704  041040  051501  020105
14290  046712  042101  051104  051505
14291  046720  020123  040527  020123
14292  046726  033461  030062  030064
14293  046734  044440  042116  041511
14294  046742  052101  047111  020107
14295  046750  047101  051040  020123
14296  046756  052502  020124  044124
14297  046764  020105  051104  053111
14298  046772  020105  054524  042520
14299  047000  005015
14300  047002  044504  020104  047516          .ASCIZ  /DID NOT CONTAIN 0 OR 1 OR 2 OR 3 OR 4/
14301  047010  020124  047503  052116
14302  047016  044501  020116  020060
14303  047024  051117  030440  047440
14304  047032  020122  020062  051117
14305  047040  031440  047440  020122
14306  047046  000064
14307
14308  047050  044124  020105  044122  EM2:    .ASCII  /THE RH BASE ADDRESS WAS 176700 INDICATING AN RP04,5,6 BUT THE DRIVE TYP
14309  047056  041040  051501  020105
14310  047064  042101  051104  051505
14311  047072  020123  040527  020123
14312  047100  033461  033466  030060
14313  047106  044440  042116  041511
14314  047114  052101  047111  020107
14315  047122  047101  051040  030120
14316  047130  026064  026065  020066
14317  047136  052502  020124  044124
14318  047144  020105  051104  053111
14319  047152  020105  054524  042520
14320  047160  005015
14321  047162  044504  020104  047516          .ASCIZ  /DID NOT CONTAIN 20020,24020,20021,24021,20022, OR 24022/
14322  047170  020124  047503  052116
14323  047176  044501  020116  030062
14324  047204  031060  026060  032062
14325  047212  031060  026060  030062
14326  047220  031060  026061  032062
14327  047226  031060  026061  030062
14328  047234  031060  026062  047440
14329  047242  020122  032062  031060
14330  047250  000062
14331
14332  047252  044124  020105  044122  EM3:    .ASCII  /THE RH BASE ADDRESS WAS 172440 INDICATING A TM02 BUT THE DRIVE TYPE/<15
14333  047260  041040  051501  020105
14334  047266  042101  051104  051505
14335  047274  020123  040527  020123
14336  047302  033461  032062  030064
14337  047310  044440  042116  041511
```

```
14338   047316  052101  047111  020107
14339   047324  020101  046524  031060
14340   047332  041040  052125  052040
14341   047340  042510  042040  044522
14342   047346  042526  052040  050131
14343   047354  006505     012
14344   047357     104  042111  047040          .ASCIZ  /DID NOT CONTAIN 142010/
14345   047364  052117  041440  047117
14346   047372  040524  047111  030440
14347   047400  031064  030460  000060
14348
14349   047406  044124  020105  044122  EM4:    .ASCII  /THE RH BASE ADDRESS WAS 176300 INDICATING MORE THAN ONE RH DEVICE BUT/<
14350   047414  041040  051501  020105
14351   047422  042101  051104  051505
14352   047430  020123  040527  020123
14353   047436  033461  031466  030060
14354   047444  044440  042116  041511
14355   047452  052101  047111  020107
14356   047460  047515  042522  052040
14357   047466  040510  020116  047117
14358   047474  020105  044122  042040
14359   047502  053105  041511  020105
14360   047510  052502  006524     012
14361   047515     124  042510  042040          .ASCII  /THE DRIVE TYPE DID NOT CONTAIN 0,1,2,3,4,20020,24020,/<15><12>
14362   047522  044522  042526  052040
14363   047530  050131  020105  044504
14364   047536  020104  047516  020124
14365   047544  047503  052116  044501
14366   047552  020116  026060  026061
14367   047560  026062  026063  026064
14368   047566  030062  031060  026060
14369   047574  032062  031060  026060
14370   047602  005015
14371   047604  030062  031060  026061          .ASCIZ  /20021,24021,20022,24022,142010/<15><12>
14372   047612  032062  031060  026061
14373   047620  030062  031060  026062
14374   047626  032062  031060  026062
14375   047634  032061  030062  030061
14376   047642  005015     000
14377   047645     111  042116  041511          .ASCIZ  /INDICATING NO RH DEVICE PRESENT/
14378   047652  052101  047111  020107
14379   047660  047516  051040  020110
14380   047666  042504  044526  042503
14381   047674  050040  042522  042523
14382   047702  052116     000
14383
14384   047705     124  042510  052440  EM5:    .ASCII  /THE UNIBUS TIMED OUT FOR EACH OF THE FOLLOWING ADDRESSES, INDICATING/<1
14385   047712  044516  052502  020123
14386   047720  044524  042515  020104
14387   047726  052517  020124  047506
14388   047734  020122  040505  044103
14389   047742  047440  020106  044124
14390   047750  020105  047506  046114
14391   047756  053517  047111  020107
14392   047764  042101  051104  051505
14393   047772  042523  026123  044440
```

```
14394  050000  042116  041511  052101
14395  050006  047111  006507     012
14396  050013     116  020117  044122          .ASCIZ  /NO RH ON THE SYSTEM SO ABORT PROGRAM/
14397  050020  047440  020116  044124
14398  050026  020105  054523  052123
14399  050034  046505  051440  020117
14400  050042  041101  051117  020124
14401  050050  051120  043517  040522
14402  050056  000115
14403
14404  050060  043101  042524  020122  EM6:    .ASCII  /AFTER AN RH CLEAR (RHCS2-BIT #5) RHCS2 DOES NOT HAVE ONLY IR/<15><12>
14405  050066  047101  051040  020110
14406  050074  046103  040505  020122
14407  050102  051050  041510  031123
14408  050110  041055  052111  021440
14409  050116  024465  051040  041510
14410  050124  031123  042040  042517
14411  050132  020123  047516  020124
14412  050140  040510  042526  047440
14413  050146  046116  020131  051111
14414  050154  005015
14415  050156  047101  020104  047125          .ASCIZ  /AND UNIT NUMBER/
14416  050164  052111  047040  046525
14417  050172  042502  000122
14418
14419  050176  043101  042524  020122  EM7:    .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14420  050204  046103  040505  044522
14421  050212  043516  052040  042510
14422  050220  051040  020110  047101
14423  050226  020104  051127  052111
14424  050234  047111  020107  047117
14425  050242  020105  047527  042122
14426  050250  044440  052116  020117
14427  050256  044122  041104  040440
14428  050264  042116  005015
14429  050270  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHDB TO HAVE WRONG VALUE GIVEN IN "BAD"/
14430  050276  020107  052111  041040
14431  050304  041501  026113  041440
14432  050312  052501  042523  020104
14433  050320  044122  041104  052040
14434  050326  020117  040510  042526
14435  050334  053440  047522  043516
14436  050342  053040  046101  042525
14437  050350  043440  053111  047105
14438  050356  044440  020116  041042
14439  050364  042101  000042
14440
14441  050370  043101  042524  020122  EM10:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14442  050376  046103  040505  044522
14443  050404  043516  052040  042510
14444  050412  051040  020110  047101
14445  050420  020104  051127  052111
14446  050426  047111  020107  047117
14447  050434  020105  047527  042122
14448  050442  044440  052116  020117
14449  050450  044122  041104  040440
```

```
14450  050456  042116  005015
14451  050462  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14452  050470  020107  052111  041040
14453  050476  041501  026113  041440
14454  050504  052501  042523  020104
14455  050512  044122  051503  020062
14456  050520  047524  044040  053101
14457  050526  020105  051127  047117
14458  050534  020107  040526  052514
14459  050542  020105  044507  042526
14460  050550  020116  047111  021040
14461  050556  040502  021104     000
14462
14463  050563     101  052106  051105  EM11:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14464  050570  041440  042514  051101
14465  050576  047111  020107  044124
14466  050604  020105  044122  040440
14467  050612  042116  053440  044522
14468  050620  044524  043516  047440
14469  050626  042516  053440  051117
14470  050634  020104  047111  047524
14471  050642  051040  042110  020102
14472  050650  047101  006504     012
14473  050655     122  040505  044504          .ASCIZ  /READING IT BACK, CAUSED RHCS1 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14474  050662  043516  044440  020124
14475  050670  040502  045503  020054
14476  050676  040503  051525  042105
14477  050704  051040  041510  030523
14478  050712  052040  020117  040510
14479  050720  042526  053440  047522
14480  050726  043516  053040  046101
14481  050734  042525  043440  053111
14482  050742  047105  044440  020116
14483  050750  041042  042101  000042
14484
14485  050756  043101  042524  020122  EM12:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14486  050764  046103  040505  044522
14487  050772  043516  052040  042510
14488  051000  051040  020110  047101
14489  051006  020104  051127  052111
14490  051014  047111  020107  047117
14491  051022  020105  047527  042122
14492  051030  044440  052116  020117
14493  051036  044122  041104  040440
14494  051044  042116  005015
14495  051050  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS3 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14496  051056  020107  052111  041040
14497  051064  041501  026113  041440
14498  051072  052501  042523  020104
14499  051100  044122  051503  020063
14500  051106  047524  044040  053101
14501  051114  020105  051127  047117
14502  051122  020107  040526  052514
14503  051130  020105  044507  042526
14504  051136  020116  047111  021040
14505  051144  040502  021104     000
```

```
14506
14507  051151     101  052106  051105  EM13:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14508  051156  041440  042514  051101
14509  051164  047111  020107  044124
14510  051172  020105  044122  040440
14511  051200  042116  053440  044522
14512  051206  044524  043516  047440
14513  051214  042516  053440  051117
14514  051222  020104  047111  047524
14515  051230  051040  042110  020102
14516  051236  047101  006504     012
14517  051243     122  040505  044504          .ASCIZ  /READING IT BACK, CAUSED RHBA TO HAVE WRONG VALUE GIVEN IN "BAD"/
14518  051250  043516  044440  020124
14519  051256  040502  045503  020054
14520  051264  040503  051525  042105
14521  051272  051040  041110  020101
14522  051300  047524  044040  053101
14523  051306  020105  051127  047117
14524  051314  020107  040526  052514
14525  051322  020105  044507  042526
14526  051330  020116  047111  021040
14527  051336  040502  021104     000
14528
14529  051343     101  052106  051105  EM14:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14530  051350  041440  042514  051101
14531  051356  047111  020107  044124
14532  051364  020105  044122  040440
14533  051372  042116  053440  044522
14534  051400  044524  043516  047440
14535  051406  042516  053440  051117
14536  051414  020104  047111  047524
14537  051422  051040  042110  020102
14538  051430  047101  006504     012
14539  051435     122  040505  044504          .ASCIZ  /READING IT BACK, CAUSED RHBAE TO HAVE WRONG VALUE GIVEN IN "BAD"/
14540  051442  043516  044440  020124
14541  051450  040502  045503  020054
14542  051456  040503  051525  042105
14543  051464  051040  041110  042501
14544  051472  052040  020117  040510
14545  051500  042526  053440  047522
14546  051506  043516  053040  046101
14547  051514  042525  043440  053111
14548  051522  047105  044440  020116
14549  051530  041042  042101  000042
14550
14551  051536  043101  042524  020122  EM15:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14552  051544  046103  040505  044522
14553  051552  043516  052040  042510
14554  051560  051040  020110  047101
14555  051566  020104  051127  052111
14556  051574  047111  020107  047117
14557  051602  020105  047527  042122
14558  051610  044440  052116  020117
14559  051616  044122  041104  040440
14560  051624  042116  005015
14561  051630  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHWC TO HAVE WRONG VALUE GIVEN IN "BAD"/
```

```
14562  051636  020107  052111  041040
14563  051644  041501  026113  041440
14564  051652  052501  042523  020104
14565  051660  044122  041527  052040
14566  051666  020117  040510  042526
14567  051674  053440  047522  043516
14568  051702  053040  046101  042525
14569  051710  043440  053111  047105
14570  051716  044440  020116  041042
14571  051724  042101  000042
14572
14573  051730  043101  042524  020122  EM16:   .ASCII  /AFTER CLEARING THE RH AND WRITING ONE WORD INTO RHDB AND/<15><12>
14574  051736  046103  040505  044522
14575  051744  043516  052040  042510
14576  051752  051040  020110  047101
14577  051760  020104  051127  052111
14578  051766  047111  020107  047117
14579  051774  020105  047527  042122
14580  052002  044440  052116  020117
14581  052010  044122  041104  040440
14582  052016  042116  005015
14583  052022  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14584  052030  020107  052111  041040
14585  052036  041501  026113  041440
14586  052044  052501  042523  020104
14587  052052  044122  051503  020062
14588  052060  047524  044040  053101
14589  052066  020105  051127  047117
14590  052074  020107  040526  052514
14591  052102  020105  044507  042526
14592  052110  020116  047111  021040
14593  052116  040502  021104     000
14594
14595  052123     101  052106  051105  EM17:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
14596  052130  041440  042514  051101
14597  052136  047111  020107  044124
14598  052144  020105  044122  040440
14599  052152  042116  053440  044522
14600  052160  044524  043516  052040
14601  052166  047527  053440  051117
14602  052174  020104  047111  047524
14603  052202  051040  042110  020102
14604  052210  047101  006504     012
14605  052215     122  040505  044504          .ASCIZ  /READING IT BACK, CAUSED RHDB TO HAVE WRONG VALUE GIVEN IN "BAD"/
14606  052222  043516  044440  020124
14607  052230  040502  045503  020054
14608  052236  040503  051525  042105
14609  052244  051040  042110  020102
14610  052252  047524  044040  053101
14611  052260  020105  051127  047117
14612  052266  020107  040526  052514
14613  052274  020105  044507  042526
14614  052302  020116  047111  021040
14615  052310  040502  021104     000
14616
14617  052315     101  052106  051105  EM20:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
```

```
14618   052322  041440  042514  051101
14619   052330  047111  020107  044124
14620   052336  020105  044122  040440
14621   052344  042116  053440  044522
14622   052352  044524  043516  052040
14623   052360  047527  053440  051117
14624   052366  020104  047111  047524
14625   052374  051040  042110  020102
14626   052402  047101  006504     012
14627   052407     122  040505  044504          .ASCIZ  /READING ONE BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14628   052414  043516  047440  042516
14629   052422  041040  041501  026113
14630   052430  041440  052501  042523
14631   052436  020104  044122  051503
14632   052444  020062  047524  044040
14633   052452  053101  020105  051127
14634   052460  047117  020107  040526
14635   052466  052514  020105  044507
14636   052474  042526  020116  047111
14637   052502  021040  040502  021104
14638   052510     000
14639
14640   052511     101  052106  051105  EM21:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
14641   052516  041440  042514  051101
14642   052524  047111  020107  044124
14643   052532  020105  044122  040440
14644   052540  042116  053440  044522
14645   052546  044524  043516  052040
14646   052554  047527  053440  051117
14647   052562  020104  047111  047524
14648   052570  051040  042110  020102
14649   052576  047101  006504     012
14650   052603     122  040505  044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHDB TO HAVE WRONG VALUE GIVEN IN "BAD"/
14651   052610  043516  044440  020124
14652   052616  040502  045503  052040
14653   052624  044527  042503  020054
14654   052632  040503  051525  042105
14655   052640  051040  042110  020102
14656   052646  047524  044040  053101
14657   052654  020105  051127  047117
14658   052662  020107  040526  052514
14659   052670  020105  044507  042526
14660   052676  020116  047111  021040
14661   052704  040502  021104     000
14662
14663   052711     101  052106  051105  EM22:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
14664   052716  041440  042514  051101
14665   052724  047111  020107  044124
14666   052732  020105  044122  040440
14667   052740  042116  053440  044522
14668   052746  044524  043516  052040
14669   052754  047527  053440  051117
14670   052762  020104  047111  047524
14671   052770  051040  042110  020102
14672   052776  047101  006504     012
14673   053003     122  040505  044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
```

```
14674   053010  043516  044440  020124
14675   053016  040502  045503  052040
14676   053024  044527  042503  020054
14677   053032  040503  051525  042105
14678   053040  051040  041510  031123
14679   053046  052040  020117  040510
14680   053054  042526  053440  047522
14681   053062  043516  053040  046101
14682   053070  042525  043440  053111
14683   053076  047105  044440  020116
14684   053104  041042  042101  000042
14685
14686   053112  043101  042524  020122  EM23:   .ASCII  /AFTER CLEARING THE RH AND WRITING EIGHT WORD INTO RHDB AND/<15><12>
14687   053120  046103  040505  044522
14688   053126  043516  052040  042510
14689   053134  051040  020110  047101
14690   053142  020104  051127  052111
14691   053150  047111  020107  044505
14692   053156  044107  020124  047527
14693   053164  042122  044440  052116
14694   053172  020117  044122  041104
14695   053200  040440  042116  005015
14696   053206  042522  042101  047111          .ASCIZ  /READING IT BACK, CAUSED RHCS2 TO HAVE WRONG VALUE GIVEN IN "BAD"/
14697   053214  020107  052111  041040
14698   053222  041501  026113  041440
14699   053230  052501  042523  020104
14700   053236  044122  051503  020062
14701   053244  047524  044040  053101
14702   053252  020105  051127  047117
14703   053260  020107  040526  052514
14704   053266  020105  044507  042526
14705   053274  020116  047111  021040
14706   053302  040502  021104     000
14707
14708   053307     124  042510  051040  EM24:   .ASCII  /THE RH WAS CLEARED AND A PATTERN OF 8 WORDS WERE WRITTEN INTO RHDB/<15>
14709   053314  020110  040527  020123
14710   053322  046103  040505  042522
14711   053330  020104  047101  020104
14712   053336  020101  040520  052124
14713   053344  051105  020116  043117
14714   053352  034040  053440  051117
14715   053360  051504  053440  051105
14716   053366  020105  051127  052111
14717   053374  042524  020116  047111
14718   053402  047524  051040  042110
14719   053410  006502     012
14720   053413     122  040505  044504          .ASCII  /READING RHDB FOR THE "N" TH. TIME GAVE WRONG VALUE IN RHDB/<15><12>
14721   053420  043516  051040  042110
14722   053426  020102  047506  020122
14723   053434  044124  020105  047042
14724   053442  020042  044124  020056
14725   053450  044524  042515  043440
14726   053456  053101  020105  051127
14727   053464  047117  020107  040526
14728   053472  052514  020105  047111
14729   053500  051040  042110  006502
```

```
14730  053506      012
14731  053507      042  021116  044440          .ASCIZ  /"N" IS GIVEN IN WORD NO/
14732  053514  020123  044507  042526
14733  053522  020116  047111  053440
14734  053530  051117  020104  047516
14735  053536      000
14736
14737  053537      124  042510  051040  EM25:   .ASCII  /THE RH WAS CLEARED AND A PATTERN OF 8 WORDS WERE WRITTEN INTO RHDB/<15>
14738  053544  020110  040527  020123
14739  053552  046103  040505  042522
14740  053560  020104  047101  020104
14741  053566  020101  040520  052124
14742  053574  051105  020116  043117
14743  053602  034040  053440  051117
14744  053610  051504  053440  051105
14745  053616  020105  051127  052111
14746  053624  042524  020116  047111
14747  053632  047524  051040  042110
14748  053640  006502      012
14749  053643      101  052106  051105          .ASCIZ  /AFTER READING ALL 8 WORDS, FOLLOWING REGISTER CONTAINED WRONG VALUE GIV
14750  053650  051040  040505  044504
14751  053656  043516  040440  046114
14752  053664  034040  053440  051117
14753  053672  051504  020054  047506
14754  053700  046114  053517  047111
14755  053706  020107  042522  044507
14756  053714  052123  051105  041440
14757  053722  047117  040524  047111
14758  053730  042105  053440  047522
14759  053736  043516  053040  046101
14760  053744  042525  043440  053111
14761  053752  047105  044440  020116
14762  053760  041042  042101  000042
14763
14764  053766  042523  052124  047111  EM33:   .ASCIZ  /SETTING RH CLEAR (RHCS2-BIT #5) CAUSED ERROR REGISTER 1 TO HAVE WRONG V
14765  053774  020107  044122  041440
14766  054002  042514  051101  024040
14767  054010  044122  051503  026462
14768  054016  044502  020124  032443
14769  054024  020051  040503  051525
14770  054032  042105  042440  051122
14771  054040  051117  051040  043505
14772  054046  051511  042524  020122
14773  054054  020061  047524  044040
14774  054062  053101  020105  051127
14775  054070  047117  020107  040526
14776  054076  052514  000105
14777
14778  054102  047101  051040  020110  EM34:   .ASCII  /AN RH CLEAR WAS GIVEN RHER1 WAS CHECKED TO HAVE ZERO. PAT (RHCS2-BIT #4
14779  054110  046103  040505  020122
14780  054116  040527  020123  044507
14781  054124  042526  020116  044122
14782  054132  051105  020061  040527
14783  054140  020123  044103  041505
14784  054146  042513  020104  047524
14785  054154  044040  053101  020105
```

```
 14786   054162   042532   047522   020056
 14787   054170   040520   020124   051050
 14788   054176   041510   031123   041055
 14789   054204   052111   021440   024464
 14790   054212   053440   051501   051440
 14791   054220   052105   005015
 14792   054224   047524   044440   053116          .ASCIZ  /TO INVERT PARITY CHECKING. RHER1 WAS READ BUT DID NOT CONTAIN WHAT IS I
 14793   054232   051105   020124   040520
 14794   054240   044522   054524   041440
 14795   054246   042510   045503   047111
 14796   054254   027107   051040   042510
 14797   054262   030522   053440   051501
 14798   054270   051040   040505   020104
 14799   054276   052502   020124   044504
 14800   054304   020104   047516   020124
 14801   054312   047503   052116   044501
 14802   054320   020116   044127   052101
 14803   054326   044440   020123   047111
 14804   054334   021040   047507   042117
 14805   054342   000042
 14806
 14807   054344   044122   041440   042514  EM35:   .ASCII  /RH CLEAR WAS GIVEN. RHER1 WAS CHECKED. PAT (RHCS2-BIT #4) WAS SET/<15><
 14808   054352   051101   053440   051501
 14809   054360   043440   053111   047105
 14810   054366   020056   044122   051105
 14811   054374   020061   040527   020123
 14812   054402   044103   041505   042513
 14813   054410   027104   050040   052101
 14814   054416   024040   044122   051503
 14815   054424   026462   044502   020124
 14816   054432   032043   020051   040527
 14817   054440   020123   042523   006524
 14818   054446      012
 14819   054447      101   042116   051040          .ASCIZ  /AND RHER1 WAS READ. RHCS1 SHOULD HAVE RDY, SC AND MCPE SET/
 14820   054454   042510   030522   053440
 14821   054462   051501   051040   040505
 14822   054470   027104   051040   041510
 14823   054476   030523   051440   047510
 14824   054504   046125   020104   040510
 14825   054512   042526   051040   054504
 14826   054520   020054   041523   040440
 14827   054526   042116   046440   050103
 14828   054534   020105   042523   000124
 14829
 14830   054542   044122   041440   042514  EM36:   .ASCII  /RH CLEAR WAS GIVEN. RHER1 WAS CHECKED. PAT (RHCS2-BIT #4) WAS SET/<15><
 14831   054550   051101   053440   051501
 14832   054556   043440   053111   047105
 14833   054564   020056   044122   051105
 14834   054572   020061   040527   020123
 14835   054600   044103   041505   042513
 14836   054606   027104   050040   052101
 14837   054614   024040   044122   051503
 14838   054622   026462   044502   020124
 14839   054630   032043   020051   040527
 14840   054636   020123   042523   006524
 14841   054644      012
```

```
14842   054645      122   042510   030522          .ASCII  /RHER1 WAS READ. "1" WAS WRITTEN INTO "TRE" IN RHCS1/<15><12>
14843   054652   053440   051501   051040
14844   054660   040505   027104   021040
14845   054666   021061   053440   051501
14846   054674   053440   044522   052124
14847   054702   047105   044440   052116
14848   054710   020117   052042   042522
14849   054716   020042   047111   051040
14850   054724   041510   030523   005015
14851   054732   047117   041440   042510          .ASCIZ  /ON CHECKING RHCS1 IT DID NOT CONTAIN WHAT IS IN GOOD/
14852   054740   045503   047111   020107
14853   054746   044122   051503   020061
14854   054754   052111   042040   042111
14855   054762   047040   052117   041440
14856   054770   047117   040524   047111
14857   054776   053440   040510   020124
14858   055004   051511   044440   020116
14859   055012   047507   042117      000
14860
14861   055017      122   020110   040527  EM37:   .ASCII  /RH WAS CLEARED. RHER1 WAS CHECKED. PAT (RHCS2-BIT #4) WAS SET/<15><12>
14862   055024   020123   046103   040505
14863   055032   042522   027104   051040
14864   055040   042510   030522   053440
14865   055046   051501   041440   042510
14866   055054   045503   042105   020056
14867   055062   040520   020124   051050
14868   055070   041510   031123   041055
14869   055076   052111   021440   024464
14870   055104   053440   051501   051440
14871   055112   052105   005015
14872   055116   044122   051105   020061          .ASCII  /RHER1 WAS READ. "1" WAS WRITTEN IN "TRE" IN RHCS1/<15><12>
14873   055124   040527   020123   042522
14874   055132   042101   020056   030442
14875   055140   020042   040527   020123
14876   055146   051127   052111   042524
14877   055154   020116   047111   021040
14878   055162   051124   021105   044440
14879   055170   020116   044122   051503
14880   055176   006461      012
14881   055201      101   020116   044122          .ASCII  /AN RH CLEAR WAS GIVEN BUT FOLLOWING REGISTER DID NOT/<15><12>
14882   055206   041440   042514   051101
14883   055214   053440   051501   043440
14884   055222   053111   047105   041040
14885   055230   052125   043040   046117
14886   055236   047514   044527   043516
14887   055244   051040   043505   051511
14888   055252   042524   020122   044504
14889   055260   020104   047516   006524
14890   055266      012
14891   055267      103   047117   040524          .ASCIZ  /CONTAIN WHAT IS IN "GOOD"/
14892   055274   047111   053440   040510
14893   055302   020124   051511   044440
14894   055310   020116   043442   047517
14895   055316   021104      000
14896
14897   055321      101   052106   051105  EM45:   .ASCII  /AFTER AN RH CLEAR "1" WAS WRITTEN IN THE DISK (TAPE) ADDRESS/<15><12>
```

```
 14898   055326   040440   020116   044122
 14899   055334   041440   042514   051101
 14900   055342   021040   021061   053440
 14901   055350   051501   053440   044522
 14902   055356   052124   047105   044440
 14903   055364   020116   044124   020105
 14904   055372   044504   045523   024040
 14905   055400   040524   042520   020051
 14906   055406   042101   051104   051505
 14907   055414   006523      012
 14908   055417      122   043505   051511          .ASCII  /REGISTER. PAT IN RHCS1 WAS SET. ON READING DISK (TAPE) ADDRESS/<15><12>
 14909   055424   042524   027122   050040
 14910   055432   052101   044440   020116
 14911   055440   044122   051503   020061
 14912   055446   040527   020123   042523
 14913   055454   027124   047440   020116
 14914   055462   042522   042101   047111
 14915   055470   020107   044504   045523
 14916   055476   024040   040524   042520
 14917   055504   020051   042101   051104
 14918   055512   051505   006523      012
 14919   055517      122   043505   051511          .ASCIZ  /REGISTER IT DID NOT CONTAIN "1"/
 14920   055524   042524   020122   052111
 14921   055532   042040   042111   047040
 14922   055540   052117   041440   047117
 14923   055546   040524   047111   021040
 14924   055554   021061      000
 14925
 14926   055557      101   052106   051105  EM46:   .ASCII  /AFTER AN RH CLEAR "1" WAS WRITTEN IN THE DISK (TAPE) ADDRESS REGISTER/<
 14927   055564   040440   020116   044122
 14928   055572   041440   042514   051101
 14929   055600   021040   021061   053440
 14930   055606   051501   053440   044522
 14931   055614   052124   047105   044440
 14932   055622   020116   044124   020105
 14933   055630   044504   045523   024040
 14934   055636   040524   042520   020051
 14935   055644   042101   051104   051505
 14936   055652   020123   042522   044507
 14937   055660   052123   051105   005015
 14938   055666   040520   020124   047111          .ASCII  /PAT IN RHCS1 WAS SET AFTER READING DISK (TAPE) ADDRESS REGISTER THE FOL
 14939   055674   051040   041510   030523
 14940   055702   053440   051501   051440
 14941   055710   052105   040440   052106
 14942   055716   051105   051040   040505
 14943   055724   044504   043516   042040
 14944   055732   051511   020113   052050
 14945   055740   050101   024505   040440
 14946   055746   042104   042522   051523
 14947   055754   051040   043505   051511
 14948   055762   042524   020122   044124
 14949   055770   020105   047506   046114
 14950   055776   053517   047111   006507
 14951   056004      012
 14952   056005      122   043505   051511          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 14953   056012   042524   020122   044504
```

```
 14954   056020   020104   047516   020124
 14955   056026   047503   052116   044501
 14956   056034   020116   044127   052101
 14957   056042   044440   020123   047111
 14958   056050   021040   047507   042117
 14959   056056   000042
 14960
 14961   056060   047101   051040   020110  EM54:   .ASCII  /AN RH CLEAR (RHCS2 BIT #5) WAS GIVEN. A16 (RHCS1 BIT #8) WAS SET/<15><1
 14962   056066   046103   040505   020122
 14963   056074   051050   041510   031123
 14964   056102   041040   052111   021440
 14965   056110   024465   053440   051501
 14966   056116   043440   053111   047105
 14967   056124   020056   030501   020066
 14968   056132   051050   041510   030523
 14969   056140   041040   052111   021440
 14970   056146   024470   053440   051501
 14971   056154   051440   052105   005015
 14972   056162   047117   051040   040505          .ASCIZ  /ON READING THE FOLLOWING REGISTER IT DID NOT CONTAIN WHAT IS IN "GOOD"/
 14973   056170   044504   043516   052040
 14974   056176   042510   043040   046117
 14975   056204   047514   044527   043516
 14976   056212   051040   043505   051511
 14977   056220   042524   020122   052111
 14978   056226   042040   042111   047040
 14979   056234   052117   041440   047117
 14980   056242   040524   047111   053440
 14981   056250   040510   020124   051511
 14982   056256   044440   020116   043442
 14983   056264   047517   021104      000
 14984
 14985   056271      101   020116   044122  EM56:   .ASCII  /AN RH CLEAR (RHCS2 BIT #5) WAS GIVEN. A16 (RHCS1 BIT #8) WAS SET/<15><1
 14986   056276   041440   042514   051101
 14987   056304   024040   044122   051503
 14988   056312   020062   044502   020124
 14989   056320   032443   020051   040527
 14990   056326   020123   044507   042526
 14991   056334   027116   040440   033061
 14992   056342   024040   044122   051503
 14993   056350   020061   044502   020124
 14994   056356   034043   020051   040527
 14995   056364   020123   042523   006524
 14996   056372      012
 14997   056373      101   046114   055040          .ASCIZ  /ALL ZEROS WERE WRITTEN IN RHBAE. THEN THE FOLLOWING REGISTER DID NOT HA
 14998   056400   051105   051517   053440
 14999   056406   051105   020105   051127
 15000   056414   052111   042524   020116
 15001   056422   047111   051040   041110
 15002   056430   042501   020056   044124
 15003   056436   047105   052040   042510
 15004   056444   043040   046117   047514
 15005   056452   044527   043516   051040
 15006   056460   043505   051511   042524
 15007   056466   020122   044504   020104
 15008   056474   047516   020124   040510
 15009   056502   042526   053440   040510
```

```
 15010   056510   020124   051511   044440
 15011   056516   020116   043442   047517
 15012   056524   021104      000
 15013
 15014   056527      101   020116   044122  EM60:   .ASCII  /AN RH CLEAR (RHCS2 BIT #5) WAS GIVEN. A17 (RHCS1 BIT #9) WAS SET/<15><1
 15015   056534   041440   042514   051101
 15016   056542   024040   044122   051503
 15017   056550   020062   044502   020124
 15018   056556   032443   020051   040527
 15019   056564   020123   044507   042526
 15020   056572   027116   040440   033461
 15021   056600   024040   044122   051503
 15022   056606   020061   044502   020124
 15023   056614   034443   020051   040527
 15024   056622   020123   042523   006524
 15025   056630      012
 15026   056631      117   020116   042522          .ASCIZ  /ON READING THE FOLLOWING REGISTER IT DID NOT CONTAIN WHAT IS IN "GOOD"/
 15027   056636   042101   047111   020107
 15028   056644   044124   020105   047506
 15029   056652   046114   053517   047111
 15030   056660   020107   042522   044507
 15031   056666   052123   051105   044440
 15032   056674   020124   044504   020104
 15033   056702   047516   020124   047503
 15034   056710   052116   044501   020116
 15035   056716   044127   052101   044440
 15036   056724   020123   047111   021040
 15037   056732   047507   042117   000042
 15038
 15039   056740   047101   051040   020110  EM62:   .ASCII  /AN RH CLEAR (RHCS2 BIT #8) WAS GIVEN. A17 (RHCS1 BIT #9) WAS SET/<15><1
 15040   056746   046103   040505   020122
 15041   056754   051050   041510   031123
 15042   056762   041040   052111   021440
 15043   056770   024470   053440   051501
 15044   056776   043440   053111   047105
 15045   057004   020056   030501   020067
 15046   057012   051050   041510   030523
 15047   057020   041040   052111   021440
 15048   057026   024471   053440   051501
 15049   057034   051440   052105   005015
 15050   057042   046101   020114   042532          .ASCIZ  /ALL ZEROS WERE WRITTEN IN RHBAE THEN THE FOLLOWING REGISTER DID NOT HAV
 15051   057050   047522   020123   042527
 15052   057056   042522   053440   044522
 15053   057064   052124   047105   044440
 15054   057072   020116   044122   040502
 15055   057100   020105   044124   047105
 15056   057106   052040   042510   043040
 15057   057114   046117   047514   044527
 15058   057122   043516   051040   043505
 15059   057130   051511   042524   020122
 15060   057136   044504   020104   047516
 15061   057144   020124   040510   042526
 15062   057152   053440   040510   020124
 15063   057160   051511   044440   020116
 15064   057166   047507   042117   005015
 15065   057174      000
```

```
15066
15067  057175    101   020116  044122  EM64:   .ASCII  /AN RH CLEAR (RHCS2-BIT #5) WAS GIVEN. IE (RHCS1 BIT #6) WAS SET/<15><12
15068  057202 041440   042514  051101
15069  057210 024040   044122  051503
15070  057216 026462   044502  020124
15071  057224 032443   020051  040527
15072  057232 020123   044507  042526
15073  057240 027116   044440  020105
15074  057246 051050   041510  030523
15075  057254 041040   052111  021440
15076  057262 024466   053440  051501
15077  057270 051440   052105  005015
15078  057276 047117   051040  040505         .ASCIZ  /ON READING THE FOLLOWING REGISTER IT DID NOT CONTAIN WHAT IS IN "GOOD"/
15079  057304 044504   043516  052040
15080  057312 042510   043040  046117
15081  057320 047514   044527  043516
15082  057326 051040   043505  051511
15083  057334 042524   020122  052111
15084  057342 042040   042111  047040
15085  057350 052117   041440  047117
15086  057356 040524   047111  053440
15087  057364 040510   020124  051511
15088  057372 044440   020116  043442
15089  057400 047517   021104     000
15090
15091  057405    101   020116  044122  EM66:   .ASCII  /AN RH CLEAR (RHCS2-BIT #5) WAS GIVEN. IE (RHCS1 BIT #6) WAS SET/<15><12
15092  057412 041440   042514  051101
15093  057420 024040   044122  051503
15094  057426 026462   044502  020124
15095  057434 032443   020051  040527
15096  057442 020123   044507  042526
15097  057450 027116   044440  020105
15098  057456 051050   041510  030523
15099  057464 041040   052111  021440
15100  057472 024466   053440  051501
15101  057500 051440   052105  005015
15102  057506 046101   020114  042532         .ASCIZ  /ALL ZEROS WERE WRITTEN IN RHCS3 THEN THE FOLLOWING REGISTER DID NOT CON
15103  057514 047522   020123  042527
15104  057522 042522   053440  044522
15105  057530 052124   047105  044440
15106  057536 020116   044122  051503
15107  057544 020063   044124  047105
15108  057552 052040   042510  043040
15109  057560 046117   047514  044527
15110  057566 043516   051040  043505
15111  057574 051511   042524  020122
15112  057602 044504   020104  047516
15113  057610 020124   047503  052116
15114  057616 044501   020116  044127
15115  057624 052101   044440  020123
15116  057632 047111   021040  047507
15117  057640 042117   000042
15118
15119  057644 044122   041440  042514  EM70:   .ASCII  /RH CLEAR WAS GIVEN. TWO SUCCESSIVE "GO" (RHCS1-BIT #0) WAS GIVEN/<15><1
15120  057652 051101   053440  051501
15121  057660 043440   053111  047105
```

```
 15122  057666  020056  053524  020117
 15123  057674  052523  041503  051505
 15124  057702  044523  042526  021040
 15125  057710  047507  020042  051050
 15126  057716  041510  030523  041055
 15127  057724  052111  021440  024460
 15128  057732  053440  051501  043440
 15129  057740  053111  047105  005015
 15130  057746  044527  044124  052517          .ASCII  /WITHOUT WAITING FOR THE FIRST "GO" TO COMPLETE. THE FOLLOWING REGISTER/
 15131  057754  020124  040527  052111
 15132  057762  047111  020107  047506
 15133  057770  020122  044124  020105
 15134  057776  044506  051522  020124
 15135  060004  043442  021117  052040
 15136  060012  020117  047503  050115
 15137  060020  042514  042524  020056
 15138  060026  044124  020105  047506
 15139  060034  046114  053517  047111
 15140  060042  020107  042522  044507
 15141  060050  052123  051105  005015
 15142  060056  044504  020104  047516          .ASCIZ  /DID NOT CONTAIN WHAT IS IN "GOOD"/
 15143  060064  020124  047503  052116
 15144  060072  044501  020116  044127
 15145  060100  052101  044440  020123
 15146  060106  047111  021040  047507
 15147  060114  042117  000042
 15148
 15149  060120  044122  041440  042514  EM76:   .ASCII  /RH CLEAR WAS GIVEN A 2 WORD WRITE WAS DONE FROM A LOCATION/<15><12>
 15150  060126  051101  053440  051501
 15151  060134  043440  053111  047105
 15152  060142  040440  031040  053440
 15153  060150  051117  020104  051127
 15154  060156  052111  020105  040527
 15155  060164  020123  047504  042516
 15156  060172  043040  047522  020115
 15157  060200  020101  047514  040503
 15158  060206  044524  047117  005015
 15159  060214  040524  042507  020104          .ASCII  /TAGED WRFROM AND WITH BAI (RHCS2 BIT #3) SET AT THE END/<15><12>
 15160  060222  051127  051106  046517
 15161  060230  040440  042116  053440
 15162  060236  052111  020110  040502
 15163  060244  020111  051050  041510
 15164  060252  031123  041040  052111
 15165  060260  021440  024463  051440
 15166  060266  052105  040440  020124
 15167  060274  044124  020105  047105
 15168  060302  006504      012
 15169  060305     117  020106  044124          .ASCIZ  /OF THE WRITE THE FOLLOWING REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 15170  060312  020105  051127  052111
 15171  060320  020105  044124  020105
 15172  060326  047506  046114  053517
 15173  060334  047111  020107  042522
 15174  060342  044507  052123  051105
 15175  060350  042040  042111  047040
 15176  060356  052117  041440  047117
 15177  060364  040524  047111  053440
```

```
 15178   060372   040510   020124   051511
 15179   060400   044440   020116   043442
 15180   060406   047517   021104      000
 15181
 15182   060413      122   020110   046103  EM104:  .ASCII  /RH CLEAR WAS GIVEN AN IPCK BIT SHOWN IN "IPCK" WAS SET/<15><12>
 15183   060420   040505   020122   040527
 15184   060426   020123   044507   042526
 15185   060434   020116   047101   044440
 15186   060442   041520   020113   044502
 15187   060450   020124   044123   053517
 15188   060456   020116   047111   021040
 15189   060464   050111   045503   020042
 15190   060472   040527   020123   042523
 15191   060500   006524      012
 15192   060503      132   051105   051517          .ASCII  /ZEROS WERE MOVED INTO RHDB. ON READING RHDB THE FOLLOWING/<15><12>
 15193   060510   053440   051105   020105
 15194   060516   047515   042526   020104
 15195   060524   047111   047524   051040
 15196   060532   042110   027102   047440
 15197   060540   020116   042522   042101
 15198   060546   047111   020107   044122
 15199   060554   041104   052040   042510
 15200   060562   043040   046117   047514
 15201   060570   044527   043516   005015
 15202   060576   042522   044507   052123          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 15203   060604   051105   042040   042111
 15204   060612   047040   052117   041440
 15205   060620   047117   040524   047111
 15206   060626   053440   040510   020124
 15207   060634   051511   044440   020116
 15208   060642   043442   047517   021104
 15209   060650      000
 15210
 15211   060651      122   020110   046103  EM106:  .ASCII  /RH CLEAR WAS GIVEN AN IPCK BIT SHOWN IN "IPCK" WAS SET/<15><12>
 15212   060656   040505   020122   040527
 15213   060664   020123   044507   042526
 15214   060672   020116   047101   044440
 15215   060700   041520   020113   044502
 15216   060706   020124   044123   053517
 15217   060714   020116   047111   021040
 15218   060722   050111   045503   020042
 15219   060730   040527   020123   042523
 15220   060736   006524      012
 15221   060741      132   051105   051517          .ASCII  /ZEROS WERE MOVED INTO RHDB. RHDB WAS READ. AN RH CLEAR WAS GIVEN/<15><1
 15222   060746   053440   051105   020105
 15223   060754   047515   042526   020104
 15224   060762   047111   047524   051040
 15225   060770   042110   027102   051040
 15226   060776   042110   020102   040527
 15227   061004   020123   042522   042101
 15228   061012   020056   047101   051040
 15229   061020   020110   046103   040505
 15230   061026   020122   040527   020123
 15231   061034   044507   042526   006516
 15232   061042      012
 15233   061043      124   042510   043040          .ASCIZ  /THE FOLLOWING REGISTER DID NOT CONTAIN WHAT IS GIVEN IN "GOOD"/
```

```
 15234  061050  046117  047514  044527
 15235  061056  043516  051040  043505
 15236  061064  051511  042524  020122
 15237  061072  044504  020104  047516
 15238  061100  020124  047503  052116
 15239  061106  044501  020116  044127
 15240  061114  052101  044440  020123
 15241  061122  044507  042526  020116
 15242  061130  047111  021040  047507
 15243  061136  042117  000042
 15244
 15245  061142  044122  041440  042514  EM114:  .ASCII  /RH CLEAR WAS GIVEN.  A WRITE CHECK/<15><12>
 15246  061150  051101  053440  051501
 15247  061156  043440  053111  047105
 15248  061164  020056  040440  053440
 15249  061172  044522  042524  041440
 15250  061200  042510  045503  005015
 15251  061206  040527  020123  047504          .ASCIZ  /WAS DONE. THE FOLLOWING REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
 15252  061214  042516  020056  044124
 15253  061222  020105  047506  046114
 15254  061230  053517  047111  020107
 15255  061236  042522  044507  052123
 15256  061244  051105  042040  042111
 15257  061252  047040  052117  041440
 15258  061260  047117  040524  047111
 15259  061266  053440  040510  020124
 15260  061274  051511  044440  020116
 15261  061302  043442  047517  021104
 15262  061310     000
 15263
 15264  061311     122  020110  046103  EM116:  .ASCII  /RH CLEAR WAS GIVEN.  A WRITE CHECK/<15><12>
 15265  061316  040505  020122  040527
 15266  061324  020123  044507  042526
 15267  061332  027116  020040  020101
 15268  061340  051127  052111  020105
 15269  061346  044103  041505  006513
 15270  061354     012
 15271  061355     127  051501  042040          .ASCII  /WAS DONE THEN ANOTHER RH CLEAR WAS GIVEN. THE FOLLOWING REGISTER/<15><1
 15272  061362  047117  020105  044124
 15273  061370  047105  040440  047516
 15274  061376  044124  051105  051040
 15275  061404  020110  046103  040505
 15276  061412  020122  040527  020123
 15277  061420  044507  042526  027116
 15278  061426  052040  042510  043040
 15279  061434  046117  047514  044527
 15280  061442  043516  051040  043505
 15281  061450  051511  042524  006522
 15282  061456     012
 15283  061457     104  042111  047040          .ASCIZ  /DID NOT CONTAIN WHAT IS IN "GOOD"/
 15284  061464  052117  041440  047117
 15285  061472  040524  047111  053440
 15286  061500  040510  020124  051511
 15287  061506  044440  020116  043442
 15288  061514  047517  021104     000
 15289
```

```
15290   061521      117   020116   020101  EM124:  .ASCII  /ON A SILO TEST TO TEST DBL RHCS3 BIT #10 THE FOLLOWING//<15><12>
15291   061526   044523   047514   052040
15292   061534   051505   020124   047524
15293   061542   052040   051505   020124
15294   061550   041104   020114   044122
15295   061556   051503   020063   044502
15296   061564   020124   030443   020060
15297   061572   044124   020105   047506
15298   061600   046114   053517   047111
15299   061606   006507      012
15300   061611      122   043505   051511          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
15301   061616   042524   020122   044504
15302   061624   020104   047516   020124
15303   061632   047503   052116   044501
15304   061640   020116   044127   052101
15305   061646   044440   020123   047111
15306   061654   021040   047507   042117
15307   061662   000042
15308   061664   042502   047506   042522  EM132:  .ASCII  /BEFORE DATA TRANSFER COMMAND WAS TO//<15><12>
15309   061672   042040   052101   020101
15310   061700   051124   047101   043123
15311   061706   051105   041440   046517
15312   061714   040515   042116   053440
15313   061722   051501   052040   006517
15314   061730      012
15315   061731      102   020105   044507          .ASCII  /BE GIVEN THE RP DRIVE STATUS REGISTER DID//<15><12>
15316   061736   042526   020116   044124
15317   061744   020105   050122   042040
15318   061752   044522   042526   051440
15319   061760   040524   052524   020123
15320   061766   042522   044507   052123
15321   061774   051105   042040   042111
15322   062002   005015
15323   062004   047516   020124   047503          .ASCIZ  /NOT CONTAIN WHAT IS IN "GOOD"/
15324   062012   052116   044501   020116
15325   062020   044127   052101   044440
15326   062026   020123   047111   021040
15327   062034   047507   042117   000042
15328   062042   042502   047506   042522  EM133:  .ASCII  /BEFORE DATA TRANSFER COMMAND WAS TO//<15><12>
15329   062050   042040   052101   020101
15330   062056   051124   047101   043123
15331   062064   051105   041440   046517
15332   062072   040515   042116   053440
15333   062100   051501   052040   006517
15334   062106      012
15335   062107      102   020105   044507          .ASCII  /BE GIVEN THE RS DRIVE STATUS REGISTER DID//<15><12>
15336   062114   042526   020116   044124
15337   062122   020105   051522   042040
15338   062130   044522   042526   051440
15339   062136   040524   052524   020123
15340   062144   042522   044507   052123
15341   062152   051105   042040   042111
15342   062160   005015
15343   062162   047516   020124   047503          .ASCIZ  /NOT CONTAIN WHAT IS IN "GOOD"/
15344   062170   052116   044501   020116
15345   062176   044127   052101   044440
```

```
15346  062204  020123  047111  021040
15347  062212  047507  042117  000042
15348  062220  042502  047506  042522  EM134:  .ASCII  /BEFORE DATA TRANSFER COMMAND WAS TO BE GIVEN/<15><12>
15349  062226  042040  052101  020101
15350  062234  051124  047101  043123
15351  062242  051105  041440  046517
15352  062250  040515  042116  053440
15353  062256  051501  052040  020117
15354  062264  042502  043440  053111
15355  062272  047105  005015
15356  062276  044124  020105  040515          .ASCII  /THE MAG TAPE DRIVE STATUS REGISTER DID NOT/<15><12>
15357  062304  020107  040524  042520
15358  062312  042040  044522  042526
15359  062320  051440  040524  052524
15360  062326  020123  042522  044507
15361  062334  052123  051105  042040
15362  062342  042111  047040  052117
15363  062350  005015
15364  062352  047503  052116  044501          .ASCIZ  /CONTAIN WHAT IS IN RHDS1-GOOD/
15365  062360  020116  044127  052101
15366  062366  044440  020123  047111
15367  062374  051040  042110  030523
15368  062402  043455  047517  000104
15369  062410  040527  020123  040527  EM135:  .ASCII  /WAS WAITING FOR A BIT TO SET IN A REGISTER/<15><12>
15370  062416  052111  047111  020107
15371  062424  047506  020122  020101
15372  062432  044502  020124  047524
15373  062440  051440  052105  044440
15374  062446  020116  020101  042522
15375  062454  044507  052123  051105
15376  062462  005015
15377  062464  044502  020124  047111          .ASCIZ  /BIT IN QUESTION IS IN "BIT WAITED"/<15><12>
15378  062472  050440  042525  052123
15379  062500  047511  020116  051511
15380  062506  044440  020116  041042
15381  062514  052111  053440  044501
15382  062522  042524  021104  005015
15383  062530     000
15384  062531     122  043505  051511          .ASCIZ  /REGISTER ADDRESS IN QUESTION IS IN "REG. ADDR"/
15385  062536  042524  020122  042101
15386  062544  051104  051505  020123
15387  062552  047111  050440  042525
15388  062560  052123  047511  020116
15389  062566  051511  044440  020116
15390  062574  051042  043505  020056
15391  062602  042101  051104  000042
15392  062610  040527  020123  040527  EM136:  .ASCII  /WAS WAITING FOR A BIT TO RESET IN A REGISTER/<15><12>
15393  062616  052111  047111  020107
15394  062624  047506  020122  020101
15395  062632  044502  020124  047524
15396  062640  051040  051505  052105
15397  062646  044440  020116  020101
15398  062654  042522  044507  052123
15399  062662  051105  005015
15400  062666  044502  020124  047111          .ASCIZ  /BIT IN QUESTION IS IN "BIT WAITED"/<15><12>
15401  062674  050440  042525  052123
```

```
15402  062702  047511  020116  051511
15403  062710  044440  020116  041042
15404  062716  052111  053440  044501
15405  062724  042524  021104  005015
15406  062732     000
15407  062733     122  043505  051511          .ASCIZ  /REGISTER ADDRESS IN QUESTION IS IN "REG. ADDR"/
15408  062740  042524  020122  042101
15409  062746  051104  051505  020123
15410  062754  047111  050440  042525
15411  062762  052123  047511  020116
15412  062770  051511  044440  020116
15413  062776  051042  043505  020056
15414  063004  042101  051104  000042
15415  063012  047117  053440  044522  EM137:  .ASCII  /ON WRITING AND READING THE REGISTER IN "REG. ADDR"/<15><12>
15416  063020  044524  043516  040440
15417  063026  042116  051040  040505
15418  063034  044504  043516  052040
15419  063042  042510  051040  043505
15420  063050  051511  042524  020122
15421  063056  047111  021040  042522
15422  063064  027107  040440  042104
15423  063072  021122  005015
15424  063076  052111  042040  042111          .ASCIZ  /IT DID NOT CONTAIN EXPECTED VALUE/
15425  063104  047040  052117  041440
15426  063112  047117  040524  047111
15427  063120  042440  050130  041505
15428  063126  042524  020104  040526
15429  063134  052514  000105
15430  063140  044122  041440  042514  EM140:  .ASCII  /RH CLEAR WAS GIVEN AN IPCK BIT SHOWN IN "IPCK" WAS SET/<15><12>
15431  063146  051101  053440  051501
15432  063154  043440  053111  047105
15433  063162  040440  020116  050111
15434  063170  045503  041040  052111
15435  063176  051440  047510  047127
15436  063204  044440  020116  044442
15437  063212  041520  021113  053440
15438  063220  051501  051440  052105
15439  063226  005015
15440  063230  042532  047522  020123          .ASCII  /ZEROS WERE MOVED INTO RHDB. ON READING  THE FOLLOWING/<15><12>
15441  063236  042527  042522  046440
15442  063244  053117  042105  044440
15443  063252  052116  020117  044122
15444  063260  041104  020056  047117
15445  063266  051040  040505  044504
15446  063274  043516  020040  044124
15447  063302  020105  047506  046114
15448  063310  053517  047111  006507
15449  063316     012
15450  063317     122  043505  051511          .ASCIZ  /REGISTER DID NOT CONTAIN WHAT IS IN "GOOD"/
15451  063324  042524  020122  044504
15452  063332  020104  047516  020124
15453  063340  047503  052116  044501
15454  063346  020116  044127  052101
15455  063354  044440  020123  047111
15456  063362  021040  047507  042117
15457  063370  000042
```

```
 15458  063372  043101  042524  020122  EM141:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15459  063400  046103  040505  044522
 15460  063406  043516  052040  042510
 15461  063414  051040  020110  047101
 15462  063422  020104  051127  052111
 15463  063430  047111  020107  053524
 15464  063436  020117  047527  042122
 15465  063444  044440  052116  020117
 15466  063452  044122  041104  040440
 15467  063460  042116  005015
 15468  063464  042522  042101  047111          .ASCIZ  /READING IT BACK TWICE, CAUSED RHCS1 TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15469  063472  020107  052111  041040
 15470  063500  041501  020113  053524
 15471  063506  041511  026105  041440
 15472  063514  052501  042523  020104
 15473  063522  044122  051503  020061
 15474  063530  047524  044040  053101
 15475  063536  020105  051127  047117
 15476  063544  020107  040526  052514
 15477  063552  020105  044507  042526
 15478  063560  020116  047111  021040
 15479  063566  040502  021104     000
 15480  063573     101  052106  051105  EM142:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15481  063600  041440  042514  051101
 15482  063606  047111  020107  044124
 15483  063614  020105  044122  040440
 15484  063622  042116  053440  044522
 15485  063630  044524  043516  052040
 15486  063636  047527  053440  051117
 15487  063644  020104  047111  047524
 15488  063652  051040  042110  020102
 15489  063660  047101  006504     012
 15490  063665     122  040505  044504          .ASCIZ  /READING IT BACK TWICE, CAUSED RHCS3 TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15491  063672  043516  044440  020124
 15492  063700  040502  045503  052040
 15493  063706  044527  042503  020054
 15494  063714  040503  051525  042105
 15495  063722  051040  041510  031523
 15496  063730  052040  020117  040510
 15497  063736  042526  053440  047522
 15498  063744  043516  053040  046101
 15499  063752  042525  043440  053111
 15500  063760  047105  044440  020116
 15501  063766  041042  042101  000042
 15502  063774  043101  042524  020122  EM143:  .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15503  064002  046103  040505  044522
 15504  064010  043516  052040  042510
 15505  064016  051040  020110  047101
 15506  064024  020104  051127  052111
 15507  064032  047111  020107  053524
 15508  064040  020117  047527  042122
 15509  064046  044440  052116  020117
 15510  064054  044122  041104  040440
 15511  064062  042116  005015
 15512  064066  042522  042101  047111          .ASCIZ  /READING IT BACK TWICE, CAUSED RHBA TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15513  064074  020107  052111  041040
```

```
 15514   064102   041501   020113   053524
 15515   064110   041511   026105   041440
 15516   064116   052501   042523   020104
 15517   064124   044122   040502   052040
 15518   064132   020117   040510   042526
 15519   064140   053440   047522   043516
 15520   064146   053040   046101   042525
 15521   064154   043440   053111   047105
 15522   064162   044440   020116   041042
 15523   064170   042101   000042
 15524   064174   043101   042524   020122  EM144:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15525   064202   046103   040505   044522
 15526   064210   043516   052040   042510
 15527   064216   051040   020110   047101
 15528   064224   020104   051127   052111
 15529   064232   047111   020107   053524
 15530   064240   020117   047527   042122
 15531   064246   044440   052116   020117
 15532   064254   044122   041104   040440
 15533   064262   042116   005015
 15534   064266   042522   042101   047111           .ASCIZ  /READING IT BACK TWICE, CAUSED RHBAE TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15535   064274   020107   052111   041040
 15536   064302   041501   020113   053524
 15537   064310   041511   026105   041440
 15538   064316   052501   042523   020104
 15539   064324   044122   040502   020105
 15540   064332   047524   044040   053101
 15541   064340   020105   051127   047117
 15542   064346   020107   040526   052514
 15543   064354   020105   044507   042526
 15544   064362   020116   047111   021040
 15545   064370   040502   021104      000
 15546   064375      101   052106   051105  EM145:   .ASCII  /AFTER CLEARING THE RH AND WRITING TWO WORD INTO RHDB AND/<15><12>
 15547   064402   041440   042514   051101
 15548   064410   047111   020107   044124
 15549   064416   020105   044122   040440
 15550   064424   042116   053440   044522
 15551   064432   044524   043516   052040
 15552   064440   047527   053440   051117
 15553   064446   020104   047111   047524
 15554   064454   051040   042110   020102
 15555   064462   047101   006504      012
 15556   064467      122   040505   044504           .ASCIZ  /READING IT BACK TWICE, CAUSED RHWC TO HAVE WRONG VALUE GIVEN IN "BAD"/
 15557   064474   043516   044440   020124
 15558   064502   040502   045503   052040
 15559   064510   044527   042503   020054
 15560   064516   040503   051525   042105
 15561   064524   051040   053510   020103
 15562   064532   047524   044040   053101
 15563   064540   020105   051127   047117
 15564   064546   020107   040526   052514
 15565   064554   020105   044507   042526
 15566   064562   020116   047111   021040
 15567   064570   040502   021104      000
 15568
 15569   064575      101   042040   053105  EM146:   .ASCII  /A DEVICE BASE ADDRESS DID NOT TIME OUT BUT/<15><12>
```

```
 15570   064602   041511   020105   040502
 15571   064610   042523   040440   042104
 15572   064616   042522   051523   042040
 15573   064624   042111   047040   052117
 15574   064632   052040   046511   020105
 15575   064640   052517   020124   052502
 15576   064646   006524      012
 15577   064651      124   042510   041440          .ASCII  /THE CORRESPONDING VECTOR DID TIME OUT./<15><12>
 15578   064656   051117   042522   050123
 15579   064664   047117   044504   043516
 15580   064672   053040   041505   047524
 15581   064700   020122   044504   020104
 15582   064706   044524   042515   047440
 15583   064714   052125   006456      012
 15584   064721      110   052111   041440          .ASCIZ  /HIT CONTINUE TO REPEAT VECTOR TIMEOUT/
 15585   064726   047117   044524   052516
 15586   064734   020105   047524   051040
 15587   064742   050105   040505   020124
 15588   064750   042526   052103   051117
 15589   064756   052040   046511   047505
 15590   064764   052125      000
 15591   064767      101   042040   053105  EM147:  .ASCII  /A DEVICE BASE ADDRESS DID NOT TIME OUT/<15><12>
 15592   064774   041511   020105   040502
 15593   065002   042523   040440   042104
 15594   065010   042522   051523   042040
 15595   065016   042111   047040   052117
 15596   065024   052040   046511   020105
 15597   065032   052517   006524      012
 15598   065037      102   052125   047040          .ASCIZ  /BUT NO UNIT NUMBERS HAD APPROPRIATE DRIVE TYPES/
 15599   065044   020117   047125   052111
 15600   065052   047040   046525   042502
 15601   065060   051522   044040   042101
 15602   065066   040440   050120   047522
 15603   065074   051120   040511   042524
 15604   065102   042040   044522   042526
 15605   065110   052040   050131   051505
 15606   065116      000
 15607
 15608
 15609   065117      120   004503   044122  DH1:    .ASCII  /PC     RHDT/
 15610   065124   052104
 15611
 15612   065126   041520   052411   044516  DH5:    .ASCII  /PC     UNIBUS ADDRESS ON WHICH TIME OUT OCCURRED/
 15613   065134   052502   020123   042101
 15614   065142   051104   051505   020123
 15615   065150   047117   053440   044510
 15616   065156   044103   052040   046511
 15617   065164   020105   052517   020124
 15618   065172   041517   052503   051122
 15619   065200   042105
 15620
 15621   065202   041520   052011   051505  DH6:    .ASCII  /PC     TEST NO RHCS2   RHCS2/<15><12>
 15622   065210   020124   047516   051040
 15623   065216   041510   031123   051011
 15624   065224   041510   031123   005015
 15625   065232   004411   047507   042117          .ASCIZ  /               GOOD    BAD/
```

```
15626   065240  041011  042101      000
15627
15628   065245     120  004503  042524  DH7:    .ASCII  /PC      TEST    RHDB    RHDB/<15><12>
15629   065252  052123  051011  042110
15630   065260  004502  044122  041104
15631   065266  005015
15632   065270  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
15633   065276  042117  041011  042101
15634   065304     000
15635
15636   065305     120  004503  042524  DH11:   .ASCII  /PC      TEST    RHCS1   RHCS1/<15><12>
15637   065312  052123  051011  041510
15638   065320  030523  051011  041510
15639   065326  030523  005015
15640   065332  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
15641   065340  042117  041011  042101
15642   065346     000
15643
15644   065347     120  004503  042524  DH12:   .ASCII  /PC      TEST    RHCS3   RHCS3/<15><12>
15645   065354  052123  051011  041510
15646   065362  031523  051011  041510
15647   065370  031523  005015
15648   065374  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
15649   065402  042117  041011  042101
15650   065410     000
15651
15652   065411     120  004503  042524  DH13:   .ASCII  /PC      TEST    RHBA    RHBA/<15><12>
15653   065416  052123  051011  041110
15654   065424  004501  044122  040502
15655   065432  005015
15656   065434  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
15657   065442  042117  041011  042101
15658   065450     000
15659
15660   065451     120  004503  042524  DH14:   .ASCII  /PC      TEST    RHBAE   RHBAE/<15><12>
15661   065456  052123  051011  041110
15662   065464  042501  051011  041110
15663   065472  042501  005015
15664   065476  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
15665   065504  042117  041011  042101
15666   065512     000
15667
15668   065513     120  004503  042524  DH15:   .ASCII  /PC      TEST    RHWC    RHWC/<15><12>
15669   065520  052123  051011  053510
15670   065526  004503  044122  041527
15671   065534  005015
15672   065536  047011  004517  047507          .ASCIZ  /       NO      GOOD    BAD/
15673   065544  042117  041011  042101
15674   065552     000
15675
15676   065553     120  004503  042524  DH24:   .ASCII  /PC      TEST    WORD    RHDB    RHDB/<15><12>
15677   065560  052123  053411  051117
15678   065566  004504  044122  041104
15679   065574  051011  042110  006502
15680   065602     012
15681   065603     011  047516  047011          .ASCIZ  /       NO      NO      GOOD    BAD/
```

```
 15682  065610  004517  047507  042117
 15683  065616  041011  042101     000
 15684
 15685  065623     120  004503  042524  DH33:   .ASCII  /PC     TEST    RHER1   RHER1/<15><12>
 15686  065630  052123  051011  042510
 15687  065636  030522  051011  042510
 15688  065644  030522  005015
 15689  065650  047011  004517  047507           .ASCIZ  /      NO      GOOD    BAD/
 15690  065656  042117  041011  042101
 15691  065664     000
 15692
 15693  065665     120  004503  042524  DH45:   .ASCII  /PC     TEST    RHDA    RHDA/<15><12>
 15694  065672  052123  051011  042110
 15695  065700  004501  044122  040504
 15696  065706  005015
 15697  065710  047011  004517  047507           .ASCIZ  /      NO      GOOD    BAD/
 15698  065716  042117  041011  042101
 15699  065724     000
 15700
 15701  065725     120  004503  042524  DH104:  .ASCII  /PC     TEST    IPCK    RHCS3   RHCS3/<15><12>
 15702  065732  052123  044411  041520
 15703  065740  004513  044122  051503
 15704  065746  004463  044122  051503
 15705  065754  006463     012
 15706  065757     011  047516  004411           .ASCIZ  /      NO              GOOD    BAD/
 15707  065764  047507  042117  041011
 15708  065772  042101     000
 15709
 15710  065775     120  004503  042524  DH105:  .ASCII  /PC     TEST    IPCK    RHCS2   RHCS2/<15><12>
 15711  066002  052123  044411  041520
 15712  066010  004513  044122  051503
 15713  066016  004462  044122  051503
 15714  066024  006462     012
 15715  066027     011  047516  004411           .ASCIZ  /      NO              GOOD    BAD/
 15716  066034  047507  042117  041011
 15717  066042  042101     000
 15718
 15719  066045     120  004503  042524  DH106:  .ASCII  /PC     TEST    IPCK    RHCS1   RHCS1/<15><12>
 15720  066052  052123  044411  041520
 15721  066060  004513  044122  051503
 15722  066066  004461  044122  051503
 15723  066074  006461     012
 15724  066077     011  047516  004411           .ASCIZ  /      NO              GOOD    BAD/
 15725  066104  047507  042117  041011
 15726  066112  042101     000
 15727
 15728  066115     120  004503  042524  DH111:  .ASCII  /PC     TEST    IPCK    RHBA    RHBA/<15><12>
 15729  066122  052123  044411  041520
 15730  066130  004513  044122  040502
 15731  066136  051011  041110  006501
 15732  066144     012
 15733  066145     011  047516  004411           .ASCIZ  /      NO              GOOD    BAD/
 15734  066152  047507  042117  041011
 15735  066160  042101     000
 15736
 15737  066163     120  004503  042524  DH112:  .ASCII  /PC     TEST    IPCK    RHBAE   RHBAE/<15><12>
```

```
15738  066170  052123  044411  041520
15739  066176  004513  044122  040502
15740  066204  004505  044122  040502
15741  066212  006505     012
15742  066215     011  047516  004411          .ASCIZ  /       NO              GOOD    BAD/
15743  066222  047507  042117  041011
15744  066230  042101     000
15745
15746  066233     120  004503  042524  DH113:  .ASCII  /PC     TEST    IPCK    RHWC    RHWC/<15><12>
15747  066240  052123  044411  041520
15748  066246  004513  044122  041527
15749  066254  051011  053510  006503
15750  066262     012
15751  066263     011  047516  004411          .ASCIZ  /       NO              GOOD    BAD/
15752  066270  047507  042117  041011
15753  066276  042101     000
15754  066301     120  004505  042524  DH132:  .ASCII  /PE     TEST    RHDS1   RHDS1/<15><12>
15755  066306  052123  051011  042110
15756  066314  030523  051011  042110
15757  066322  030523  005015
15758  066326  047011  004517  047507          .ASCIZ  /       NO      GOOD BAD/
15759  066334  042117  041040  042101
15760  066342     000
15761  066343     120  004503  042524  DH135:  .ASCII  /PC     TEST    PC OF   BIT WAIT  REG/<15><12>
15762  066350  052123  050011  020103
15763  066356  043117  041011  052111
15764  066364  053440  044501  020124
15765  066372  051040  043505  005015
15766  066400  047011  004517  040527          .ASCIZ  /       NO      WAT     FOR       ADDR/
15767  066406  004524  047506  004522
15768  066414  020040  042101  051104
15769  066422     000
15770  066423     120  004503  042524  DH137:  .ASCII  /PC     TEST    REG.    GOOD    BAD/<15><12>
15771  066430  052123  051011  043505
15772  066436  004456  047507  042117
15773  066444  041011  042101  005015
15774  066452  047011  004517  042101          .ASCIZ  /       NO      ADDR    DATA    DATA/
15775  066460  051104  042011  052101
15776  066466  004501  040504  040524
15777  066474     000
15778
15779  066475     120  020103  020040  DH146:  .ASCIZ  /PC       BASE    VECTOR/
15780  066502  020040  041040  051501
15781  066510  020105  020040  053040
15782  066516  041505  047524  000122
15783  066524  041520  020040  020040  DH147:  .ASCIZ  /PC       BASE ADDRESS/
15784  066532  020040  040502  042523
15785  066540  040440  042104  042522
15786  066546  051523     000
15787
15788
15789          066552                          .EVEN
15790
15791  066552  001116  001126  000000  DT1:    .WORD   $ERRPC,$BDDAT,0
15792  066560  001116  004046  004050  DT5:    .WORD   $ERRPC,RSCS1,RPCS1,TMCS1,MIXCS1,0
15793  066566  004052  004054  000000
```

```
 15794   066574   001116   004656   001124  DT6:     .WORD   $ERRPC,TSTNM,$GDDAT,$BDDAT,0
 15795   066602   001126   000000
 15796   066606   001116   004656   004664  DT24:    .WORD   $ERRPC,TSTNM,SILONM,$GDDAT,$BDDAT,0
 15797   066614   001124   001126   000000
 15798   066622   001116   004656   004662  DT104:   .WORD   $ERRPC,TSTNM,IP,$GDDAT,$BDDAT,0
 15799   066630   001124   001126   000000
 15800   066636   001116   004656   004662  DT105:   .WORD   $ERRPC,TSTNM,IP,$GDDAT,$BDDAT,0
 15801   066644   001124   001126   000000
 15802   066652   001116   004656   004662  DT106:   .WORD   $ERRPC,TSTNM,IP,$GDDAT,$BDDAT,0
 15803   066660   001124   001126   000000
 15804   066666   001116   004656   004672  DT135:   .WORD   $ERRPC,TSTNM,WAITBT,WAITPC,WAITRE,0
 15805   066674   004666   004670   000000
 15806   066702   001116   004656   001122  DT137:   .WORD   $ERRPC,TSTNM,$BDADR,$GDDAT,$BDDAT,0
 15807   066710   001124   001126   000000
 15808   066716   001116   005324   005326  DT146:   .WORD   $ERRPC,TESTDV,TESTVC,0
 15809   066724   000000
 15810   066726   001116   005324   000000  DT147:   .WORD   $ERRPC,TESTDV,0
 15811
 15812
 15813
 15814   066734      000      000           DF1:     .BYTE   0,0
 15815   066736      000      000      000  DF5:     .BYTE   0,0,0,0,0
 15816   066741      000      000
 15817   066743      000      000      000  DF6:     .BYTE   0,0,0,0
 15818   066746      000
 15819   066747      000      000      000  DF24:    .BYTE   0,0,0,0,0
 15820   066752      000      000
 15821   066754      000      000      000  DF104:   .BYTE   0,0,0,0,0
 15822   066757      000      000
 15823   066761      000      000      000  DF105:   .BYTE   0,0,0,0,0
 15824   066764      000      000
 15825   066766      000      000      000  DF106:   .BYTE   0,0,0,0,0
 15826   066771      000      000
 15827   066773      000      000      000  DF135:   .BYTE   0,0,0,0,0
 15828   066776      000      000
 15829   067000      000      000      000  DF137:   .BYTE   0,0,0,0,0
 15830   067003      000      000
 15831   067005      000      000      000  DF146:   .BYTE   0,0,0
 15832   067010      000      000           DF147:   .BYTE   0,0
 15833                                               .EVEN
 15834            000001                             .END
```

```
ACL     = 000040            2774#
ACU     = 100000            2709#
AOE     = 001000            2641#
APE     = 100000            2583#
AS      = 000016            2806#
ATA     = 100000            2629#  12942
ATABLE  004674              2941#
ATTENT  005244              3025#
ATO     = 000001            2661#
AT1     = 000002            2662#
AT2     = 000004            2663#
AT3     = 000010            2664#
AT4     = 000020            2665#
AT5     = 000040            2666#
AT6     = 000100            2667#
AT7     = 000200            2668#
A16     = 000400            2603#  7363   7368
A17     = 001000            2604#  7483   7488
BA      = 000004            2800#
BAI     = 000010            2556#  7833   7859   10447  10493  11248  11294  11935  11981  12653  12699
BASEAD  004704              2948#
BASECH  042240             13378#
BASEMA  004712              2951#
BASEVC  004714              2954#
BASINX  004766              2968#
BASM    004722              2957#
BASP    004716              2955#
BASPAD  004706              2949#
BAST    004720              2956#
BASTAD  004710              2950#
BEGIN   005522              1392   3094#  14274
BEGIN2  005532              1398   3096#
BFEVEN  003000              2825#  9985   9990   10098  10103  10325  10330  10439  10444  10672  10677  10785  10790
                           11012  11017  11240  11245  11359  11364  11472  11477  11699  11704  11927  11932  12051
                           12056  12170  12175  12407  12412
BFODD   003002              2827#  10212  10217  10557  10562  10899  10904  11125  11130  11585  11590  11813  11818
                           12288  12293  12526  12531  12645  12650
BIT0    = 000001            1366#  7191   7199   7388
BIT00   = 000001            1356#  1366   4024
BIT01   = 000002            1355#  1365
BIT02   = 000004            1354#  1364
BIT03   = 000010            1353#  1363
BIT04   = 000020            1352#  1362   13233
BIT05   = 000040            1351#  1361   13233
BIT06   = 000100            1350#  1360
BIT07   = 000200            1349#  1359
BIT08   = 000400            1348#  1358   13475
BIT09   = 001000            1347#  1357   13483  13875
BIT1    = 000002            1365#  7508
BIT10   = 002000            1346#  3706   3708   3714   3740   13860
BIT11   = 004000            1345#  13490
BIT12   = 010000            1344#  3706   3714   3740
BIT13   = 020000            1343#  13867
BIT14   = 040000            1342#  3715   3741   3794   3819   3873   3899   3952   3977   4031   4057   4112   4138
                           13461
BIT15   = 100000            1341#
```

```
BIT2   = 000004          1364#
BIT3   = 000010          1363#
BIT4   = 000020          1362#   3866
BIT5   = 000040          1361#   3866
BIT6   = 000100          1360#
BIT7   = 000200          1359#   3866
BIT8   = 000400          1358#   3715    3741    3794    3819    3866    3873    3899    3952    3977    4031    4057    4112
                         4138
BIT9   = 001000          1357#   3715    3741    3794    3819    3866    3873    3899    3952    3977    4031    4057    4112
                         4138
BOT    = 000002          2614#  13233   13241   13271
BPIB   = 001400          2738#  13298   13338
BPTVEC= 000014           1373#
BSGIVA  004724           2960#
BSGIVV  004744           2964#
BUFEW   003000           2824#   9530*   9532    9763*   9765
BUFOW   003002           2826#   9064*   9066    9297*   9299
CC     = 000036          2815#
CHECK   040340          12925#
CHECKC  040376          12929   12933#
CHECKT  040362          12930#
CH7    = 010000          2748#
CLDISK  040306           3702    3782    3860    3940    4018    4099    4184    4373    4549    4817    5211    5631    6053
                         6474    6911    7183    7358    7478    7598    7719    7826    7971    8046    8207    8282    8444
                         8584    8746    8886    9051    9120    9284    9353    9517    9586    9750    9819    9979   10092
                        10206   10319   10433   10551   10666   10779   10893   11006   11119   11234   11353   11466   11579
                        11693   11807   11921   12041   12160   12278   12397   12516   12635   12909#
CLR    = 000040          2558#   3717    3743    3796    3821    3875    3901    3954    3979    4033    4059    4114    4140
                         7011   12909
CLRBIT  004654           2926#   3708*   3721    3748    3866*   3879    3906    4024*   4037    4064    4105*   4118    4145
CMNDRP  041556           3356   13174#
CMNDRS  041466           3374   13126#
CMNDTM  041674           3336   12044   12163   12281   12400   12519   12638   13229#
COMAND  004642           2917#  13126*  13147   13174*  13201   13229*  13296   13346
COMND   004640           2916#   3668*   7729    7835    9060    9069    9293    9302    9526    9535    9759    9768    9993
                        10106   10220   10333   10448   10565   10680   10793   10907   11020   11133   11249   11367   11480
                        11593   11707   11821   11936   12044   12060   12163   12179   12281   12297   12400   12416   12519
                        12535   12638   12655
CR     = 000015          1281#  13635   13645
CRLF   = 000200          1282#   3250   13606   13645   13903
CSF    = 000002          2696#
CSU    = 000010          2698#
CS1    = 000000          2798#
CS2    = 000010          2803#
DA     = 000006          2802#
DB     = 000022          2808#
DBL    = 002000          2578#   7779    9099    9332    9565    9798   10111   10570   10798   11138   11598   11712   12064
                        12301   12420
DC     = 000034          2814#
DCK    = 100000          2647#
DCL    = 000100          2775#
DCLEAR  004142           2893#  13280
DDISP  = 177570          1288#   1471    3125
DEN1   = 000400          2735#
DEN2   = 001000          2736#
DEN4   = 002000          2737#
```

```
DEVPNT  005300          3046#   3211    3243*   3259
DEI     = 000040        2619#
DFF20   = 000002        2615#
DFF5    = 000001        2613#
DF1     066734          1531    1543    1554    1567    15814#
DF104   066754          2211    2244    2250    2256    2262    15821#
DF105   066761          2217    2238    15823#
DF106   066766          2232    2440    15825#
DF135   066773          2410    2421    15827#
DF137   067000          2433    15829#
DF146   067005          2522    15831#
DF147   067010          2531    15832#
DF24    066747          1789    15819#
DF5     066736          1580    15815#
DF6     066743          1592    1605    1619    1633    1647    1661    1675    1689    1703    1717    1731    1745    1759
                        1773    1803    1809    1818    1827    1836    1845    1857    1871    1885    1901    1917    1923
                        1929    1935    1941    1947    1964    1981    1987    1993    1999    2005    2011    2022    2028
                        2039    2045    2056    2062    2073    2079    2090    2096    2107    2113    2126    2132    2138
                        2144    2150    2156    2169    2175    2181    2187    2193    2199    2272    2282    2293    2299
                        2305    2311    2317    2323    2333    2339    2345    2351    2357    2363    2376    2389    2399
                        2453    2468    2483    2498    2513    15817#
DH1     065117          1528    1540    1551    1564    15609#
DH104   065725          2209    2242    15701#
DH105   065775          2215    2236    15710#
DH106   066045          2230    2438    15719#
DH11    065305          1628    1807    1880    1896    1912    1976    2020    2043    2054    2077    2088    2111    2124
                        2167    2291    2337    2448    15636#
DH111   066115          2248    15728#
DH112   066163          2254    15737#
DH113   066233          2260    15746#
DH12    065347          1642    1813    1927    1991    2094    2105    2136    2179    2280    2303    2331    2463    15644#
DH13    065411          1656    1822    1933    1997    2142    2185    2309    2349    2478    15652#
DH132   066301          2371    2384    2397    15754#
DH135   066343          2404    2415    15761#
DH137   066423          2427    15770#
DH14    065451          1670    1831    1939    2003    2026    2037    2060    2071    2148    2191    2315    2355    2493
                        15660#
DH146   066475          2518    15779#
DH147   066524          2528    15783#
DH15    065513          1684    1840    1945    2009    2154    2197    2321    2361    2508    15668#
DH24    065553          1783    15676#
DH33    065623          1852    1866    15685#
DH45    065665          1959    15693#
DH5     065126          1576    15612#
DH6     065202          1587    1614    1698    1726    1754    1768    1798    1921    1985    2130    2173    2270    2297
                        2343    15621#
DH7     065245          1600    1712    1740    15628#
DIGB    = 000004        2616#
DISPLA  001142          1471#   3125*   3133*   13504*  13859*
DISPRE  000174          1389#   3133
DLT     = 100000        2569#
DL64    = 000020        2618#
DMD     = 000001        2651#
DPEEW   = 020000        2581#
DPEOW   = 040000        2582#
DPR     = 000400        2622#   12943   13129   13181   13231
```

```
DRY    = 000200          2621#  12943   13129   13181   13231   13287   13330   13355
DS     = 000012          2804#
DSWR   = 177570          1287#   1470    3124
DT     = 000026          2810#
DTE    = 010000          2644#
DTSY   = 001000          2657#
DT1      066552          1530    1542    1553    1566   15761#
DT104    066622          2210    2243    2249    2255    2261   15798#
DT105    066636          2216    2237   15800#
DT106    066652          2231    2439   15802#
DT135    066666          2409    2420   15804#
DT137    066702          2432   15806#
DT146    066716          2521   15808#
DT147    066726          2530   15810#
DT24     066606          1788   15796#
DT5      066560          1579   15792#
DT6      066574          1591    1604    1618    1632    1646    1660    1674    1688    1702    1716    1730    1744    1758
                         1772    1802    1808    1817    1826    1835    1844    1856    1870    1884    1900    1916    1922
                         1928    1934    1940    1946    1963    1980    1986    1992    1998    2004    2010    2021    2027
                         2038    2044    2055    2061    2072    2078    2089    2095    2106    2112    2125    2131    2137
                         2143    2149    2155    2168    2174    2180    2186    2192    2198    2271    2281    2292    2298
                         2304    2310    2316    2322    2332    2338    2344    2350    2356    2362    2375    2388    2398
                         2452    2467    2482    2497    2512   15794#
DVA    = 004000          2606#   3706    3707    3714    3740    4262    4704    5086    5499    5922    6343    6776    7016
                         7368    7435    7488    7555    7608    7675    7739    7840    7981    8052    8217    8288    8455
                         8522    8590    8757    8824    8892    9126    9359    9592    9825   10017   10131   10244   10357
                        10473   10590   10704   10818   10931   11044   11158   11274   11391   11504   11618   11732   11845
                        11961   12084   12202   12321   12440   12558   12679   12935
ECH    = 000100          2638#
ECI    = 004000          2722#  13199
EC1    = 000044          2818#
EC2    = 000046          2819#
EMTVEC= 000030           1376#   3109*   3110*
EM1      046676          1524   14288#
EM10     050370          1608   14441#
EM104    060413          2202    2214   15182#
EM106    060651          2220    2235    2241    2247    2253    2259   15211#
EM11     050563          1622   14463#
EM114    061142          2265    2275   15245#
EM116    061311          2285    2296    2302    2308    2314    2320   15264#
EM12     050756          1636   14485#
EM124    061521          2326    2336    2342    2348    2354    2360   15290#
EM13     051151          1650   14507#
EM132    061664          2365   15308#
EM133    062042          2378   15328#
EM134    062220          2391   15348#
EM135    062410          2401   15369#
EM136    062610          2412   15392#
EM137    063012          2423   15415#
EM14     051343          1664   14529#
EM140    063140          2437   15430#
EM141    063372          2442   15458#
EM142    063573          2457   15480#
EM143    063774          2472   15502#
EM144    064174          2487   15524#
EM145    064375          2502   15546#
```

```
EM146   064575          2515   15569#
EM147   064767          2525   15591#
EM15    051536          1678   14551#
EM16    051730          1692   14573#
EM17    052123          1706   14595#
EM2     047050          1534   14308#
EM20    052315          1720   14617#
EM21    052511          1734   14640#
EM22    052711          1748   14663#
EM23    053112          1762   14686#
EM24    053307          1776   14708#
EM25    053537          1792    1806    1812    1821    1830    1839   14737#
EM3     047252          1546   14332#
EM33    053766          1848   14764#
EM34    054102          1860   14778#
EM35    054344          1874   14807#
EM36    054542          1888   14830#
EM37    055017          1904    1920    1926    1932    1938    1944   14861#
EM4     047406          1557   14349#
EM45    055321          1950   14897#
EM46    055557          1967    1984    1990    1996    2002    2008   14926#
EM5     047705          1570   14384#
EM54    056060          2014    2025   14961#
EM56    056271          2031    2042   14985#
EM6     050060          1583   14404#
EM60    056527          2048    2059   15014#
EM62    056740          2065    2076   15039#
EM64    057175          2082    2093    2099    2110   15067#
EM66    057405         15091#
EM7     050176          1595   14419#
EM70    057644          2116    2129    2135    2141    2147    2153   15119#
EM76    060120          2159    2172    2178    2184    2190    2196   15149#
EPAR  = 000010          2730#  13299   13339
ERFLG$  005240          3019#
ERR   = 040000          2628#
ERRVEC= 000004          1369#    3122    3123*   3134*   3159*   3160*   3263*   3312*   3577*   3578*  13466   13467*  13469*
                       13472*
ER1   = 000014          2805#
ER2   = 000040          2816#
ER3   = 000042          2817#
EXT1  = 000001          2676#
EXT10 = 000010          2679#
EXT2  = 000002          2677#
EXT20 = 000020          2680#
EXT4  = 000004          2678#
EXT40 = 000040          2681#
FBAE    005466          3080#    3339*   3359*   3377*   3669
FBASEA  005426          3076#    3330*   3396    3603    3659
FC    = 000006          2801#
FCS3    005476          3081#    3341*   3361*   3379*   3670
FDEVIC  005366          3071#    3324*   3334    3344    3346    3348    3350    3352    3354    3364    3366    3368    3370
                        3372     3402    3417    3424    3431    3433*   3439    3441*   3447    3449*   3455    3457*   3463
                        3465*    3471    3473*   3479    3481*   3487    3489*   3495    3497*   3605    3614    3621    3628
                        3635
FDRIVR  005456          3079#    3336*   3356*   3374*   3668
FEN   = 000200          2702#
```

```
FER    = 000020            2636#
FIRST    005242            3020#   3146    3156*
FMT1   = 000020            2731#
FMT2   = 000040            2732#
FMT22  = 010000            2723#  13199
FMT4   = 000100            2733#
FMT8   = 000200            2734#
FORBAE   004776            2972#
FRHNM    005436            3077#   3331*   3332*   3333*   3390    3592   12763
FSLAVE   005446            3078#   3310*   3408    3611    3667
FTYPE    005406            3074#   3328*
FUNITN   005376            3073#   3326*   3327*   3509    3649    3665
FUTABL   004134            2889#
FVECTR   005416            3075#   3329*   3515    3655    3666
GIVE     005006            2979#
GNS    = ****** U          1388    3154    3219    3223    3234    3240    3246    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   12757   12767
                          12773   12779   12783   12846   12852   12870   12874   12880   12885   12891   12897   13049   13055
                          13062   13066   13070   13077   13081   13381   13387   13398   13404   13410   13414   13420   13426
                          13433   13907   13915   13923   13931   13939   13947   13955   13963   13971   13979   13987   13995
                          14003   14011   14019   14027   14035   14043   14051   14223   14224   14225   14226   14227   14230
                          14231   14232   14233
GO     = 000001            2600#   3706    3714    3740    7731    7735   13150   13177   13204   13245   13281   13304   13350
GRV    = 000010            2617#
HCE    = 000200            2639#
HCI    = 002000            2721#
HCRC   = 000400            2640#
HERADD   005516            3090#  12887
HT     = 000011            1279#  13604   13645
IAE    = 002000            2642#
IE     = 000100            2577#   2601#   7603    7608    7628
ILF    = 000001            2632#
ILLEGL   004206            2911#
ILR    = 000002            2633#
IOTVEC = 000020            1374#   3107*   3108*
IP       004662            2933#   7974*   8210*   8447*   8749*  15798   15800   15802
IPCK0  = 000001            2573#   7976
IPCK1  = 000002            2574#   8212
IPCK2  = 000004            2575#   8449
IPCK3  = 000010            2576#   8751
IR     = 000100            2559#   3787    4189    4242    4379    4431    4554    4590    4642    4683    4822    5063    5217
                           5475    5637    5898    6059    6319    6480    6751    7041    7240    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
IXE    = 004000            2706#
KIPAR0 = 172340            1432#
KIPAR1 = 172342            1433#
KIPAR2 = 172344            1434#
KIPAR3 = 172346            1435#
KIPAR4 = 172350            1436#
KIPAR5 = 172352            1437#
KIPAR6 = 172354            1438#
KIPAR7 = 172356            1439#
KIPDR0 = 172300            1421#
```

```
KIPDR1= 172302            1422#
KIPDR2= 172304            1423#
KIPDR3= 172306            1424#
KIPDR4= 172310            1425#
KIPDR5= 172312            1426#
KIPDR6= 172314            1427#
KIPDR7= 172316            1428#
LA    = 000020            2807#
LBT   = 002000            2624#  12942
LERADD  005514            3089#  12893
LF    = 000012            1280#  13639   13645
LOPCT   005000            2973#
LOPCT1  005002            2974#
LTRY    005320            3048#   3226*   3530*   3541*
MCLK  = 000002            2652#
MCPE  = 020000            2607#   3706    3714    3740
MDPE  = 000400            2562#   8023    8259    8498    8542    8800    8844
MEMERR  005520            3091#  12899
MHS   = 001000            2704#
MINX  = 000004            2653#
MIXCS1  004054            2833#  15792
MMVEC = 000250            1410#
MOL   = 010000            2626#  12943   13129   13181   13231
MPE   = 000400            2561#
MR    = 000024            2809#
MRD   = 000020            2655#
MSE   = 000020            2699#
MSTCK = 000010            2654#
MWR   = 000040            2656#
MXF   = 001000            2563#
NED   = 010000            2566#   3291
NEM   = 004000            2565#
NHS   = 002000            2705#
NML   = 000300            2739#  13298   13338
NMRHS   004764            2967#
NOGO    004644            2918#   7728*  13148   13202   13348   13460*
NOPERA  004134            2890#
NTYPNT  005362            3066#   3279*   3282    3286    3324    3548*
OCYL  = 100000            2777#
OF    = 000032            2812#
OFREV = 000200            2720#
OFSETC  004166            2903#
OF100 = 000004            2715#
OF200 = 000010            2716#
OF25  = 000001            2713#
OF400 = 000020            2717#
OF50  = 000002            2714#
OF800 = 000040            2718#
OPERSE  040640           13044#  13687
OPI   = 020000            2645#
OR    = 000200            2560#   4214    4404    4580    4590    4616    4642    4853    4862    5248    5258    5670    5680
                          6092    6102    6515    6525    7758    8498    8800    9076    9309    9542    9775   11410   11523
                         11637   11751   11864   11981
PAR   = 000010            2635#
PARE7O= 000114            3092#   3157*   3158*
PARITY  037732            3157   12867#
```

```
PAT    = 000020          2557#   6935    7195    7240
PCJSR    040336         12920#  12925*  12926*  12930*  12931*
PE     = 020000          2567#
PERMX    005264          3035#
PERMXV   005274          3040#
PERNUM   005276          3042#
PERRP    005260          3033#
PERRPV   005270          3038#
PERRS    005256          3032#   3209
PERRSV   005266          3037#
PERTU    005262          3034#
PERTUV   005272          3039#
PGE    = 002000          2564#   7758
PIP    = 020000          2627#  13249   13260   13308   13319
PIRQ   = 177772          1286#
PIRQVE= 000240           1380#
PKACK    004172          2905#  13176
PLU    = 020000          2708#
POINTT   005360          3065#   3278*
PRE    = 000020          2773#
PRESEN   004056          2834#
PRITEM   045064          3145*   3672*  13046*  13895#  14056*  14073   14077*
PROG   = 001000          2623#
PR0    = 000000          1303#
PR1    = 000040          1304#
PR2    = 000100          1305#
PR3    = 000140          1306#
PR4    = 000200          1307#
PR5    = 000240          1308#
PR6    = 000300          1309#
PR7    = 000340          1310#
PS     = 177776          1283#   1284    3138*  12754*  13045*
PSEL   = 002000          2605#
PSU    = 000001          2770#
PSW    = 177776          1284#
PUTREG   041426         13100#
PWRVEC= 000024            1375#   3113*   3114*  14241*  14242*  14251*  14257*  14269*  14270*
RDCHR  = 104406          13758   14230#
RDLIN  = 104407          13808   14231#
RDOCT  = 104410           3225    3242    3251   13072   13083   13389   13406   14232#
RDY    = 000200           2602#   3706    3707    3714    3740    4262    4704    5086    5499    5922    6343    6776    7016
                          7368    7435    7488    7555    7608    7675    7840    7981    8052    8217    8288    8455    8522
                          8590    8757    8824    8892    9126    9359    9592    9825   10017   10131   10244   10357   10473
                         10590   10704   10818   10931   11044   11158   11274   11391   11504   11618   11732   11845   11961
                         12084   12202   12321   12440   12558   12679   12935   13155   13209   13366
READAT   004160          2900#  10681   10794   10908   11021   11134   11250
READIN   004174          2906#
RECALI   004140          2892#
REFOR    004162          2901#
REINTO   004424          2915#
RELEAS   004144          2894#
RESVEC= 000010            1370#
RETCL    004170          2904#
REVRED   004200          2908#  12061   12180   12298   12417   12536   12656   13296
REVWRT   004204          2910#  11368   11481   11594   11708   11822   11937
REWIND   004176          2907#  13244
```

```
RHAS    004076  2864# 13989
RHBA    004064  2857#  4035*  4036   4041   4061*  4062   4067   4299   4471   4743   5129   5544   5967
                6388   6823   7094   7283   7725*  7834*  7900   8124   8360   8662   8964   9058*  9066*
                9198   9291*  9299*  9431   9524*  9532*  9664   9757*  9765*  9897   9990*  10103* 10217*
               10330* 10444* 10562* 10677* 10790* 10904* 11017* 11130* 11245* 11364* 11477* 11590* 11704*
               11818* 11932* 12056* 12175* 12293* 12412* 12531* 12650* 13917
RHBAE   004130  2879#  3669*  4116*  4117   4122   4142*  4143   4148   4317   4489   4761   5149   5565
                5988   6409   6845   7118   7302   7390   7408*  7414   7510   7528*  7534   7918   8146
                8382   8684   8986   9220   9453   9686   9919
RHCA    004114  2873# 13196* 13981
RHCC    004100  2865# 13106  14021
RHCCA   004116  2874#
RHCS1   004060  2855#  3270   3306*  3547*  3552*  3658   3719*  3720   3725   3745*  3746   3751   4264
                4706   5088   5501   5924   6345   6778   6962   6980*  6987   7018   7222   7363*  7370
                7437   7483*  7490   7557   7603*  7610   7677   7731*  7735*  7741   7842   7983   8054
                8219   8290   8457   8524   8592   8759   8826   8894   9128   9361   9594   9827   10015*
               10019  10129* 10133  10242* 10246  10355* 10359  10471* 10475  10588* 10592  10702* 10706
               10816* 10820  10929* 10933  11042* 11046  11156* 11160  11272* 11276  11389* 11393  11502*
               11506  11616* 11620  11730* 11734  11843* 11847  11959* 11963  12082* 12086  12200* 12204
               12319* 12323  12438* 12442  12556* 12560  12677* 12681  12769  12911* 13147* 13150* 13154
               13178* 13201* 13204* 13208  13244* 13245* 13282* 13303* 13304* 13346* 13350* 13365  13383
               13416  13933
RHCS2   004070  2861#  3276*  3291   3326   3551*  3553   3717*  3743*  3796*  3798*  3799   3803   3821*
                3823*  3824   3828   3875*  3901*  3954*  3979*  4033*  4059*  4114*  4140*  4192   4213
                4245   4382   4403   4434   4557   4579   4593   4615   4645   4686   4825   4852   4865
                5066   5220   5247   5261   5478   5640   5669   5683   5901   6062   6091   6105   6322
                6483   6514   6528   6754   6935*  7011*  7012*  7044   7195*  7243   7761   7833*  7862
                8026   8078   8262   8314   8501   8545   8616   8803   8847   8918   9079   9152   9312
                9385   9545   9618   9778   9851  10039  10153  10266  10379  10447* 10496  10612  10726
               10840  10953  11066  11180  11248* 11297  11413  11526  11640  11754  11867  11935* 11984
               12106  12224  12343  12462  12580  12653* 12702  12909* 12910* 13925
RHCS3   004132  2880#  3670*  3877*  3878   3883   3903*  3904   3909   4282   4453   4725   5109   5523
                5946   6367   6801   7070   7264   7630   7648*  7654   7781   7882   7976*  8004   8102
                8212*  8240   8338   8449*  8479   8566   8640   8751*  8781   8868   8942   9101   9176
                9334   9409   9567   9642   9800   9875  10000  10113  10227  10340  10455  10572  10687
               10800  10914  11027  11140  11256  11374  11487  11600  11714  11828  11943  12066  12185
               12303  12422  12541  12661
RHDA    004066  2859#  7191*  7201  13144*
RHDB    004102  2867#  4209*  4226   4399*  4416   4575*  4611*  4627   4667   4845*  4889   4911   4933
                4955   4977   4999   5021   5043   5241*  5286   5310   5334   5358   5382   5406   5430
                5454   5663*  5709   5733   5757   5781   5805   5829   5853   5877   6084*  6130   6154
                6178   6202   6226   6250   6274   6298   6507*  6554   6579   6604   6629   6654   6679
                6704   6729   7978*  7998   8214*  8234   8451*  8452*  8473   8520   8753*  8754*  8775
                8822  13390
RHDST   004066  2860# 13197* 13973
RHDS1   004072  2862# 13131  13183  13232  13241  13248  13259  13270  13286  13307  13318  13329  13354
               13941
RHDT    004106  2869#  3284   3299   3328  14045
RHEC1   004124  2877# 14029
RHEC2   004126  2878# 14037
RHER1   004074  2863#  6919   6941  13949
RHER2   004120  2875# 13957
RHER3   004122  2876# 13965
RHFC    004066  2858# 13301* 13345*
RHLA    004100  2866# 14013
RHMR    004104  2868# 14005
```

CROSS REFERENCE TABLE -- USER SYMBOLS


```
RHOF    004112          2872#  13199*  13997
RHSN    004110          2870#  14053
RHTC    004112          2871#   3298*  13230*  13298*  13299*  13338*  13339*
RHWC    004062          2856#   3956*   3957    3961    3981*   3982    3986    4335    4507    4779    5169    5586    6009
                        6430    6867    7142    7321    7724*   7832*   7936    8168    8404    8706    9008    9057*   9065*
                        9242    9290*   9298*   9475    9523*   9531*   9708    9756*   9764*   9941    9992*   10059   10105*
                       10173   10219*  10286   10332*  10349   10446*  10517   10564*  10632   10679*  10746   10792*  10865*
                       10906*  10973   11019*  11086   11132*  11200   11247*  11318   11366*  11433   11479*  11546   11592*
                       11660   11706*  11774   11820*  11887   11934*  12005   12058*  12126   12177*  12244   12295*  12363
                       12414*  12482   12533*  12600   12652*  12723   12912*  13104   13106   13340   13909
RMR     =  000004       2634#
RPCS1   =  004050       2831#  15792
RPVEC   =  002716       2539#   3140   13400   13407*  13422   13436*
RPVECT  =  043042       3141   13430#  13436
RSCS1   =  004046       2830#  15792
SC      =  100000       2609#   3706    3714    3740    7739    7981    8217    8522    8824   11391   11504   11618   11732
                       11845   11961
SC1     =  000100       2682#
SC10    =  001000       2685#
SC2     =  000200       2683#
SC20    =  002000       2686#
SC4     =  000400       2684#
SEECOM  004164          2902#
SERCH   004146          2895#
SETBIT  004652          2925#   3707*   3722    3747    3787*   3800    3825    3865*   3880    3905    3945*   3958    3983
                        4023*   4038    4063    4104*   4119    4144
SETSIZ  006500          3215    3259#
SILONM  004664          2934#   4885*   4907*   4929*   4951*   4973*   4995*   5017*   5039*   5282*   5306*   5330*   5354*
                        5378*   5402*   5426*   5450*   5705*   5729*   5753*   5777*   5801*   5825*   5849*   5873*   6126*
                        6150*   6174*   6198*   6222*   6246*   6270*   6294*   6550*   6575*   6600*   6625*   6650*   6675*
                        6700*   6725*   15796
SKI     =  040000       2776#
SLAVE   004660          2932#   3667*   13230
SLVPR   =  002000       2747#   3300
SN      =  000030       2811#
SND1    006102          3149    3151    3156#
SPACFD  004202          2909#  13303
SR0     =  177572       1414#
SR1     =  177574       1415#
SR2     =  177576       1416#
SR3     =  172516       1417#
SS1     007354          3343    3363    3380    3385#
SS2     006514          3262#   3532    3543
SS3     010464          3534#   3544
SS4     010506          3263    3541#   3562
SS5     010524          3294    3547#
SS6     006570          3277#   3557
SS7     006604          3281#   3550
SS8     010532          3290    3304    3548#
STACK   =  001000       1274#   3105    3699    3779    3857    3937    4015    4096    4181    4370    4546    4814    5208
                        5628    6050    6471    6908    7180    7355    7475    7595    7716    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
START   005536          3095    3098#
STKLMT= 177774          1285#
```

```
ST200     005012      2981#   3094*   3096*   3207
SWR       001140      1470#   3103    3124*   3126    3132*   3715    3741    3794    3819    3873    3899    3952    3977
                      4031    4057    4112    4138    13461   13475   13477   13483   13490   13860   13867   13872   13875
                      13896   14249   14262*
SWREG     000176      1390#   3132
SW0     = 000001      1338#
SW00    = 000001      1328#   1338
SW01    = 000002      1327#   1337
SW02    = 000004      1326#   1336
SW03    = 000010      1325#   1335
SW04    = 000020      1324#   1334
SW05    = 000040      1323#   1333
SW06    = 000100      1322#   1332    13898
SW07    = 000200      1321#   1331
SW08    = 000400      1320#   1330
SW09    = 001000      1319#   1329
SW1     = 000002      1337#
SW10    = 002000      1318#
SW11    = 004000      1317#
SW12    = 010000      1316#
SW13    = 020000      1315#
SW14    = 040000      1314#
SW15    = 100000      1313#
SW2     = 000004      1336#
SW3     = 000010      1335#
SW4     = 000020      1334#
SW5     = 000040      1333#
SW6     = 000100      1332#
SW7     = 000200      1331#
SW8     = 000400      1330#
SW9     = 001000      1329#
S1      = 000001      2727#
S2      = 000002      2728#
S4      = 000004      2729#
TBITVE= 000014        1371#
TC      = 000032      2813#
TDF     = 000040      2700#
TESTAD    040636      13043#
TESTDV    005324      3050#   3264*   3266    3271    3330    3337    3357    3375    15808   15810
TESTVC    005326      3051#   3265*   3313    3329    15808
TFOUND    005322      3049#   3261*   3309    3323    3331    3527*   3534
TIEOUT    037574      3159    3577    12843#
TIMCNT    040442      12968#  12984   12989   13004
TKVEC   = 000060      1378#   13667*  13668*
TMCS1     004052      2832#   15792
TMPSLV    005364      3068#   3297*   3298    3302    3305*   3310
TMP0      005250      3028#   3227*   3255*
TMP1      005252      3029#   3228*   3236    3248    3254*
TMP4      005254      3030#
TN      = 000067      3699#   3700    3779#   3780    3857#   3858    3937#   3938    4015#   4016    4096#   4097    4181#
                      4182    4370#   4371    4546#   4547    4814#   4815    5208#   5209    5628#   5629    6050#   6051
                      6471#   6472    6908#   6909    7180#   7181    7355#   7356    7475#   7476    7595#   7596    7716#
                      7717    7823#   7824    7968#   7969    8204#   8205    8441#   8442    8743#   8744    9048#   9049
                      9281#   9282    9514#   9515    9747#   9748    9976#   9977    10089#  10090   10203#  10204   10316#
                      10317   10430#  10431   10548#  10549   10663#  10664   10776#  10777   10890#  10891   11003#  11004
                      11116#  11117   11231#  11232   11350#  11351   11463#  11464   11576#  11577   11690#  11691   11804#
```

```
                    11805   11918#  11919   12038#  12039   12157#  12158   12275#  12276   12394#  12395   12513#  12514
                    12632#  12633
TOTALA  005246       3026#
TPVEC = 000064       1379#
TRAPVE= 000034       1377#   3111*   3112*   3139*
TRE    = 040000      2608#   3306    3547    3552    3706    3714    3740    6980    7739    7981    8217    8522    8824
TRK1   = 004000      2687#
TRK10  = 040000      2690#
TRK2   = 010000      2688#
TRK20  = 100000      2691#
TRK4   = 020000      2689#
TRTVEC= 000014       1372#
TSBAE   005140       3003#
TSBASE  005220       3012#
TSCOMD  005200       3009#
TSCS3   005160       3006#
TSDEVI  005034       2985#
TSINDX  005116       2997#
TSRHNO  005014       2983#
TSSLAV  005074       2993#
TSTNM   004656       2931#   3700*   3780*   3858*   3938*   4016*   4097*   4182*   4371*   4547*   4815*   5209*   5629*
                     6051*   6472*   6909*   7181*   7356*   7476*   7596*   7717*   7824*   7969*   8205*   8442*   8744*
                     9049*   9282*   9515*   9748*   9977*  10090*  10204*  10317*  10431*  10549*  10664*  10777*  10891*
                    11004*  11117*  11232*  11351*  11464*  11577*  11691*  11805*  11919*  12039*  12158*  12276*  12395*
                    12514*  12633*  13051   15794   15796   15798   15800   15802   15804   15806
TSTOTL  005114       2996#
TSTUNT  005506       3083#   3536*   3581    12760   12789
TST1    006140       3205#
TST10   013254       4095#
TST11   013556       4180#
TST12   014104       4369#
TST13   014406       4545#
TST14   015070       4813#
TST15   016000       5207#
TST16   016712       5627#
TST17   017624       6049#
TST2    010610       3538    3575#   12788   12824
TST20   020540       6470#
TST21   021454       6907#
TST22   022066       7179#
TST23   022362       7354#
TST24   022544       7474#
TST25   022726       7594#
TST26   023110       7715#
TST27   023312       7822#
TST3    011406       3698#
TST30   023574       7967#
TST31   024176       8203#
TST32   024600       8440#
TST33   025326       8742#
TST34   026054       9047#
TST35   026452       9280#
TST36   027050       9513#
TST37   027446       9746#
TST4    011710       3778#
TST40   030044       9975#
```

```
TST41   030272          10088#
TST42   030520          10202#
TST43   030746          10315#
TST44   031174          10429#
TST45   031430          10547#
TST46   031656          10662#
TST47   032104          10775#
TST5    012170           3856#
TST50   032332          10889#
TST51   032560          11002#
TST52   033006          11115#
TST53   033234          11230#
TST54   033470          11349#
TST55   033716          11462#
TST56   034144          11575#
TST57   034372          11689#
TST6    012472           3936#
TST60   034620          11803#
TST61   035046          11917#
TST62   035302          12037#
TST63   035540          12046   12156#
TST64   035776          12165   12274#
TST65   036234          12283   12393#
TST66   036472          12402   12512#
TST67   036730          12521   12631#
TST7    012752           4014#
TST70   037174          12640   12752#
TSUNIT  005054           2990#
TSVEC   005120           3000#
TUF     = 000100         2701#
TYPDS   = 104405         3237    3249    3391    3409    3510    3593    3612    3650   12764   12776   12813   14099   14227#
TYPE    = 104401         3152    3217    3221    3232    3238    3244    3250    3386    3392    3398    3404    3419    3426
                         3434    3442    3450    3458    3466    3474    3482    3490    3498    3505    3511    3517    3521
                         3584    3588    3595    3599    3607    3616    3623    3630    3637    3645    3651   12755   12765
                        12771   12777   12781   12811   12814   12844   12850   12868   12872   12878   12883   12889   12895
                        13047   13053   13059   13060   13064   13068   13075   13079   13379   13385   13396   13402   13409
                        13412   13418   13424   13431   13566   13609   13684   13692   13762   13765   13769   13837   13839
                        13862   13870   13903   13905   13913   13921   13929   13937   13945   13953   13961   13969   13977
                        13985   13993   14001   14009   14017   14025   14033   14041   14049   14058   14080   14082   14085
                        14087   14102   14175   14223#  14271
TYPERR  046006          13901   14057#
TYPNT   005330           3061#   3278    3288
TYPOC   = 104402         3397    3516    3604    3656   12770   12849   12854   12877   12882   12888   12894   12900   13052
                        13058   13384   13401   13417   13423   13435   13910   13918   13926   13934   13942   13950   13958
                        13966   13974   13982   13990   13998   14006   14014   14022   14030   14038   14046   14054   14066
                        14095   14224#
TYPON   = 104404       14226#
TYPOS   = 104403       14225#
UNIT    005010           2980#   3665#   4190    4243    4380    4432    4555    4591    4643    4684    4823    4863    5064
                         5218    5259    5476    5638    5681    5899    6060    6103    6320    6481    6526    6752    7012
                         7042    7241    7759    7860    8024    8076    8260    8312    8499    8543    8614    8801    8845
                         8916    9077    9150    9310    9383    9543    9616    9776    9849   10037   10151   10264   10377
                        10494   10610   10724   10838   10951   11064   11178   11295   11411   11524   11638   11752   11865
                        11982   12104   12222   12341   12460   12578   12700   12910
UNLOAD  004136           2891#
UNS     = 040000         2646#
```

```
USEVEC  005004           2975#
US1    = 000001          2553#
US2    = 000002          2554#
US4    = 000004          2555#
UWR    = 000010          2772#
VECPNT  005310           3047#   3252*   3260
VECTOR  005512           3088#   3666*
VUF    = 000002          2771#
VU30   = 010000          2707#
VV     = 000100          2620#  12928   12943   13181
WAITBT  004672           2937#  12977*  12985   12990   13006   15804
WAITPC  004666           2935#  12974*  12975*  15804
WAITRE  004670           2936#  12976*  12985   12990   12994   13006   13012   13016   15804
WAIT.T  040444          12970#  14233
WAT    = 104411           4212    4402    4578    4614    4851    5246    5668    6090    6513   13153   13207   13247   13258
                         13269   13285   13306   13317   13328   13353   13364   14233#
WATO    004646           2921#  12982   13257*  13267*  13316*  13326*  13458*
WC     = 000002          2799#  13105
WCE    = 040000          2568#   9076    9309    9542    9775
WCEEW  = 004000          2579#   9565    9798
WCEOW  = 010000          2580#   9099    9332
WCF    = 000040          2637#
WCU    = 000001          2695#
WLE    = 004000          2643#
WORKBS  004770           2969#
WORKNM  004772           2970#
WORKVC  004774           2971#
WORUNT  005510           3085#   3537*   3580   12759   12786*  12789*
WRCHDT  004152           2897#
WRCHEK  004150           2896#   9070    9303    9536    9769
WRFROM  004210           2914#   7725    7834    7898    7977*   7978    8213*   8214    8450*   8451    8452    8752*   8753
                         8754    9056*   9058    9289*   9291    9522*   9524    9755*   9757
WRIDAT  004154           2898#   7730    7836    9061    9294    9527    9760    9994   10107   10221   10334   10449   10566
WRIFOR  004156           2899#
WRL    = 004000          2625#
WRTBIT  004650           2924#   3706*   3786*   3864*   3944*   4022*   4103*
WRU    = 000400          2703#
WSU    = 000004          2697#
$AUTOB  001134           1467#
$BDADR  001122           1462#   3725*   3751*   3803*   3828*   3883*   3909*   3961*   3986*   4041*   4067*   4122*   4148*
                        12938*  12945*  15806
$BDDAT  001126           1464#   3284*   3288    3299*   3300    3720*   3723    3746*   3749    3799*   3801    3824*   3826
                         3878*   3881    3904*   3907    3957*   3959    3982*   3984    4036*   4039    4062*   4065    4117*
                         4120    4143*   4146    4192*   4193    4226*   4227    4245*   4246    4264*   4265    4282*   4283
                         4299*   4300    4317*   4318    4335*   4336    4382*   4383    4416*   4417    4434*   4435    4453*
                         4454    4471*   4472    4489*   4490    4507*   4508    4557*   4558    4593*   4594    4627*   4628
                         4645*   4646    4667*   4668    4686*   4687    4706*   4707    4725*   4726    4743*   4744    4761*
                         4762    4779*   4780    4825*   4826    4865*   4866    4889*   4890    4911*   4912    4933*   4934
                         4955*   4956    4977*   4978    4999*   5000    5021*   5022    5043*   5044    5066*   5067    5088*
                         5089    5109*   5110    5129*   5130    5149*   5150    5169*   5170    5220*   5221    5261*   5262
                         5286*   5287    5310*   5311    5334*   5335    5358*   5359    5382*   5383    5406*   5407    5430*
                         5431    5454*   5455    5478*   5479    5501*   5502    5523*   5524    5544*   5545    5565*   5566
                         5586*   5587    5640*   5641    5683*   5684    5709*   5710    5733*   5734    5757*   5758    5781*
                         5782    5805*   5806    5829*   5830    5853*   5854    5877*   5878    5901*   5902    5924*   5925
                         5946*   5947    5967*   5968    5988*   5989    6009*   6010    6062*   6063    6105*   6106    6130*
                         6131    6154*   6155    6178*   6179    6202*   6203    6226*   6227    6250*   6251    6274*   6275
```

```
                    6298*  6299   6322*  6323   6345*  6346   6367*  6368   6388*  6389   6409*  6410   6430*
                    6431   6483*  6484   6528*  6529   6554*  6555   6579*  6580   6604*  6605   6629*  6630
                    6654*  6655   6679*  6680   6704*  6705   6729*  6730   6754*  6755   6778*  6779   6801*
                    6802   6823*  6824   6845*  6846   6867*  6868   6919*  6920   6941*  6942   6962*  6963
                    6987*  6988   7018*  7019   7044*  7045   7070*  7071   7094*  7095   7118*  7119   7142*
                    7143   7201*  7202   7222*  7223   7243*  7244   7264*  7265   7283*  7284   7302*  7303
                    7321*  7322   7370*  7371   7390*  7391   7414*  7415   7437*  7438   7490*  7491   7510*
                    7511   7534*  7535   7557*  7558   7610*  7611   7630*  7631   7654*  7655   7677*  7678
                    7741*  7742   7761*  7762   7781*  7782   7842*  7843   7862*  7863   7882*  7883   7900*
                    7901   7918*  7919   7936*  7937   7983*  7984   8004*  8005   8026*  8027   8054*  8055
                    8078*  8079   8102*  8103   8124*  8125   8146*  8147   8168*  8169   8219*  8220   8240*
                    8241   8262*  8263   8290*  8291   8314*  8315   8338*  8339   8360*  8361   8382*  8383
                    8404*  8405   8457*  8458   8479*  8480   8501*  8502   8524*  8525   8545*  8546   8566*
                    8567   8592*  8593   8616*  8617   8640*  8641   8662*  8663   8684*  8685   8706*  8707
                    8759*  8760   8781*  8782   8803*  8804   8826*  8827   8847*  8848   8868*  8869   8894*
                    8895   8918*  8919   8942*  8943   8964*  8965   8986*  8987   9008*  9009   9079*  9080
                    9101*  9102   9128*  9129   9152*  9153   9176*  9177   9198*  9199   9220*  9221   9242*
                    9243   9312*  9313   9334*  9335   9361*  9362   9385*  9386   9409*  9410   9431*  9432
                    9453*  9454   9475*  9476   9545*  9546   9567*  9568   9594*  9595   9618*  9619   9642*
                    9643   9664*  9665   9686*  9687   9708*  9709   9778*  9779   9800*  9801   9827*  9828
                    9851*  9852   9875*  9876   9897*  9898   9919*  9920   9941*  9942   10000*  10001  10019*
                   10020  10039*  10040  10059*  10060  10113*  10114  10133*  10134  10153*  10154  10173*  10174
                   10227*  10228  10246*  10247  10266*  10267  10286*  10287  10340*  10341  10359*  10360  10379*
                   10380  10399*  10400  10455*  10456  10475*  10476  10496*  10497  10517*  10518  10572*  10573
                   10592*  10593  10612*  10613  10632*  10633  10687*  10688  10706*  10707  10726*  10727  10746*
                   10747  10800*  10801  10820*  10821  10840*  10841  10860*  10861  10914*  10915  10933*  10934
                   10953*  10954  10973*  10974  11027*  11028  11046*  11047  11066*  11067  11086*  11087  11140*
                   11141  11160*  11161  11180*  11181  11200*  11201  11256*  11257  11276*  11277  11297*  11298
                   11318*  11319  11374*  11375  11393*  11394  11413*  11414  11433*  11434  11487*  11488  11506*
                   11507  11526*  11527  11546*  11547  11600*  11601  11620*  11621  11640*  11641  11660*  11661
                   11714*  11715  11734*  11735  11754*  11755  11774*  11775  11828*  11829  11847*  11848  11867*
                   11868  11887*  11888  11943*  11944  11963*  11964  11984*  11985  12005*  12006  12066*  12067
                   12086*  12087  12106*  12107  12126*  12127  12185*  12186  12204*  12205  12224*  12225  12244*
                   12245  12303*  12304  12323*  12324  12343*  12344  12363*  12364  12422*  12423  12442*  12443
                   12462*  12463  12482*  12483  12541*  12542  12560*  12561  12580*  12581  12600*  12601  12661*
                   12662  12681*  12682  12702*  12703  12723*  12724  12994*  13012*  13016*  13131*  13132  13183*
                   13184  13232*  13233*  13234  15791  15794  15796  15798  15800  15802  15806
$BELL   001216      1496*  13692  13778  13862  13883
$CHARC  044024     13611*  13621* 13628  13637* 13642*
$CKSWR= ****** U   14230
$CMTAG  001100      1450*   3100   3101   3109   3115   3116
$CM1    = 000006    1482*   1483*  1484*  1485*  1486*  1487*  1488*
$CM2    = 000014    1482*   1483*  1484*  1485*  1486*  1487*  1488*
$CM3    = 000006    1480*   1482
$CM4    = 000006    1488*   1489*  1490*  1491*  1492*  1493*  1494*
$CNTLC  044527     13684   13778*
$CNTLG  044541     13780*
$CNTLU  044534     13779*
$CRLF   001223      1498*  13059  13610  13645  13778  13842  13870  13883  14058  14082  14087
$DBLK   043600     13532  13566  13574*
$DOAGN  037550     12807  12816  12822*
$DTBL   043570     13535  13570*
$ENDAD  037540      1394  12818*
$ENDCT  037506     12809*
$ENDMG  037557     12811  12826*
$ENULL  037554     12814  12825*
```

```
$EOP    037452  12799#
$EOPCT  037500  12806#  12810
$ERFLG  001103   1453#  13448  13479  13481  13487*  13508  13857*  13883
$ERMAX  001115   1459#   3117*  13481  13503*  13508
$ERROR  044730   3109   13856#
$ERRPC  001116   1460#  13864*  13865*  13866  13883  14064  15791  15792  15794  15796  15798  15800  15802
                15804  15806  15808  15810
$ERRTB  001226   1515#  14072
$ERRTY  045066  13869  13896#
$ERTTL  001112   1457#  12775  12785*  13863*  13883
$ESCAP  001214   1495#   3116*  13502*  13878  13880  13883
$FILLC  001156   1478#  13614  13645
$FILLS  001155   1477#  13645
$GDADR  001120   1461#
$GDDAT  001124   1463#   3713*  3714*  3719  3721*  3722*  3723  3739*  3740*  3745  3747*  3748*  3749
                3792*  3793*  3798  3800*  3801  3817*  3818*  3823  3825*  3826  3871*  3872*  3877
                3879*  3880*  3881  3897*  3898*  3903  3905*  3906*  3907  3950*  3951*  3956  3958*
                3959  3975*  3976*  3981  3983*  3984  4029*  4030*  4035  4037*  4038*  4039  4055*
                4056*  4061  4063*  4064*  4065  4110*  4111*  4116  4118*  4119*  4120  4136*  4137*
                4142  4144*  4145*  4146  4189*  4190*  4193  4224*  4227  4242*  4243*  4246  4262*
                4265  4280*  4283  4297*  4300  4315*  4318  4333*  4336  4379*  4380*  4383  4414*
                4417  4431*  4432*  4435  4451*  4454  4469*  4472  4487*  4490  4505*  4508  4554*
                4555*  4558  4590*  4591*  4594  4625*  4628  4642*  4643*  4646  4665*  4668  4683*
                4684*  4687  4704*  4707  4723*  4726  4741*  4744  4759*  4762  4777*  4780  4822*
                4823*  4826  4862*  4863*  4866  4887*  4890  4909*  4912  4931*  4934  4953*  4956
                4975*  4978  4997*  5000  5019*  5022  5041*  5044  5063*  5064*  5067  5086*  5089
                5107*  5110  5127*  5130  5147*  5150  5167*  5170  5217*  5218*  5221  5258*  5259*
                5262  5284*  5287  5308*  5311  5332*  5335  5356*  5359  5380*  5383  5404*  5407
                5428*  5431  5452*  5455  5475*  5476*  5479  5499*  5502  5521*  5524  5542*  5545
                5563*  5566  5584*  5587  5637*  5638*  5641  5680*  5681*  5684  5707*  5710  5731*
                5734  5755*  5758  5779*  5782  5803*  5806  5827*  5830  5851*  5854  5875*  5878
                5898*  5899*  5902  5922*  5925  5944*  5947  5965*  5968  5986*  5989  6007*  6010
                6059*  6060*  6063  6102*  6103*  6106  6128*  6131  6152*  6155  6176*  6179  6200*
                6203  6224*  6227  6248*  6251  6272*  6275  6296*  6299  6319*  6320*  6323  6343*
                6346  6365*  6368  6386*  6389  6407*  6410  6428*  6431  6480*  6481*  6484  6525*
                6526*  6529  6552*  6555  6577*  6580  6602*  6605  6627*  6630  6652*  6655  6677*
                6680  6702*  6705  6727*  6730  6751*  6752*  6755  6776*  6779  6799*  6802  6821*
                6824  6843*  6846  6865*  6868  6917*  6920  6939*  6942  6960*  6963  6985*  6988
                7016*  7019  7041*  7042*  7045  7068*  7071  7092*  7095  7116*  7119  7140*  7143
                7199*  7202  7220*  7223  7240*  7241*  7244  7262*  7265  7281*  7284  7300*  7303
                7319*  7322  7368*  7371  7388*  7391  7412*  7415  7435*  7438  7488*  7491  7508*
                7511  7532*  7535  7555*  7558  7608*  7611  7628*  7631  7652*  7655  7675*  7678
                7739*  7742  7758*  7759*  7762  7779*  7782  7840*  7843  7859*  7860*  7863  7880*
                7883  7898*  7901  7916*  7919  7934*  7937  7981*  7984  8002*  8005  8023*  8024*
                8027  8052*  8055  8075*  8076*  8079  8100*  8103  8122*  8125  8144*  8147  8166*
                8169  8217*  8220  8238*  8241  8259*  8260*  8263  8288*  8291  8311*  8312*  8315
                8336*  8339  8358*  8361  8380*  8383  8402*  8405  8455*  8458  8477*  8480  8498*
                8499*  8502  8522*  8525  8542*  8543*  8546  8564*  8567  8590*  8593  8613*  8614*
                8617  8638*  8641  8660*  8663  8682*  8685  8704*  8707  8757*  8760  8779*  8782
                8800*  8801*  8804  8824*  8827  8844*  8845*  8848  8866*  8869  8892*  8895  8915*
                8916*  8919  8940*  8943  8962*  8965  8984*  8987  9006*  9009  9076*  9077*  9080
                9099*  9102  9126*  9129  9149*  9150*  9153  9174*  9177  9196*  9199  9218*  9221
                9240*  9243  9309*  9310*  9313  9332*  9335  9359*  9362  9382*  9383*  9386  9407*
                9410  9429*  9432  9451*  9454  9473*  9476  9542*  9543*  9546  9565*  9568  9592*
                9595  9615*  9616*  9619  9640*  9643  9662*  9665  9684*  9687  9706*  9709  9775*
                9776*  9779  9798*  9801  9825*  9828  9848*  9849*  9852  9873*  9876  9895*  9898
```

F09

CROSS REFERENCE TABLE -- USER SYMBOLS


```
                      9917*   9920    9939*   9942    9998*  10001   10017*  10020   10036*  10037*  10040   10057*  10060
                     10111*  10114   10131*  10134   10150*  10151*  10154   10171*  10174   10225*  10228   10244*  10247
                     10263*  10264*  10267   10284*  10287   10338*  10341   10357*  10360   10376*  10377*  10380   10397*
                     10400   10453*  10456   10473*  10476   10493*  10494*  10497   10515*  10518   10570*  10573   10590*
                     10593   10609*  10610*  10613   10630*  10633   10685*  10688   10704*  10707   10723*  10724*  10727
                     10744*  10747   10798*  10801   10818*  10821   10837*  10838*  10841   10858*  10861   10912*  10915
                     10931*  10934   10950*  10951*  10954   10971*  10974   11025*  11028   11044*  11047   11063*  11064*
                     11067   11084*  11087   11138*  11141   11158*  11161   11177*  11178*  11181   11198*  11201   11254*
                     11257   11274*  11277   11294*  11295*  11298   11316*  11319   11372*  11375   11391*  11394   11410*
                     11417*  11414   11431*  11434   11485*  11488   11504*  11507   11523*  11524*  11527   11544*  11547
                     11598*  11601   11618*  11621   11637*  11638*  11641   11658*  11661   11712*  11715   11732*  11735
                     11751*  11752*  11755   11772*  11775   11826*  11829   11845*  11848   11864*  11865*  11868   11885*
                     11888   11941*  11944   11961*  11964   11981*  11982*  11985   12003*  12006   12064*  12067   12084*
                     12087   12103*  12104*  12107   12124*  12127   12183*  12186   12202*  12205   12221*  12222*  12225
                     12242*  12245   12301*  12304   12321*  12324   12340*  12341*  12344   12361*  12364   12420*  12423
                     12440*  12443   12459*  12460*  12463   12480*  12483   12539*  12542   12558*  12561   12577*  12578*
                     12581   12598*  12601   12659*  12662   12679*  12682   12699*  12700*  12703   12721*  12724   13129*
                     13132   13181*  13184   13231*  13234   15794   15796   15798   15800   15802   15806
$GET42  037530       12815#
$GTSWR= ****** U     14229
$HD     = 000000      1257
$HIOCT  044726      13830*  13841#
$ICNT   001104       1454#  13494*  13495   13497*  13507
$ILLUP  046660      14241   14257   14276#
$INTAG  001135       1468#
$ITEMB  001114       1458#  13866*  13883   14061
$LF     001224       1499#  13645   13769   13778   13842   13883
$LPADR  001106       1455#   3118*  13074*  13085   13485*  13500*  13505   13507
$LPERR  001110       1456#   3119*   3711*   3737*   3790*   3815*   3869*   3895*   3948*   3973*   4027*   4053*   4108*
                     4134*  13057   13084*  13485   13501*  13507   13877
$MAIL = ****** U      3136   13500   13598   13872
$MNEW   044557      13783#
$MSWR   044546      13781#
$MXCNT  043362      13498   13507#
$NULL   001154       1476#  13616   13645
$NWTST= 000001       3180#   3182    3570#   3572    3682#   3684    3763#   3765    3840#   3842    3921#   3923    3998#
                     4000    4079#   4081    4164#   4166    4353#   4355    4524#   4526    4796#   4798    5188#   5190
                     5607#   5609    6029#   6031    6450#   6452    6888#   6890    7165#   7167    7339#   7341    7459#
                     7461    7579#   7581    7699#   7701    7798#   7800    7954#   7956    8190#   8192    8426#   8428
                     8728#   8730    9031#   9033    9264#   9266    9497#   9499    9730#   9732    9964#   9966   10077#
                    10079   10191#  10193   10304#  10306   10417#  10419   10536#  10538   10651#  10653   10764#  10766
                    10878#  10880   10991#  10993   11104#  11106   11218#  11220   11338#  11340   11451#  11453   11564#
                    11566   11678#  11680   11792#  11794   11905#  11907   12025#  12027   12144#  12146   12262#  12264
                    12381#  12383   12500#  12502   12618#  12620   12745#  12747
$OCNT   046430      14147*  14176*  14189#
$OMODE  046432      14142*  14146*  14151   14154*  14165*  14191#
$OVER   043346      13462   13478   13486   13496   13504#
$PASS   001100       1451#  12803*  12804*  12812   12825   13492   13508
$POWER  046666      14272   14279#
$PWRAD  046654      14274#
$PWRDN  046514       3113   14241#  14269
$PWRMG  046650      14272#
$PWRUP  046566      14251   14257#
$QUES   001222       1497#  13645   13762   13778   13839   13842   13883
$RDCHR  044320      13731#  14230
$RDDEC= ****** U    14233
```

```
$RDLIN  044410          13754#  14231
$RDOCT  044570          13803#  14232
$RDSZ = 000011          13747#
$REGAD  001160           1480#
$REG0   001162           1482#
$REG1   001164           1483#
$REG2   001166           1484#
$REG3   001170           1485#
$REG4   001172           1486#
$REG5   001174           1487#
$RTNAD  037552          12824#
$R2A   = ****** U       14233
$SAVRE= ****** U        14233
$SAVR6  046664          14250*  14258   14259*  14260*  14278#
$SCOPE  043104           3107   12855   12901   13457#
$SETUP= 000017           3099#   3106    3107    3109    3111    3113    3115    3116    3118    3801   13458   13688   13689
                        13720   13785   13857   13875   13882
$SS1   = 000000          3136#
$STUP  = 177777          3099#
$SVLAD  043320          13470   13499#
$SWR   = 167700          1248#   1257    1262    1263    1264    1265    1266    1267    1268    1494    1495    1496    3115
                         3116    3118    3119    3206    3576    3699    3779    3857    3937    4015    4096    4181    4370
                         4546    4814    5208    5628    6050    6471    6908    7180    7355    7475    7595    7716    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   12753   12796   12802   12817   12823   12825   13449
                        13450   13451   13452   13453   13461   13473   13475   13476   13479   13480   13481   13488   13489
                        13490   13501   13504   13507   13848   13849   13850   13851   13852   13860   13867   13872   13875
                        13883   14275
$SWRMK= 000000           1268    1269   13453   13454   13477
$TIMES  001212           1494#   3115*   3206*   3576*  12753*  12802*  13488*  13495  13498*  13507
$TKB    001146           1473#  13648   13669   13680   13701
$TKCNT  044030          13649#  13664*  13690   13707*  13738   13740*
$TKINT  044050           3144   13664#  13685
$TKQEN= 044047          13653#  13715   13743
$TKQIN  044032          13650#  13665*  13666   13713*  13714*  13715   13717*
$TKQOU  044034          13651#  13666*  13741   13742*  13743   13745*
$TKQSR  044036          13652#  13665   13717   13745
$TKS    001144           1472#  13648   13670*  13697*  13699   13705*
$TKSRV  044120          13667   13680#
$TMP0   001176           1488#
$TMP1   001200           1489#   7998*   8234*   8473*   8775*
$TMP2   001202           1490#   8520*   8822*
$TMP3   001204           1491#
$TMP4   001206           1492#
$TMP5   001210           1493#
$TN    = 000071          1248#   1257    3180    3206#   3538    3570    3576#   3682    3699#   3763    3779#   3840    3857#
                         3921    3937#   3998    4015#   4079    4096#   4164    4181#   4353    4370#   4524    4546#   4796
                         4814#   5188    5208#   5607    5628#   6029    6050#   6450    6471#   6888    6908#   7165    7180#
                         7339    7355#   7459    7475#   7579    7595#   7699    7716#   7798    7823#   7954    7968#   8190
                         8204#   8426    8441#   8728    8743#   9031    9048#   9264    9281#   9497    9514#   9730    9747#
                         9964    9976#  10077   10089#  10191   10203#  10304   10316#  10417   10430#  10536   10548#  10651
                        10663#  10764   10776#  10878   10890#  10991   11003#  11104   11116#  11218   11231#  11338   11350#
                        11451   11463#  11564   11576#  11678   11690#  11792   11804#  11905   11918#  12025   12038#  12045
                        12144   12157#  12164   12262   12275#  12282   12381   12394#  12401   12500   12513#  12520   12618
                        12632#  12639   12745   12753#
```

H09

CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0318

```
$TPB     001152            1475#  13634*  13645
$TPFLG   001157            1479#  13592   13645
$TPS     001150            1474#  13632   13645
$TRAP    046434            3111   14200#
$TRAP2   046456           14211#  14222
$TRP   = 000012           14215#  14224#  14225#  14226#  14227#  14228#  14230   14231#  14232#  14233#  14234#
$TRPAD   046470           14205   14222#
$TSTNM   001102            1452#  12801*  13448   13477   13499*  13504   13508   13859   13883
$TTYIN   044516           13755   13756   13773   13777#
$TYPBN=  ******  U        14228
$TYPDS   043364           13520#  14227
$TYPE    043610           13592#  14215   14223
$TYPEC   043760           13613   13620   13627   13632#  13633
$TYPEX   044026           13638   13640   13643#
$TYPOC   046232           14145#  14224
$TYPON   046246           14144   14147#  14226
$TYPOS   046206           14140#  14225
$XTSTR   043124           13464#
$$GET4=  000000           12817#
$OFILL   046431           14141*  14145*  14155   14190#
$40CAT=  ******  U        13461   13869
.      = 067012            1384#   1388#   1393#   1395#   1397#   1442#   1449#   1500    2823#   2827#   2914#   2915#   2960#
                           2964#   2983#   2985#   2990#   2993#   3000#   3003#   3006#   3009#   3012#   3046#   3047#   3071#
                           3073#   3074#   3075#   3076#   3077#   3078#   3079#   3080#   3081#   3104    3118    3119    3235#
                           3247#   3389#   3395#   3401#   3407#   3422#   3469#   3508#   3598#   3602#   3610#   3619#   3648#
                          12825   12829   12847#  12853#  12875#  12881#  12892#  12898#  13063#  13078#  13388#  13399#  13405#
                          13415#  13507   13508   13574#  13645   13648   13652#  13653   13654#  13777#  13778   13785   13842
                          13883   13924#  13932#  13940#  13948#  13956#  13964#  13972#  14028#  14036#  14108   14112#  14253
                          14277   15789#
```

```
ALLREG   1258#  13904  13912  13920  13928  13936  13944  13952  13960  13968  13976  13984  13992  14000  14008
        14016  14024  14032  14040  14048
BITDBL   1258#   9962  10075  10189  10302  10415  10534  10649  10762  10876  10989  11102  11216  11336  11449
        11562  11676  11790  11903  12023  12142  12260  12379  12498  12616
BTBANG   1258#   3681   3762   3839   3920   3997   4078
CHECKD   1258#
CHECKV   1258#
COMMEN      1#  1381#
COMPAR   1258#  4188  4223  4241  4261  4279  4296  4314  4332  4378  4413  4430  4450  4468  4486
         4504  4553  4589  4624  4641  4664  4682  4703  4722  4740  4758  4776  4821  4861  4886
         4908  4930  4952  4974  4996  5018  5040  5062  5085  5106  5126  5146  5166  5216  5257
         5283  5307  5331  5355  5379  5403  5427  5451  5474  5498  5520  5541  5562  5583  5636
         5679  5706  5730  5754  5778  5802  5826  5850  5874  5897  5921  5943  5964  5985  6006
         6058  6101  6127  6151  6175  6199  6223  6247  6271  6295  6318  6342  6364  6385  6406
         6427  6479  6524  6551  6576  6601  6626  6651  6676  6701  6726  6750  6775  6798  6820
         6842  6864  6916  6938  6959  6984  7015  7040  7067  7091  7115  7139  7198  7219  7239
         7261  7280  7299  7318  7367  7387  7411  7434  7487  7507  7531  7554  7607  7627  7651
         7674  7738  7757  7778  7839  7858  7879  7897  7915  7933  7980  8001  8022  8051  8074
         8099  8121  8143  8165  8216  8237  8258  8287  8310  8335  8357  8379  8401  8454  8476
         8497  8521  8541  8563  8589  8612  8637  8659  8681  8703  8756  8778  8799  8823  8843
         8865  8891  8914  8939  8961  8983  9005  9075  9098  9125  9148  9173  9195  9217  9239
         9308  9331  9358  9381  9406  9428  9450  9472  9541  9564  9591  9614  9639  9661  9683
         9705  9774  9797  9824  9847  9872  9894  9916  9938  9997  10016  10035  10056  10110  10130
        10149  10170  10224  10243  10262  10283  10337  10356  10375  10396  10452  10472  10492  10514  10569
        10589  10608  10629  10684  10703  10722  10743  10797  10817  10836  10857  10911  10930  10949  10970
        11024  11043  11062  11083  11137  11157  11176  11197  11253  11273  11293  11315  11371  11390  11409
        11430  11484  11503  11522  11543  11597  11617  11636  11657  11711  11731  11750  11771  11825  11844
        11863  11884  11940  11960  11980  12002  12063  12083  12102  12123  12182  12201  12220  12241  12300
        12320  12339  12360  12419  12439  12458  12479  12538  12557  12576  12597  12658  12678  12698  12720
        13128  13180
ENDCOM      1#  1381#
ERROR    1275#  3315  3559  3726  3752  3804  3829  3884  3910  3962  3987  4042  4068  4123  4149
         4197  4231  4250  4269  4287  4304  4322  4340  4387  4421  4439  4458  4476  4494  4512
         4562  4598  4632  4650  4672  4691  4711  4730  4748  4766  4784  4830  4870  4894  4916
         4938  4960  4982  5004  5026  5048  5071  5093  5114  5134  5154  5174  5225  5266  5291
         5315  5339  5363  5387  5411  5435  5459  5483  5506  5528  5549  5570  5591  5645  5688
         5714  5738  5762  5786  5810  5834  5858  5882  5906  5929  5951  5972  5993  6014  6067
         6110  6135  6159  6183  6207  6231  6255  6279  6303  6327  6350  6372  6393  6414  6435
         6488  6533  6559  6584  6609  6634  6659  6684  6709  6734  6759  6783  6806  6828  6850
         6872  6924  6946  6967  6992  7023  7049  7075  7099  7123  7147  7206  7227  7248  7269
         7288  7307  7326  7375  7395  7419  7442  7495  7515  7539  7562  7615  7635  7659  7682
         7746  7766  7786  7847  7867  7887  7905  7923  7941  7988  8009  8031  8059  8083  8107
         8129  8151  8173  8224  8245  8267  8295  8319  8343  8365  8387  8409  8462  8484  8506
         8529  8550  8571  8597  8621  8645  8667  8689  8711  8764  8786  8808  8831  8852  8873
         8899  8923  8947  8969  8991  9013  9084  9106  9133  9157  9181  9203  9225  9247  9317
         9339  9366  9390  9414  9436  9458  9480  9550  9572  9599  9623  9647  9669  9691  9713
         9783  9805  9832  9856  9880  9902  9924  9946  10005  10024  10044  10064  10118  10138  10158
        10178  10232  10251  10271  10291  10345  10364  10384  10404  10460  10480  10501  10522  10577  10597
        10617  10637  10692  10711  10731  10751  10805  10825  10845  10865  10919  10938  10958  10978  11032
        11051  11071  11091  11145  11165  11185  11205  11261  11281  11302  11323  11379  11398  11418  11438
        11492  11511  11531  11551  11605  11625  11645  11665  11719  11739  11759  11779  11833  11852  11872
        11892  11948  11968  11989  12010  12071  12091  12111  12131  12190  12209  12229  12249  12308  12328
        12348  12368  12427  12447  12467  12487  12546  12565  12585  12605  12666  12686  12707  12728  12939
        12946  12995  13013  13136  13188  13237
ESCAPE      1#  1381#
GETPRI      1#  1381#
```

GETSWR 1# 1381#

MSG 3179# 3182 3570# 3572 3682# 3684 3763# 3765 3840# 3842 3921# 3923 3998# 4000 4079#
4081 4163# 4166 4188# 4198 4222# 4232 4251 4270 4288 4305 4323 4341 4352# 4355
4378# 4388 4412# 4422 4440 4459 4477 4495 4513 4523# 4526 4553# 4563 4589# 4599
4624# 4633 4651 4662# 4673 4692 4712 4731 4749 4767 4785 4795# 4798 4821# 4831
4861# 4871 4884# 4895 4917 4939 4961 4983 5005 5027 5049 5072 5094 5115 5135
5155 5175 5187# 5190 5215# 5226 5256# 5267 5281# 5292 5316 5340 5364 5388 5412
5436 5460 5484 5507 5529 5550 5571 5592 5606# 5609 5635# 5646 5678# 5689 5704#
5715 5739 5763 5787 5811 5835 5859 5883 5907 5930 5952 5973 5994 6015 6028#
6031 6057# 6068 6100# 6111 6125# 6136 6160 6184 6208 6232 6256 6280 6304 6328
6351 6373 6394 6415 6436 6449# 6452 6478# 6489 6523# 6534 6549# 6560 6585 6610
6635 6660 6685 6710 6735 6760 6784 6807 6829 6851 6873 6887# 6890 6915# 6925
6937# 6947 6968 6982# 6993 7024 7050 7076 7100 7124 7148 7164# 7167 7197# 7207
7228 7249 7270 7289 7308 7327 7338# 7341 7365# 7376 7396 7420 7443 7458# 7461
7485# 7496 7516 7540 7563 7578# 7581 7605# 7616 7636 7660 7683 7698# 7701 7737#
7747 7767 7787 7797# 7800 7838# 7848 7868 7888 7906 7924 7942 7953# 7956 7979#
7989 8000# 8010 8032 8060 8084 8108 8130 8152 8174 8189# 8192 8215# 8225 8236#
8246 8268 8296 8320 8344 8366 8388 8410 8425# 8428 8453# 8463 8475# 8485 8507
8530 8551 8572 8598 8622 8646 8668 8690 8712 8727# 8730 8755# 8765 8777# 8787
8809 8832 8853 8874 8900 8924 8948 8970 8992 9014 9030# 9033 9072# 9085 9107
9134 9158 9182 9204 9226 9248 9263# 9266 9305# 9318 9340 9367 9391 9415 9437
9459 9481 9496# 9499 9538# 9551 9573 9600 9624 9648 9670 9692 9714 9729# 9732
9771# 9784 9806 9833 9857 9881 9903 9925 9947 9963# 9966 9996# 10006 10025 10045
10065 10076# 10079 10109# 10119 10139 10159 10179 10190# 10193 10223# 10233 10252 10272 10292
10303# 10306 10336# 10346 10365 10385 10405 10416# 10419 10451# 10461 10481 10502 10523 10535#
10538 10568# 10578 10598 10618 10638 10650# 10653 10683# 10693 10712 10732 10752 10763# 10766
10796# 10806 10826 10846 10866 10877# 10880 10910# 10920 10939 10959 10979 10990# 10993 11023#
11033 11052 11072 11092 11103# 11106 11136# 11146 11166 11186 11206 11217# 11220 11252# 11262
11282 11303 11324 11337# 11340 11370# 11380 11399 11419 11439 11450# 11453 11483# 11493 11512
11532 11552 11563# 11566 11596# 11606 11626 11646 11666 11677# 11680 11710# 11720 11740 11760
11780 11791# 11794 11824# 11834 11853 11873 11893 11904# 11907 11939# 11949 11969 11990 12011
12024# 12027 12062# 12072 12092 12112 12132 12143# 12146 12181# 12191 12210 12230 12250 12261#
12264 12299# 12309 12329 12349 12369 12380# 12383 12418# 12428 12448 12468 12488 12499# 12502
12537# 12547 12566 12586 12606 12617# 12620 12657# 12667 12687 12708 12729 12744# 12747 13127#
13137 13175# 13189

MSG1 4883# 4901 4907# 4923 4929# 4945 4951# 4967 4973# 4989 4995# 5011 5017# 5033 5039#
5055 5061# 5078 5100 5121 5141 5161 5181 5280# 5299 5305# 5323 5329# 5347 5353#
5371 5377# 5395 5401# 5419 5425# 5443 5449# 5467 5473# 5491 5514 5536 5557 5578
5599 5703# 5722 5728# 5746 5752# 5770 5776# 5794 5800# 5818 5824# 5842 5848# 5866
5872# 5890 5896# 5914 5937 5959 5980 6001 6022 6124# 6143 6149# 6167 6173# 6191
6197# 6215 6221# 6239 6245# 6263 6269# 6287 6293# 6311 6317# 6335 6358 6380 6401
6422 6443 6548# 6568 6574# 6593 6599# 6618 6624# 6643 6649# 6668 6674# 6693 6699#
6718 6724# 6743 6749# 6768 6792 6815 6837 6859 6881 6983# 7003 7014# 7034 7060
7086 7110 7134 7158 7366# 7380 7386# 7400 7410# 7424 7433# 7447 7486# 7500 7506#
7520 7530# 7544 7553# 7567 7606# 7620 7626# 7640 7650# 7664 7673# 7687

MSG2 7980# 7993 8001# 8016 8038 8050# 8066 8090 8114 8136 8158 8180 8216# 8229 8237#
8252 8274 8286# 8302 8326 8350 8372 8394 8416 8454# 8467 8476# 8491 8513 8536
8557 8578 8588# 8604 8628 8652 8674 8696 8718 8756# 8769 8778# 8793 8815 8838
8859 8880 8890# 8906 8930 8954 8976 8998 9020 9073# 9091 9113 9124# 9140 9164
9188 9210 9232 9254 9306# 9324 9346 9357# 9373 9397 9421 9443 9465 9487 9539#
9557 9579 9590# 9606 9630 9654 9676 9698 9720 9772# 9790 9812 9823# 9839 9863
9887 9909 9931 9953 9996# 10011 10030 10050 10070 10109# 10124 10144 10164 10184 10223#
10238 10257 10277 10297 10336# 10351 10370 10390 10410 10451# 10467 10487 10508 10529 10568#
10583 10603 10623 10643 10683# 10698 10717 10737 10757 10796# 10811 10831 10851 10871 10910#
10925 10944 10964 10984 11023# 11038 11057 11077 11097 11136# 11151 11171 11191 11211 11252#
11268 11288 11309 11330 11370# 11385 11404 11424 11444 11483# 11498 11517 11537 11557 11596#

```
        11611   11631   11651   11671   11710#  11725   11745   11765   11785   11824#  11839   11858   11878   11898   11939#
        11955   11975   11996   12017   12062#  12077   12097   12117   12137   12181#  12196   12215   12235   12255   12299#
        12314   12334   12354   12374   12418#  12433   12453   12473   12493   12537#  12552   12571   12591   12611   12657#
        12673   12693   12714   12735
MULT        1#   1381#
NEWTST      1#   1381#   3180    3570    3682    3763    3840    3921    3998    4079    4164    4353    4524    4796    5188
         5607    6029    6450    6888    7165    7339    7459    7579    7699    7798    7954    8190    8426    8728    9031
         9264    9497    9730    9964   10077   10191   10304   10417   10536   10651   10764   10878   10991   11104   11218
        11338   11451   11564   11678   11792   11905   12025   12144   12262   12381   12500   12618   12745
PEDPE    1258#   7952    8188    8424    8726
POP         1#   1381#  13017   13110   13561   13831   14262   14263
PUSH        1#   1381#  12970   13100   13520   13805   14243   14249
REPORT      1#   1381#
RHCLEA   1258#   3701    3781    3859    3939    4017    4098    4183    4372    4548    4816    5210    5630    6052    6473
         6910    7182    7357    7477    7597    7718    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
SAVTST   1258#
SCOPE    1276#   3205    3575    3698    3778    3856    3936    4014    4095    4180    4369    4545    4813    5207    5627
         6049    6470    6907    7179    7354    7474    7594    7715    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   12800
SETPRI      1#   1381#  13734
SETTRA  14215#  14224   14225   14226   14227   14230   14231   14232   14233
SETUP       1#   1381#   3099
SIZDEV   1258#
SKIP        1#   1258#   1381#   12045   12164   12282   12401   12520   12639
SLASH       1#   1381#
SMORE    1258#  13458
SPACE    1381#
STARS       1#   1381#   1441    1445    1500    2534    2540    2592    2594    2793    3180    3204    3570    3574    3682
         3697    3763    3777    3840    3855    3921    3935    3998    4013    4079    4094    4164    4179    4353    4368
         4524    4544    4796    4812    5188    5206    5607    5626    6029    6048    6450    6469    6888    6906    7165
         7178    7339    7353    7459    7473    7579    7593    7699    7714    7798    7821    7954    7966    8190    8202
         8426    8439    8728    8741    9031    9046    9264    9279    9497    9512    9730    9745    9964    9974   10077
        10087   10191   10201   10304   10314   10417   10428   10536   10546   10651   10661   10764   10774   10878   10888
        10991   11001   11104   11114   11218   11229   11338   11348   11451   11461   11564   11574   11678   11688   11792
        11802   11905   11916   12025   12036   12144   12155   12262   12273   12381   12392   12500   12511   12618   12630
        12745   12751   12793   12839   12841   12864   12866   13445   13510   13577   13647   13723   13747   13791   13844
        14117   14194   14239   14255
STARTT   1258#   3699    3779    3857    3937    4015    4096    4181    4370    4546    4814    5208    5628    6050    6471
         6908    7180    7355    7475    7595    7716    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
SWRSU       1#   1381#  3120#
TJUMP    1258#   3538
TRMTRP  14215#
TTSTNO   1258#   3700    3780    3858    3938    4016    4097    4182    4371    4547    4815    5209    5629    6051    6472
         6909    7181    7356    7476    7596    7717    7824    7969    8205    8442    8744    9049    9282    9515    9748
         9977   10090   10204   10317   10431   10549   10664   10777   10891   11004   11117   11232   11351   11464   11577
        11691   11805   11919   12039   12158   12276   12395   12514   12633
TYPBIN      1#   1381#
TYPDEC      1#   1381#  12812
TYPNAM      1#   1381#
TYPNUM      1#   1381#
```

```
TYPOCS          1#    1381#
TYPOCT          1#    1381#
TYPTXT          1#    1381#    3152    3216    3221    3231    3238    3244    3385    3392    3398    3404    3419    3426    3434
             3442     3450     3458    3466    3474    3482    3490    3498    3504    3511    3517    3521    3584    3588    3595
             3599     3607     3616    3623    3630    3637    3644    3651   12755   12765   12771   12777   12781   12843   12850
            12867    12872    12878   12883   12889   12895   13047   13053   13060   13064   13068   13075   13079   13378   13385
            13396    13402    13408   13412   13418   13424   13430   13905   13913   13921   13929   13937   13945   13953   13961
            13969    13977    13985   13993   14001   14009   14017   14025   14033   14041   14049
WCEOW        1258#    9029     9262    9495    9728
WT           1258#    4211     4401    4577    4613    4850    5245    5667    6089    6512   13152   13206   13246   13256   13268
            13284    13305    13315   13327   13352   13363
$$CMRE       1443#    1482     1483    1484    1485    1486    1487
$$CMTM       1443#    1488     1489    1490    1491    1492    1493
$$ESCA          1#    1381#
$$NEWT          1#    1381#    3180    3570    3682    3763    3840    3921    3998    4079    4164    4353    4524    4796    5188
             5607     6029     6450    6888    7165    7339    7459    7579    7699    7798    7954    8190    8426    8728    9031
             9264     9497     9730    9964   10077   10191   10304   10417   10536   10651   10764   10878   10991   11104   11218
            11338    11451    11564   11678   11792   11905   12025   12144   12262   12381   12500   12618   12745
$$SET       14215#   14224    14225   14226   14227   14230   14231   14232   14233
$$SKIP          1#    1381#
.EQUAT          1#    1248#    1271
.HEADE          1#    1248#
.KT11           1#    1248#    1406
.SETUP          1#    1248#    3098
.SWRHI          1#    1248#    1258
.SWRLO       1248#    1269#
.$ACTI          1#
.$APTB          1#
.$APTH          1#
.$APTY          1#
.$ASTA          1#
.$CATC          1#    1248#    1382
.$CMTA          1#    1248#    1443
.$DB2D          1#
.$DB20          1#
.$DIV           1#
.$EOP           1#    1248#   12791
.$ERRO          1#    1248#   13[illegible]2
.$ERRT          1#    1248#
.$MULT          1#
.$POWE          1#    1248#   14237
.$RAND          1#
.$RDDE          1#
.$RDOC          1#    1248#   13789
.$READ          1#    1248#   13645
.$R2AZ          1#
.$SAVE          1#
.$SB2D          1#
.$SB20          1#
.$SCOP          1#    1248#   13443
.$SIZE          1#
.$SUPR          1#
.$TRAP          1#    1248#   14192
.$TYPB          1#
.$TYPD          1#    1248#   13508
.$TYPE          1#    1248#   13575
```

```
CERHAD0 MACY11 30(1046)  21-DEC-77  13:06  PAGE 326
CERHAD.P11    21-DEC-77 12:48          CROSS REFERENCE TABLE -- MACRO NAMES                    SEQ 0323

.$TYPO       1#   1248#  14115
.$40CA       1#
.1170        1#


. ABS.  067012      000

 ERRORS DETECTED:  0

 CERHAD.BIN,CERHAD.LST/CRF/SOL/NL:TOC=CERHAD.SML,CERHAD.P11
 RUN-TIME:  34 50 3 SECONDS
 RUN-TIME RATIO: 343/87=3.9
 CORE USED:  37K  (73 PAGES)
```

EOFICERHADSEG 00010000 780330 PDP10 #11

NO9

60N