DRV11J

DRV11J DIAG TST PRT 1

CVDRCA0

AH-F758A-MC

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-F756A-MC

PRODUCT NAME: CVDRCA0 DRV11J DIAG TST PRT1

DATE CREATED: OCTOBER 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

1.0 ABSTRACT

THE DRV11-J IS A GENERAL PURPOSE PARALLEL INTERFACE FOR THE LSI-11 BUS. IT HAS A BASIC CONFIGURATION OF 64 TRI-STATE IN/OUT LINES DIVIDED INTO FOUR GROUPS OR PORTS OF 16 BIT WORDS.

THERE ARE TWO 4K DIAGNOSTICS FOR THE DRV11-J OPTION. THE DRV11-J DIAGNOSTIC TEST PART 1 OF 2 CONTAINS A SERIES OF TESTS WITHOUT DRV11J INTERRUPTS DESIGNED TO TEST ALL LOGIC FUNCTIONS AND DATA PATHS MADE ACCESSIBLE WITH THE LOOPBACK CABLE INSERTED INTO THE DRV11-J I/O CONNECTORS.

THE DRV11-J DIAGNOSTIC PART 2 OF 2 IS A SERIES OF TESTS WITH DRV11J INTERRUPTS DESIGNED TO TEST ALL LOGIC AND DATA PATHS MADE ACCESSIBLE WITH THE LOOPBACK CABLE INSERTED INTO THE I/O CONNECTORS.

THE DRV11-J IS CONTAINED ON A DOUBLE HEIGHT MODULE. THE MODULE CONTAINS TWO 50 PIN CONNECTORS FOR INTERFACING TO EXTERNAL USER DEVICES.

FOR DIAGNOSTIC TESTING, THE DRV11-J CABLE(BC05W-02) MUST BE INSTALLED WITH 1/2 TWIST BETWEEN THE 50 PIN CONNECTORS.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP11/03,11/23 COMPUTER OR LSI-11 PROCESSOR WITH A MINIMUM OF 4K MEMORY.
2. SERIAL LINE INTERFACE AND CONSOLE TERMINAL
3. DRV11-J OPTION WITH A BC05W-02 CABLE

2.2 STORAGE

THE PROGRAM USES THE LOWER 4K OF MEMORY.

3.0 LOADING PROCEDURE

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED PAPERTAPES OR XXDP MEDIA (FILES WITH .BIC OR .BIN EXTENSIONS ONLY).

4.0 STARTING PROCEDURE

1. MAKE SURE THE DRV11-J CABLE IS INSERTED WITH 1/2 A TWIST ON THE I/O CONNECTORS OF THE DRV11-J OPTION. THIS WILL CONNECT PORT A TO PORT C AND PORT B TO PORT D.
2. MAKE SURE THE DEVICE BUS ADDRESSES AGREE WITH THE DEFAULT VALUES DEFINED IN SECTION 7.1. IF NOT, CHANGE

LOCATION(S) AS DESIRED VIA THE 'ADDRESS/' ODT COMMAND.

3. THE PROGRAM SHOULD ALWAYS BE STARTED AT 200. STARTING AT 200(200G OR .R CVDRCA UNDER XXDP+),THE PROGRAM INITIALIZES ITSELF,PRINTS ITS ID(FIRST TIME ONLY) AND THEN PRINTS THAT THE DRV11-J CABLE IS REQUIRED(FIRST TIME ONLY) AND THEN PRINTS: SWR=XXXXXX NEW=

   WHERE XXXXXX REPRESENTS THE CURRENT VALUE OF THE SOFTWARE SWITCH REGISTER. IF NO CHANGES ARE REQUIRED IN THE SWITCH REGISTER THEN JUST HIT CARRIAGE RETURN.
   IF CHANGES ARE REQUIRED,THEN A NEW VALUE MAY BE TYPED FOLLOWED BY A CARRIAGE RETURN.
   REFER TO SECTION 5.0 FOR SWITCH REGISTER OPTIONS.

4. IF THE FOLLOWING ERROR OCCURS RIGHT AFTER STARTUP IT IS POSSIBLE THAT THE DRV11J LOOPBACK CABLE MAY NOT BE INSTALLED OR WAS NOT INSTALLED PROPERLY:

   REG READ/WRITE ER
   ERRPC TSTNUM BUSADR EXPCT RCVD
   002224 000002 *164160 100700 000700
   *BUSADR MAY BE DIFFERENT DEPENDING ON THE CSR OF THE DRV11J.

5.0 SOFTWARE SWITCH REGISTER

5.1 OPTIONS

THE PROGRAM SWITCH DEFAULT MODE IS SWR = 000000
IF USING A VIDEO TERMINAL ,BIT 15 = 1(HALT ON ERROR)
MAY BE HELPFUL IN KEEPING THE ERROR ON THE SCREEN.

SWITCH	OCTAL	FUNCTION
SW15=1	100000	HALT ON ERROR
SW14=1	040000	LOOP ON TEST
SW13=1	020000	INHIBIT ERROR TYPEOUTS
SW11=1	004000	INHIBIT ITERATIONS
SW09=1	001000	LOOP ON ERROR
SW08=1	0004XX	LOOP ON TEST IN SWR <7-0>

5.2 CONTROL

1. THE SOFTWARE SWITCH REGISTER 'SWREG' (LOC. 176) CAN BE CHANGED BY USING THE ODT FACILITIES.

2. THE SOFTWARE SWITCH REGISTER CAN BE CHANGED UNDER PROGRAM CONTROL BY TYPING THE 'CONTROL & G' KEYS. THIS KEYBOARD OPERATION WILL PRINT OUT THE CURRENT CONTENTS AND ACCEPT NEW OCTAL SWITCH REGISTER DATA TERMINATED WITH A CARRIAGE RETURN.

3. ONCE THE ODT MODE HAS BEEN ENTERED BECAUSE OF AN ERROR CONDITION WITH BIT15 SET (HALT ON ERROR), STEP #2 ABOVE IS OF NO VALUE, SO RESORT TO STEP #1 TO ALTER THE SOFTWARE SWITCH REGISTER IF DESIRED BEFORE TYPING 'P' (CONTINUE).

6.0 ERROR REPORTING

6.1 ERROR COMMENT

ALL ERRORS ARE ACCOMPANIED WITH AN ENGLISH LANGUAGE DESCRIPTIVE COMMENT AS TO THE TYPE OF FAILURE. FURTHER QUALIFICATION OF THE ERROR CAN BE OBTAINED IF NEEDED FROM THE COMMENT AT THE ERROR PC OR FROM THE TEST ITSELF.

TO CONSERVE MEMORY SPACE FOR THE 4K PROGRAM MEMORY REQUIREMENT,REGISTER 5 (R5) IS RESERVED FOR THE $BDDAT LOCATION (1126).

EXAMPLE: CMP $GDDAT,(R5) IS THE SAME AS CMP $GDDAT,$BDDAT.

6.2.1 ERROR DATA

```
ERROR TITLE HEADING
ERRPC   TSTNUM  BUSADR  EXPCT   RCVD
XXXXXX  XXXXXX  XXXXXX  XXXXXX  XXXXXX

ERRPC           LISTING ADDRESS WHERE THE ERROR WAS DETECTED
TSTNUM          TEST NUMBER WHERE THE ERROR OCCURRED
BUSADR          DRV11J BUS REG ADDRESS OF CONCERNED OPERATION
EXPCT           DATA THAT WAS EXPECTED
RCVD            DATA THAT WAS RECEIVED
```

6.2.2 ERROR TITLES

```
REG TIMEOUT ER          ;REGISTER TIMEOUT ERROR

REG READ/WRITE ER       ;REGISTER READ/WRITE ERROR

IRR REG ER              ;INTERRUPT REQUEST REGISTER ERROR

ACR REG ER              ;AUTOCLEAR REGISTER ERROR

IMR REG ER              ;INTERRUPT MASK REGISTER ERROR

ISR REG ER              ;INTERRUPT SERVICE REGISTER ERROR

CHIP STAT ER            ;CHIP STATUS ERROR
```

7.0 MISCELLANEOUS

7.1 DRV11-J BUS ADDRESS MODIFICATION

MODIFY LOCATION '$BASE' (ADDR: 1244) IF BASE BUS ADDRESS
IS NOT 164160.

7.2 XXDP/APT NOTES

THIS DIAGNOSTIC DOES SUPPORT ACT AND APT ENVIRONMENTS.

7.3 POWER FAIL

A POWER FAILURE WILL CAUSE A RESTART MESSAGE ON POWER UP AT
WHICH TIME THE PROGRAM IS RESTARTED (ONLY ON SYSTEMS WITH
NON-VOLATILE MEMORY AND WITH APPROPRIATE HARDWARE).

7.4 MULTIPLE DRV11J INTERFACE TESTING

THIS PROGRAM DOES NOT ''AUTO-SIZE'' THE NUMBER OF DRV11J'S CONNECTED.
THIS DIAGNOSTIC WILL TEST SEQUENTIALLY UP TO 4 DRV11J INTERFACES
WITH CONTIGUOUS BUS ADDRESSES. THIS IS ACCOMPLISHED
BY THE USER SETTING UP LOCATION 'DEVM' (ADDR: 1246)
WITH A BIT MAP INDICATING WHAT INTERFACES ARE TO BE TESTED.
I.E. BIT0 = 1 SAYS TEST 1ST DRV11J,
BIT1 = 1 SAYS TEST 2ND DRV11J
BIT2 = 1 SAYS TEST 3RD DRV11J
BIT3 = 1 SAYS TEST 4TH DRV11J

1ST UNIT = STARTING CSRA	164160	$DEVM = 1
2ND UNIT = STARTING CSRA	164140	$DEVM = 3
3RD UNIT = STARTING CSRA	164120	$DEVM = 7
4TH UNIT = STARTING CSRA	164100	$DEVM = 17

7.5 RESTRICTIONS

8.0 EXECUTION TIME

EXECUTION TIME RANGES FROM ABOUT <5 SECONDS ON FIRST
PASS WITH NO ITERATIONS TO <20 SECONDS WITH ITERATIONS
ENABLED AFTER FIRST PASS,PER DRV11J UNIT CONNECTED.
AN 'END PASS'' MESSAGE INDICATES ALL TESTS HAVE COMPLETED
ON ALL SELECTED UNITS.

9.0 PROGRAM TEST DESCRIPTIONS

GENERAL

THIS DIAGNOSTIC CONTAINS A SERIES OF INDEPENDENT TESTS DESIGNED
TO TEST LOGIC FUNCTIONS AND DATA PATHS OF THE DRV11J OPTION.
TESTING IS ACCOMPLISHED WITH THE AID OF THE
DRV11J LOOP BACK CABLE PROVIDED FOR DIAGNOSTIC TESTING.
A COMPLETE LIST OF TESTS IS AVAILABLE IN THE TABLE OF CONTENTS
AT THE BEGINNING OF THE LISTING. THE COMMENT FIELD WITHIN
EACH TEST CAN BE BENEFICIAL IN TEST UNDERSTANDING.

TESTING WITH THE DRV11J CABLE ALLOWS FOR TESTING
OF PORT A WITH PORT C AND THE TESTING OF PORT B
WITH PORT D.

9.1 TESTS T1-T22 REGISTER CHECKING

THESE TESTS WILL WRITE/READ PORTS A TO C
AND WRITE/READ PORTS B TO D WITH FLOATING
ONES AND FLOATING ZEROS,GROWING ONES AND
GROWING ZEROS AND DATA PATTERNS.
TESTS ARE PERFORMED TO INSURE INTERACTION WITH
CONNECTED PORTS AND NO INTERACTION BETWEEN
UNCONNECTED PORTS.

9.2 T22-T42 INTERRUPT CONTROL REGISTER CHECKING

TESTS ARE MADE TO THE INTERRUPT CONTROL CHIP
REGISTERS IRR,ACR,IMR AND LIMITED TESTING OF
ISR REGISTER UNTIL INTERRUPTS ARE PERFORMED.
TESTS ARE PERFORMED ON THE REGISTERS WITH
FLOATING ONES,FLOATING ZEROS,GROWING ONES
AND GROWING ZEROS. CHIP RESET CAPABILITIES
ARE TESTED AS WELL AS UNIQUENESS OF REGISTERS
WITHIN A CHIP GROUP AND THE UNIQUENESS OF
ONE GROUP'S REGISTERS TO ANOTHER GROUP'S
REGISTERS.

9.3 TEST 43 - TEST STATUS BITS GINT,S2,S1,S0,GP1,GP2

EXERCISE THE STATUS BITS S2,S1,S0 AND GINT FOR
EACH GROUP CHIP BY SETTING SINGLE IRR BITS TO CAUSE THE
STATUS BITS TO GO FROM 120 TO 127. EACH IRR BIT
IS THEN MASKED TO RETURN STATUS BACK TO ORIGINAL
STATUS WITH NO IRR BITS SET.

9.4 T44-T47 POLLED MODE TESTING

THIS TEST WILL WRITE PATTERNS INTO DBRA WITH
EITHER LOW TO HIGH OR HIGH TO LOW POLARITIES.
AFTER PLACING ALL ONES IN DBRA AND THEN SELECTING
ACTIVE LOW,CLEARING DBRA WILL NOW SET IRR BITS
0-7,GROUP 1 AND IRR BITS0-3,GROUP2.
THE RPLY SIGNALS WILL SET IRR BITS 4-7
WHEN WRITING DBRA IN OUTPUT MODE,THIS WILL

SET THE RPLY FOR DBRC.(IRR6,GROUP 2)
WHEN READING DBRC IN INPUT MODE WILL SET
THE RPLY FOR DBRA.(IRR BIT 4,GROUP 2)
WHEN WRITING DBRB IN OUTPUT MODE,THIS WILL
SET THE RPLY FOR DBRD(IRR BIT7,GROUP2)
WHEN READING DBRD IN INPUT MODE,THIS WILL
SET THE RPLY FOR DBRB(IRR BIT5,GROUP 2).
TEST ALL DATA PATTERNS TO SET IRR BITS
GROUP1 AND GROUP2 AND FOR THE SETTING OF RPLY
BITS IN THE GROUP 2 IRR REGISTER BY WRITING IN
OUTPUT MODE AND READING IN INPUT MODE.
TEST THAT NO RPLY BITS SET WHEN READING IN OUTPUT
MODE AND WHEN WRITING IN INPUT MODE.

9.5 T50-T51 CSR'S WITH RESET
SET UP CSR'S IN OUTPUT MODE AND CLEAR
DIRECTION BIT OUT OF EACH CSR EXCEPT FOR
CSRA WHICH WILL CLEAR DIR BIT AND I/E BIT.

10.0 LISTING

```
                                .TITLE  CVDRCA  DRV11J DIAG TST PRT1
                                ;*COPYRIGHT (C) 1979
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY BILL HEAVEY
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
165400                          $SWR=165400
000001                          $TN=1
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .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                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
```

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

                                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
                100000                  SW15=   100000
                040000                  SW14=   40000
                020000                  SW13=   20000
                010000                  SW12=   10000
                004000                  SW11=   4000
                002000                  SW10=   2000
                001000                  SW09=   1000
                000400                  SW08=   400
                000200                  SW07=   200
                000100                  SW06=   100
                000040                  SW05=   40
                000020                  SW04=   20
                000010                  SW03=   10
                000004                  SW02=   4
                000002                  SW01=   2
                000001                  SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

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

```
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

                                ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
000004                          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                          TBITVEC=14              ;;''T'' BIT
000014                          TRTVEC= 14              ;;TRACE TRAP
000014                          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                          PWRVEC= 24              ;;POWER FAIL
000030                          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                          TRAPVEC=34              ;;''TRAP'' TRAP
000060                          TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                          TPVEC=  64              ;;TTY PRINTER VECTOR
000240                          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
164160                          ABASE=  164160          ;BASE ADDRESS
000001                          ADEVM=  1               ;DEFAULT TO ONE DRV11J
100000                          RDY=    BIT15
000400                          DIR=    BIT8
001000                          IE=     BIT9

                                ;CHIP COMMAND SUMMARY
000020                          CIRMR=  20              ;CLEAR IRR AND IMR
000030                          CSIRMR= 30              ;CLEAR SINGLE IRR AND IMR BIT

000040                          CIMR=   40              ;CLEAR IMR
000050                          CSIMR=  50              ;CLEAR SINGLE IMR BIT
000060                          SIMR=   60              ;SET ALL IMR BITS
000070                          SSIMR=  70              ;SET SINGLE IMR BITS

000100                          CIRR=   100             ;CLEAR IRR
000110                          CSIRR=  110             ;CLEAR SINGLE IRR BITS
000120                          SIRR=   120             ;SET ALL IRR BITS
000130                          SSIRR=  130             ;SET SINGLE IRR BITS

000140                          CHPISR= 140             ;CLEAR HIGHEST PRIORITY ISR BIT
000160                          CISR=   160             ;CLEAR ISR
000170                          CSISR=  170             ;CLEAR SINGLE ISR BIT

000200                          LMD04=  200             ;LOAD MODE BITS M0-M4
000240                          LMD57=  240             ;LOAD MODE BITS M5-M7

                                ;CHIP MODE BIT PRESELECTION
000240                          MISR=   240
000244                          MIMR=   244
000250                          MIRR=   250
000254                          MACR=   254

                                ;CHIP WRITE PRESELECTION
```

```
   169            000300                      PACR=   300             ;PRESELECT AUTO CLEAR REG. FOR WRITING
   170            000260                      PIMR=   260             ;PRESELECT IMR REG. FOR WRITING
   171            000340                      PVMA=   340             ;PRESELECT VECTOR MEMORY ADDRESS
   172
   173                                        .SBTTL  TRAP CATCHER
   174
   175            000000                              .=0
   176                                        ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ''.+2,HALT''
   177                                        ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
   178                                        ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
   179            000174                              .=174
   180  000174   000000                      DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
   181  000176   000000                      SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
   182                                        .SBTTL  STARTING ADDRESS(ES)
   183  000200   000137  001402                      JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
   184            000100                              .=100
   185  000100   000104  000200  000002              .WORD   104,200,2       ;IF 'B EVENT' ON Q BUS IS CONNECTED
   186                                                                        ;IGNORE IT'S INTERRUPT - JUST DO A RTI
```

```
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
                                        $SVPC=.                 ;SAVE PC
000106                                  .=46
000046                                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000046  012746                         .=52
000052                                  .WORD   0               ;;2)SET LOC.52 TO ZERO .
000052  000000                         .=$SVPC                 ;; RESTORE PC
000106                                  .=1000
001000
                                                                ;LONGEST TEST TIME
                                                                ;1ST PASS RUN TIME
                                                                ;ADDITIONAL RUN TIME
                                .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************************
001000                                  .$X=.   ;;SAVE CURRENT LOCATION
000024                                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                         200     ;;FOR APT START UP
000044                                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                         $APTHDR ;;POINT TO APT HEADER BLOCK
001000                                  .=.$X   ;;RESET LOCATION COUNTER
                                ;;******************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                          $APTHD:
001000  000000                  $HIBTS: .WORD   0               ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001170                  $MBADR: .WORD   $MAIL           ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000006                  $TSTM:  .WORD   6.              ;;RUN TIM OF LONGEST TEST
001006  000024                  $PASTM: .WORD   20.             ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000024                  $UNITM: .WORD   20.             ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000031                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                        .SBTTL  COMMON TAGS

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

                001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
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         $TIMES:  0                       ;;MAX. NUMBER OF ITERATIONS
001162  000000         $ESCAPE: 0                       ;;ESCAPE ON ERROR ADDRESS
001164  077            $QUES:   .ASCII  /?/             ;;QUESTION MARK
001165  015            $CRLF:   .ASCII  <15>            ;;CARRIAGE RETURN
001166  000012         $LF:     .ASCIZ  <12>            ;;LINE FEED
                        ;;*****************************************************************
                        .SBTTL  APT MAILBOX-ETABLE

                        ;;*****************************************************************
                        .EVEN
001170                  $MAIL:                          ;;APT MAILBOX
001170  000000         $MSGTY:  .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001172  000000         $FATAL:  .WORD   AFATAL          ;;FATAL ERROR NUMBER
001174  000000         $TESTN:  .WORD   ATESTN          ;;TEST NUMBER
001176  000000         $PASS:   .WORD   APASS           ;;PASS COUNT
001200  000000         $DEVCT:  .WORD   ADEVCT          ;;DEVICE COUNT
001202  000000         $UNIT:   .WORD   AUNIT           ;;I/O UNIT NUMBER
001204  000000         $MSGAD:  .WORD   AMSGAD          ;;MESSAGE ADDRESS
001206  000000         $MSGLG:  .WORD   AMSGLG          ;;MESSAGE LENGTH
```

```
                                $ETABLE:                        ;;APT ENVIRONMENT TABLE
001210                          $ENV:   .BYTE   AENV            ;;ENVIRONMENT BYTE
001210  000                     $ENVM:  .BYTE   AENVM           ;;ENVIRONMENT MODE BITS
001211  000                     $SWREG: .WORD   ASWREG          ;;APT SWITCH REGISTER
001212  000000                  $USWR:  .WORD   AUSWR           ;;USER SWITCHES
001214  000000                  $CPUOP: .WORD   ACPUOP          ;;CPU TYPE,OPTIONS
001216  000000                  ;*                              BITS 15-11=CPU TYPE
                                ;*                                      11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                                      11/70=06,PDQ=07,G=10
                                ;*                              BIT 10=REAL TIME CLOCK
                                ;*                              BIT  9=FLOATING POINT PROCESSOR
                                ;*                              BIT  8=MEMORY MANAGEMENT
001220  000                     $MAMS1: .BYTE   AMAMS1          ;;HIGH ADDRESS,M.S. BYTE
001221  000                     $MTYP1: .BYTE   AMTYP1          ;;MEM. TYPE,BLK#1
                                ;*                              MEM.TYPE BYTE   --  (HIGH BYTE)
                                ;*                                      900 NSEC CORE=001
                                ;*                                      300 NSEC BIPOLAR=002
                                ;*                                      500 NSEC MOS=003
001222  000000                  $MADR1: .WORD   AMADR1          ;;HIGH ADDRESS,BLK#1
                                ;*                                MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
001224  000                     $MAMS2: .BYTE   AMAMS2          ;;HIGH ADDRESS,M.S. BYTE
001225  000                     $MTYP2: .BYTE   AMTYP2          ;;MEM.TYPE,BLK#2
001226  000000                  $MADR2: .WORD   AMADR2          ;;MEM.LAST ADDRESS,BLK#2
001230  000                     $MAMS3: .BYTE   AMAMS3          ;;HIGH ADDRESS,M.S.BYTE
001231  000                     $MTYP3: .BYTE   AMTYP3          ;;MEM.TYPE,BLK#3
001232  000000                  $MADR3: .WORD   AMADR3          ;;MEM.LAST ADDRESS,BLK#3
001234  000                     $MAMS4: .BYTE   AMAMS4          ;;HIGH ADDRESS,M.S.BYTE
001235  000                     $MTYP4: .BYTE   AMTYP4          ;;MEM.TYPE,BLK#4
001236  000000                  $MADR4: .WORD   AMADR4          ;;MEM.LAST ADDRESS,BLK#4
001240  000000                  $VECT1: .WORD   AVECT1          ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001242  000000                  $VECT2: .WORD   AVECT2          ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001244  164160                  $BASE:  .WORD   ABASE           ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001246  000001                  $DEVM:  .WORD   ADEVM           ;;DEVICE MAP
001250  000000                  $CDW1:  .WORD   ACDW1           ;;CONTROLLER DESCRIPTION WORD#1
001252                          $ETEND:
                                .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


001252                          $ERRTB:
                                ;ERROR  1
001252  016557                          EM1             ;REG TIMEOUT ER
001254  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001256  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
001260  000000                          0

                                ;ERROR  2
001262  016576                          EM2             ;REG READ/WRITE ER
001264  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001266  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
001270  000000                          0

                                ;ERROR  3
001272  016620                          EM3             ;IRR REG ER
001274  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001276  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
001300  000000                          0

                                ;ERROR  4
001302  016633                          EM4             ;ACR REG ER
001304  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001306  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
001310  000000                          0

                                ;ERROR  5
001312  016646                          EM5             ;IMR REG ERROR
001314  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001316  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
001320  000000                          0

                                ;ERROR  6
001322  016661                          EM6             ;ISR REG ERROR
001324  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001326  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
001330  000000                          0


                                ;ERROR  7
001332  016674                          EM7             ;CHIP STAT ER
001334  016711                          DH1             ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001336  017034                          DT1             ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT(R5)
```

```
   371  001340  000000                          0
   372
   373
   374                                  ; BUS REGISTER ADDRESS POINTERS
   375
   376
   377  001342  164160                  DRCSA:  ABASE
   378  001344  164162                  DRDBA:  ABASE+2
   379  001346  164164                  DRCSB:  ABASE+4
   380  001350  164166                  DRDBB:  ABASE+6
   381  001352  164170                  DRCSC:  ABASE+10
   382  001354  164172                  DRDBC:  ABASE+12
   383  001356  164174                  DRCSD:  ABASE+14
   384  001360  164176                  DRDBD:  ABASE+16
   385
   386
   387                                  ;COMMON PROGRAM LOCATION(S)
   388
   389  001362  000000                  TSTNUM: 0               ;CONTAINS TEST NUMBER ON ERROR
   390  001364  000001                  DMAP:   1
   391  001366  000000                  INTFLG: .WORD   0
   392  001370  000000                  XXDP:   .WORD   0
   393  001372  000000                  IMRLOC: .WORD   0
   394  001374  000000                  ISRLOC: .WORD   0
   395  001376  000000                  IRRLOC: .WORD   0
   396  001400  000000                  ACRLOC: .WORD   0
```

```
                                        .SBTTL  PROGRAM START
001402                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001402  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001406  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001410  022706  001140                  CMP     #SWR,R6 ;;DONE?
001414  001374                          BNE     .-6             ;;LOOP BACK IF NO
001416  012706  001100                  MOV     #1100,SP                ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
001422  012737  015142  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001430  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001436  012737  014614  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001444  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001452  012737  016402  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001460  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
001466  012737  016176  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001474  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001502  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001506  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001512  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
001520  012737  001520  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001526  012737  001526  001110          MOV     #.,$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.
001534  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001540  012737  001574  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001546  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001554  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001562  022777  177777  177350          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001570  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001572  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001574  012716  001602          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001600  000002                          RTI
001602  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001610  012737  000174  001142          MOV     #DISPREG,DISPLAY
001616  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001622  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
001626  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
001634  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
001636  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
001644                          67$:
001644  005037  001172                  CLR     $FATAL          ;CLEAR ERROR NUMBER
001650  005037  001170                  CLR     $MSGTYP         ;CLEAR MESSAGE TYPE
001654  005037  001174                  CLR     $TESTN          ;CLEAR TEST NUMBER
                                ;CHECK OPERATING ENVIRONMENT
001660  005737  000042                  TST     @#42            ;ARE WE IN ACT/XXDP AUTO MODE?
001664  001410                          BEQ     1$              ;BRANCH IF NO
001666  023737  000042  000046          CMP     @#42,@#46       ;IS IT ACT AUTO MODE?
001674  001410                          BEQ     2$              ;BRANCH IF YES
001676  012737  177777  001370          MOV     #-1,XXDP        ;SET XXDP CHAIN MODE INDICATOR
001704  000404                          BR      2$
001706  123727  001210  000001  1$:     CMPB    $ENV,#1         ;ARE WE IN APT AUTO MODE?
001714  001003                          BNE     3$              ;BRANCH IF NO
```

```
453  001716  112737  000001  001134  2$:     MOVB    #1,$AUTOB       ;SET AUTO MODE INDICATOR
454
455                                  ;PRINT TITLE IF NOT IN ACT OR APT AUTO MODE
456  001724  005227  177777          3$:     INC     #-1             ;FIRST TIME?
457  001730  001012                          BNE     5$              ;SKIP TITLE IF NO
458  001732  005737  001134                  TST     $AUTOB          ;ARE WE IN AUTO MODE?
459  001736  001403                          BEQ     4$              ;BRANCH TO TITLE TYPEOUT IF NOT
460  001740  005737  001370                  TST     XXDP            ;IS THE AUTO MODE UNDER XXDP?
461  001744  001404                          BEQ     5$              ;SKIP TITLE IF NOT
462  001746  104401  016462          4$:     TYPE    ,TITLED         ;PRINT OUT THE TITLE
463  001752  104401  016530                  TYPE    ,TLCABL         ;PRINT 'DRV11J CABLE REQ'D''
464
465                                  ;GET THE VALUE IN THE SOFTWARE SWITCH REGISTER
466  001756  005737  001134          5$:     TST     $AUTOB          ;ARE WE IN AUTOMATIC MODE?
467  001762  001001                          BNE     START1          ;BRANCH IF YES
468  001764  104406                          GTSWR                   ;ASK FOR SWR INPUT FROM CONSOLE
469  001766  005037  001202          START1: CLR     $UNIT           ;CLEAR UNIT NUMBER
470  001772  013737  001246  001364          MOV     $DEVM,DMAP      ;POSITION OF DRV11-J'S
471  002000  042737  177760  001364          BIC     #177760,DMAP    ;UP TO 4 DRV11-J'S ONLY
472  002006  013701  001244                  MOV     $BASE,R1        ;GET BASE ADDRESS
473  002012  010137  001342                  MOV     R1,DRCSA        ;MAKE BUS REG. POINTER = $BASE
474  002016  032737  000001  001364          BIT     #1,DMAP         ;IS FIRST DRV11-J SELECTED?
475  002024  001002                          BNE     NEXPAS          ;YES
476  002026  000137  012624                  JMP     NXDEV1          ;ADVANCE BASE DRV11-J ADDRESS
477  002032  012700  001342          NEXPAS: MOV     #DRCSA,R0       ;SET UP REGISTER ADDRESS POINTERS
478  002036  010120                  NEXPA1: MOV     R1,(R0)+        ;LOAD EM,R1 = DRV11-J CSRA ADDRESS
479  002040  062701  000002                  ADD     #2,R1           ;DO POINTERS FOR ALL REGS, A THRU D
480  002044  022700  001362                  CMP     #DRDBD+2,R0     ;ALL DONE?
481  002050  001372                          BNE     NEXPA1          ;BR IF NOT
482  002052  012706  001100                  MOV     #STACK,SP       ;ALWAYS RESET STACK
483  002056  012705  001126                  MOV     #$BDDAT,R5      ;INIT R5 WITH $BDDAT
484  002062  013737  001202  001200          MOV     $UNIT,$DEVCT    ;LOAD APT COUNTER WITH UNIT NO.
485  002070  106427  000340                  MTPS    #PR7            ;SET PRIORITY TO 7
486
487
488
489                                  ;;*****************************************************************
490                                  ;*TEST 1         TEST THAT ALL  REGISTERS ARE ADDRESSABLE
491                                  ;;*****************************************************************
492  002074  000004                  TST1:   SCOPE
493  002076  005037  001124                  CLR     $GDDAT          ;NO DATA COMPARE
494  002102  005015                          CLR     (R5)            ;NO DATA COMPARE
495  002104  012737  002140  000004          MOV     #2$,@#ERRVEC    ;SET UP TIMEOUT RETURN ADDRS
496  002112  013700  001342                  MOV     DRCSA,R0        ;SET UP 1ST DRV11 BUS ADRS
497  002116  012701  000010                  MOV     #8.,R1          ;SET UP REG COUNT
498  002122  010037  001122          1$:     MOV     R0,$BDADR       ;SET UP CURRENT DRV BUS ADRS
499  002126  005010                          CLR     (R0)            ;SEE IF THERE
500  002130  005720                          TST     (R0)+           ;BUMP TO NEXT
501  002132  005301                          DEC     R1              ;COUNT 8 OF THEM
502  002134  001403                          BEQ     3$              ;BR IF ALL DONE
503  002136  000771                          BR      1$              ;TRY NEXT
504  002140  022626                  2$:     CMP     (SP)+,(SP)+     ;FIX STACK SINCE NO RTI
505  002142  104001                          ERROR   1               ;BUS ADRS INDICATED DID NOT RESPOND
506  002144  012737  000006  000004  3$:     MOV     #ERRVEC+2,@#ERRVEC  ;RESTORE LOC 4
507
508                                  ;;*****************************************************************
```

```
                                        ;*TEST 2        TEST CSRA W/R DIR BIT,INT. CHIP RESET STATUS
                                        ;;**************************************************************
002152  000004                        TST2:   SCOPE
002154  013700  001342                        MOV     DRCSA,R0        ;GET CSR ADDRESS
002160  013701  001352                        MOV     DRCSC,R1        ;STORE CSRC ADDRESS
002164  005010                        CLR     (R0)            ;INIT CSRA
002166  005011                        CLR     (R1)            ;INIT CSRC
002170  010037  001122                MOV     R0,$BDADR       ;STORE CSR ADDRESS
                                                                ;SET UP EXPECTED DATA
002174  012737  100700  001124        MOV     #RDY!DIR!BIT7!BIT6,$GDDAT
002202  112760  000001  000001        MOVB    #BIT0,1(R0)     ;SET DIRECTION BIT
002210  011015                        MOV     (R0),(R5)       ;GET CSR DATA
002212  042715  000007                BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
002216  023715  001124                CMP     $GDDAT,(R5)
002222  001401                        BEQ     100$
002224  104002                        ERROR   2               ;CSRA ERROR
002226  010137  001122        100$:   MOV     R1,$BDADR       ;STORE CSRC ADDRESS
002232  012737  000200  001124        MOV     #BIT7,$GDDAT    ;STORE EXPECTED
002240  011115                        MOV     (R1),(R5)       ;STORE CSRC
002242  042715  000007                BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
002246  023715  001124                CMP     $GDDAT,(R5)
002252  001401                        BEQ     1$
002254  104002                        ERROR   2               ;CSRC ERROR
002256  010037  001122        1$:     MOV     R0,$BDADR       ;CSRA ADDRESS
002262  012737  100300  001124        MOV     #RDY!BIT7!BIT6,$GDDAT
002270  105060  000001                CLRB    1(R0)           ;CLEAR CSRA DIR BIT
002274  011015                        MOV     (R0),(R5)       ;READ CSR
002276  042715  000007                BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
002302  023715  001124                CMP     $GDDAT,(R5)
002306  001401                        BEQ     2$              ;;BR IF EQUAL
002310  104002                        ERROR   2
002312  012737  000300  001124  2$:   MOV     #BIT7!BIT6,$GDDAT
002320  112761  000001  000001        MOVB    #BIT0,1(R1)     ;CSRC TO OUTPUT MODE
002326  011015                        MOV     (R0),(R5)       ;READ CSRA
002330  042715  000007                BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
002334  023715  001124                CMP     $GDDAT,(R5)     ;CHECK IF RDY BIT CLEARED
002340  001401                        BEQ     TST3            ;;BR IF EQUAL
002342  104002                        ERROR   2               ;CSRA REG ERROR

                                        ;;**************************************************************
                                        ;*TEST 3        TEST CSRA INT. ENABLE BIT
                                        ;;**************************************************************
002344  000004                        TST3:   SCOPE
002346  004737  013002                        JSR     PC,CLRCSR       ;CLEAR CSR REGISTER
002352  013700  001342                        MOV     DRCSA,R0        ;GET CSRA ADDRESS
002356  013701  001352                        MOV     DRCSC,R1        ;GET CSRC ADDRESS
002362  112761  000001  000001        MOVB    #BIT0,1(R1)     ;SET DIRECTION BIT CSRC
002370  010037  001122                MOV     R0,$BDADR       ;STORE CSRA ADDRESS
002374  012737  001300  001124        MOV     #BIT9!BIT7!BIT6,$GDDAT
002402  012710  001000                MOV     #IE,(R0)        ;SET INTERRUPT ENABLE BIT
002406  011015                        MOV     (R0),(R5)
002410  042715  000007                BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
002414  023715  001124                CMP     $GDDAT,(R5)
002420  001401                        BEQ     1$
002422  104002                        ERROR   2               ;INT. ENABLE ERROR,CSRA
002424  012737  000300  001124  1$:   MOV     #BIT7!BIT6,$GDDAT
```

```
565  002432  105060  000001                  CLRB    1(R0)               ;CLEAR I/E BIT
566  002436  011015                          MOV     (R0),(R5)
567  002440  042715  000007                  BIC     #7,(R5)             ;CLEAR UNDEFINED BITS
568  002444  023715  001124                  CMP     $GDDAT,(R5)
569  002450  001401                          BEQ     TST4                ;;BR IF EQUAL
570  002452  104002                          ERROR   2                   ;CSRA ERROR
571
572                                  ;;**********************************************************************
573                                  ;*TEST 4         TEST CSRA I/E,DIR BIT
574                                  ;;**********************************************************************
575  002454  000004                  TST4:   SCOPE
576  002456  004737  013002                  JSR     PC,CLRCSR           ;CLEAR CSR REGISTERS
577  002462  013700  001342                  MOV     DRCSA,R0            ;GET CSRA ADDRESS
578  002466  010037  001122                  MOV     R0,$BDADR           ;SAVE CSRA ADDRESS
579  002472  012737  101700  001124          MOV     #101700,$GDDAT      ;EXPECTED I/E,DIR
580  002500  112760  000003  000001          MOVB    #BIT1!BIT0,1(R0)
581                                                                      ;SET I/E AND DIR BIT
582  002506  011015                          MOV     (R0),(R5)
583  002510  042715  000007                  BIC     #7,(R5)             ;CLEAR UNDEFINED BITS
584  002514  023715  001124                  CMP     $GDDAT,(R5)
585  002520  001401                          BEQ     1$
586  002522  104002                          ERROR   2                   ;CSRA ERROR
587  002524  012737  100300  001124  1$:     MOV     #RDY!BIT7!BIT6,$GDDAT
588  002532  005010                          CLR     (R0)
589  002534  011015                          MOV     (R0),(R5)
590  002536  042715  000007                  BIC     #7,(R5)             ;CLEAR UNDEFINED BITS
591  002542  023715  001124                  CMP     $GDDAT,(R5)
592  002546  001401                          BEQ     TST5                ;BR IF EQUAL
593  002550  104002                          ERROR   2                   ;CSRA ERROR
594
595
596                                  ;;**********************************************************************
597                                  ;*TEST 5         TEST CSRB W/R DIR BIT,INT CHIP RESET STATUS
598                                  ;;**********************************************************************
599  002552  000004                  TST5:   SCOPE
600  002554  004737  013002                  JSR     PC,CLRCSR           ;CLEAR CSR REGISTERS
601  002560  013700  001346                  MOV     DRCSB,R0            ;GET CSR ADDRESS
602  002564  013701  001356                  MOV     DRCSD,R1            ;GET CSRD ADDRESS
603  002570  010037  001122                  MOV     R0,$BDADR           ;STORE CSR ADDRESS
604  002574  012737  100400  001124          MOV     #RDY!DIR,$GDDAT     ;SET UP EXPECTED DATA
605  002602  112760  000001  000001          MOVB    #BIT0,1(R0)         ;SET DIRECTION BIT
606  002610  011015                          MOV     (R0),(R5)           ;GET CSRB DATA
607  002612  023715  001124                  CMP     $GDDAT,(R5)
608  002616  001401                          BEQ     100$
609  002620  104002                          ERROR   2                   ;CSRB ERROR
610  002622  010137  001122          100$:   MOV     R1,$BDADR           ;STORE CSRD ADDRESS
611  002626  005037  001124                  CLR     $GDDAT              ;STORE EXPECTED
612  002632  011115                          MOV     (R1),(R5)           ;READ CSRD
613  002634  023715  001124                  CMP     $GDDAT,(R5)
614  002640  001401                          BEQ     1$
615  002642  104002                          ERROR   2                   ;CSRD ERROR
616  002644  010037  001122          1$:     MOV     R0,$BDADR           ;STORE CSRB ADDRESS
617  002650  012737  100000  001124          MOV     #RDY,$GDDAT         ;STORE EXPECTED
618  002656  105060  000001                  CLRB    1(R0)               ;CLEAR CSR DIR BIT
619  002662  011015                          MOV     (R0),(R5)           ;READ CSR
620  002664  023715  001124                  CMP     $GDDAT,(R5)
```

```
                                        BEQ     2$                      ;;BR IF EQUAL
                                        ERROR   2
                                2$:     CLR     $GDDAT                  ;STORE EXPECTED
                                        MOVB    #BIT0,1(R1)             ;CSRD TO OUTPUT MODE
                                        MOV     (R0),(R5)               ;READ CSRB
                                        CMP     $GDDAT,(R5)             ;RDY BIT CLEARED
                                        BEQ     TST6                    ;;BR IF EQUAL
                                        ERROR   2                       ;CSRB REG ERROR

                        ;;**************************************************************
                        ;*TEST 6        TEST CSRC W/R DIR BIT,INT CHIP RESET STATUS
                        ;;**************************************************************
                        TST6:   SCOPE
                                JSR     PC,CLRCSR               ;CLEAR CSR REGISTERS
                                MOV     DRCSC,R0                ;GET CSR ADDRESS
                                MOV     DRCSA,R1                ;GET CSRA ADDRESS
                                MOV     R0,$GDADR               ;STORE CSR ADDRESS
                                MOV     #RDY!DIR!BIT7,$GDDAT
                                MOVB    #BIT0,1(R0)             ;SET DIRECTION BIT
                                MOV     (R0),(R5)               ;GET CSR DATA
                                BIC     #7,(R5)                 ;CLEAR UNDEFINED BITS
                                CMP     $GDDAT,(R5)
                                BEQ     10$
                                ERROR   2
                        100$:   MOV     R1,$BDADR               ;STORE CSR ADDRESS
                                MOV     #BIT7!BIT6,$GDDAT
                                MOV     (R1),(R5)               ;READ CSRA
                                BIC     #7,(R5)                 ;CLEAR UNDEFINED BITS
                                CMP     $GDDAT,(R5)
                                BEQ     1$                      ;CHECK CSRA
                                ERROR   2                       ;CSRA ERROR
                        1$:     MOV     R0,$BDADR               ;STORE CSR ADDRESS
                                MOV     #RDY!BIT7,$GDDAT        ;STORE EXPECTED DATA
                                CLRB    1(R0)                   ;CLEAR CSR DIR BIT
                                MOV     (R0),(R5)               ;READ CSR
                                BIC     #7,(R5)                 ;CLEAR UNDEFINED BITS
                                CMP     $GDDAT,(R5)
                                BEQ     2$                      ;;BR IF EQUAL
                                ERROR   2
                        2$:     MOV     #BIT7,$GDDAT            ;EXPECTED DATA
                                MOVB    #BIT0,1(R1)             ;CSRA TO OUTPUT MODE
                                MOV     (R0),(R5)               ;READ CSRC
                                BIC     #7,(R5)                 ;CLEAR UNDEFINED BITS
                                CMP     $GDDAT,(R5)             ;RDY BIT CLEARED
                                BEQ     TST7                    ;;BR IF EQUAL
                                ERROR   2                       ;CSRC REG ERROR

                        ;;**************************************************************
                        ;*TEST 7        TEST CSRD W/R DIR BIT,INT CHIP RESET STATUS
                        ;;**************************************************************
                        TST7:   SCOPE
                                JSR     PC,CLRCSR               ;CLEAR CSR REGISTERS
                                MOV     DRCSD,R0                ;GET CSR ADDRESS
                                MOV     DRCSB,R1                ;CSRB ADDRESS
                                MOV     R0,$BDADR               ;STORE CSR ADDRESS
                                MOV     #RDY!DIR,$GDDAT         ;SET UP EXPECTED DATA
```

```
 677  003142  112760  000001  000001          MOVB    #BIT0,1(R0)     ;SET DIRECTION BIT
 678  003150  011015                          MOV     (R0),(R5)       ;GET CSR DATA
 679  003152  023715  001124                  CMP     $GDDAT,(R5)
 680  003156  001401                          BEQ     100$
 681  003160  104002                          ERROR   2
 682  003162  010137  001122          100$:   MOV     R1,$BDADR       ;STORE CSRB ADDRESS
 683  003166  005037  001124                  CLR     $GDDAT          ;STORE EXPECTED
 684  003172  011115                          MOV     (R1),(R5)
 685  003174  023715  001124                  CMP     $GDDAT,(R5)
 686  003200  001401                          BEQ     1$
 687  003202  104002                          ERROR   2               ;CSRB ERROR
 688  003204  010037  001122          1$:     MOV     R0,$BDADR       ;STORE CSRD ADDRESS
 689  003210  012737  100000  001124          MOV     #RDY,$GDDAT     ;STORE EXPECTED
 690  003216  105060  000001                  CLRB    1(R0)           ;CLEAR CSR DIR BIT
 691  003222  011015                          MOV     (R0),(R5)       ;READ CSR
 692  003224  023715  001124                  CMP     $GDDAT,(R5)
 693  003230  001401                          BEQ     2$              ;;BR IF EQUAL
 694  003232  104002                          ERROR   2
 695  003234  005037  001124          2$:     CLR     $GDDAT          ;EXPECTED
 696  003240  112761  000001  000001          MOVB    #BIT0,1(R1)     ;CSRD TO OUTPUT MODE
 697  003246  011015                          MOV     (R0),(R5)       ;READ CSR
 698  003250  023715  001124                  CMP     $GDDAT,(R5)     ;RDY BIT CLEARED
 699  003254  001401                          BEQ     TST10           ;;BR IF EQUAL
 700  003256  104002                          ERROR   2               ;CSRD REG ERROR
 701
 702                                  ;;*****************************************************************
 703                                  ;*TEST 10       TEST DBRA W/R IN OUTPUT MODE
 704                                  ;;*****************************************************************
 705  003260  000004                  TST10:  SCOPE
 706  003262  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
 707  003266  012703  013110                  MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
 708  003272  012737  003322  001110          MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
 709  003300  013700  001342                  MOV     DRCSA,R0        ;GET CSRA
 710  003304  013701  001344                  MOV     DRDBA,R1        ;GET DBRA ADDRESS
 711  003310  010137  001122                  MOV     R1,$BDADR       ;STORE DBRA ADDRESS
 712  003314  112760  000001  000001          MOVB    #BIT0,1(R0)     ;SET CSRA IN OUTPUT MODE
 713  003322  011337  001124          1$:     MOV     (R3),$GDDAT     ;SAVE EXPECTED DATA
 714  003326  005011                          CLR     (R1)            ;CLEAR DBRA
 715  003330  051311                          BIS     (R3),(R1)       ;WRITE INTO DBRA
 716  003332  011115                          MOV     (R1),(R5)       ;READ DBRA
 717  003334  023715  001124                  CMP     $GDDAT,(R5)     ;CHECK W/R DBRA
 718  003340  001401                          BEQ     2$              ;NEXT PAT. IF EQUAL
 719  003342  104002                          ERROR   2               ;DBRA W/R ERROR
 720  003344  005723                  2$:     TST     (R3)+           ;INC FOR NEXT PATTERN
 721  003346  020327  013410                  CMP     R3,#ENDDAT      ;CHECK FOR END
 722  003352  001363                          BNE     1$              ;DO NEXT PATTERN
 723
 724                                  ;;*****************************************************************
 725                                  ;*TEST 11       TEST DBRB W/R IN OUTPUT MODE
 726                                  ;;*****************************************************************
 727  003354  000004                  TST11:  SCOPE
 728  003356  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
 729  003362  012703  013110                  MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
 730  003366  012737  003416  001110          MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
 731  003374  013700  001346                  MOV     DRCSB,R0        ;GET CSRB
 732  003400  013701  001350                  MOV     DRDBB,R1        ;GET DBRB ADDRESS
```

```
  733  003404  010137  001122                    MOV     R1,$BDADR       ;STORE DBRB ADDRESS
  734  003410  112760  000001  000001            MOVB    #BIT0,1(R0)     ;SET CSRB IN OUTPUT MODE
  735  003416  011337  001124          1$:       MOV     (R3),$GDDAT     ;SAVE EXPECTED DATA
  736  003422  005011                            CLR     (R1)            ;CLEAR DBRB
  737  003424  051311                            BIS     (R3),(R1)       ;WRITE INTO DBRB
  738  003426  011115                            MOV     (R1),(R5)       ;READ DBRB
  739  003430  023715  001124                    CMP     $GDDAT,(R5)     ;CHECK W/R DBRB
  740  003434  001401                            BEQ     2$              ;NEXT PAT. IF EQUAL
  741  003436  104002                            ERROR   2               ;DBRB W/R ERROR
  742  003440  005723                  2$:       TST     (R3)+           ;INC FOR NEXT PATTERN
  743  003442  020327  013410                    CMP     R3,#ENDDAT      ;CHECK FOR END
  744  003446  001363                            BNE     1$              ;DO NEXT PATTERN
  745
  746
  747                                    ;;*******************************************************************
  748                                    ;*TEST 12        TEST DBRC W/R IN OUTPUT MODE
  749                                    ;;*******************************************************************
  750  003450  000004                    TST12:  SCOPE
  751  003452  004737  013002                    JSR     PC,CLRCSR       ;CLEAR ALL CSRS
  752  003456  012703  013110                    MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
  753  003462  012737  003512  001110            MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
  754  003470  013700  001352                    MOV     DRCSC,R0        ;GET CSRC
  755  003474  013701  001354                    MOV     DRDBC,R1        ;GET DBRC ADDRESS
  756  003500  010137  001122                    MOV     R1,$BDADR       ;STORE DBRC ADDRESS
  757  003504  112760  000001  000001            MOVB    #BIT0,1(R0)     ;SET CSRC IN OUTPUT MODE
  758  003512  011337  001124          1$:       MOV     (R3),$GDDAT     ;SAVE EXPECTED DATA
  759  003516  005011                            CLR     (R1)            ;CLEAR DBRC
  760  003520  051311                            BIS     (R3),(R1)       ;WRITE INTO DBRC
  761  003522  011115                            MOV     (R1),(R5)       ;READ DBRC
  762  003524  023715  001124                    CMP     $GDDAT,(R5)     ;CHECK W/R DBRC
  763  003530  001401                            BEQ     2$              ;NEXT PAT. IF EQUAL
  764  003532  104002                            ERROR   2               ;DBRC W/R ERROR
  765  003534  005723                  2$:       TST     (R3)+           ;INC FOR NEXT PATTERN
  766  003536  020327  013410                    CMP     R3,#ENDDAT      ;CHECK FOR END
  767  003542  001363                            BNE     1$              ;DO NEXT PATTERN
  768
  769
  770                                    ;;*******************************************************************
  771                                    ;*TEST 13        TEST DBRD W/R IN OUTPUT MODE
  772                                    ;;*******************************************************************
  773  003544  000004                    TST13:  SCOPE
  774  003546  004737  013002                    JSR     PC,CLRCSR       ;CLEAR ALL CSRS
  775  003552  012703  013110                    MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
  776  003556  012737  003606  001110            MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
  777  003564  013700  001356                    MOV     DRCSD,R0        ;GET CSRD
  778  003570  013701  001360                    MOV     DRDBD,R1        ;GET DBRD ADDRESS
  779  003574  010137  001122                    MOV     R1,$BDADR       ;STORE DBRD ADDRESS
  780  003600  112760  000001  000001            MOVB    #BIT0,1(R0)     ;SET CSRD IN OUTPUT MODE
  781  003606  011337  001124          1$:       MOV     (R3),$GDDAT     ;SAVE EXPECTED DATA
  782  003612  005011                            CLR     (R1)            ;CLEAR DBRD
  783  003614  051311                            BIS     (R3),(R1)       ;WRITE INTO DBRD
  784  003616  011115                            MOV     (R1),(R5)       ;READ DBRD
  785  003620  023715  001124                    CMP     $GDDAT,(R5)     ;CHECK W/R DBRD
  786  003624  001401                            BEQ     2$              ;NEXT PAT. IF EQUAL
  787  003626  104002                            ERROR   2               ;DBRD W/R ERROR
  788  003630  005723                  2$:       TST     (R3)+           ;INC FOR NEXT PATTERN
```

```
   789  003632  020327  013410                  CMP     R3,#ENDDAT      ;CHECK FOR END
   790  003636  001363                          BNE     1$              ;DO NEXT PATTERN
```

```
                                ;;*********************************************************************
                                ;*TEST 14        TEST CSR UNIQUENESS,CSRS (A-B),(C-D)
                                ;;*********************************************************************
003640  000004                  TST14:  SCOPE
003642  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
                                                                ;CSRA = R0
                                                                ;CSRB = R1
                                                                ;CSRC = R2
                                                                ;CSRD = R3
003646  112760  000003  000001          MOVB    #3,1(R0)        ;CSRA OUTPUT MODE,I/E
003654  112763  000001  000001          MOVB    #BIT0,1(R3)     ;CSRD OUTPUT MODE
003662  010037  001122                  MOV     R0,$BDADR
003666  012737  101700  001124          MOV     #101700,$GDDAT  ;EXPECTED DATA
003674  011015                          MOV     (R0),(R5)
003676  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
003702  023715  001124                  CMP     $GDDAT,(R5)
003706  001401                          BEQ     1$
003710  104002                          ERROR   2               ;CSRA ERROR
003712  010137  001122          1$:     MOV     R1,$BDADR       ;CHECK CSRB
003716  005037  001124                  CLR     $GDDAT
003722  011115                          MOV     (R1),(R5)
003724  023715  001124                  CMP     $GDDAT,(R5)
003730  001401                          BEQ     2$
003732  104002                          ERROR   2               ;CSRB ERROR
003734  010337  001122          2$:     MOV     R3,$BDADR
003740  012737  100400  001124          MOV     #100400,$GDDAT
003746  011315                          MOV     (R3),(R5)
003750  023715  001124                  CMP     $GDDAT,(R5)
003754  001401                          BEQ     3$
003756  104002                          ERROR   2               ;CSRD ERROR
003760  010237  001122          3$:     MOV     R2,$BDADR       ;CHECK CSRC
003764  012737  000200  001124          MOV     #BIT7,$GDDAT    ;EXPECTED
003772  011215                          MOV     (R2),(R5)
003774  042715  000007                  BIC     #7,(R5)
004000  023715  001124                  CMP     $GDDAT,(R5)
004004  001401                          BEQ     TST15           ;;BR IF EQUAL
004006  104002                          ERROR   2               ;CSRC ERROR

                                ;;*********************************************************************
                                ;*TEST 15        TEST CSR UNIQUENESS,CSRS (A-D),(C-B)
                                ;;*********************************************************************
004010  000004                  TST15:  SCOPE
004012  004737  013002                  JSR     PC,CLRCSR       ;CLR ALL CSRS
                                                                ;CSRA = R0
                                                                ;CSRB = R1
                                                                ;CSRC = R2
                                                                ;CSRD = R3
004016  112760  000003  000001          MOVB    #3,1(R0)        ;CSRA OUTPUT,I/E
004024  112761  000001  000001          MOVB    #BIT0,1(R1)     ;CSRB OUTPUT MODE
004032  010037  001122                  MOV     R0,$BDADR       ;CSRA ADDRESS
004036  012737  101700  001124          MOV     #101700,$GDDAT
004044  011015                          MOV     (R0),(R5)
004046  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
004052  023715  001124                  CMP     $GDDAT,(R5)
```

```
  847  004056  001400                          BEQ     1$
  848  004060  010137  001122            1$:   MOV     R1,$BDADR       ;CHECK CSRB
  849  004064  012737  100400  001124          MOV     #100400,$GDDAT
  850  004072  011115                          MOV     (R1),(R5)
  851  004074  023715  001124                  CMP     $GDDAT,(R5)
  852  004100  001401                          BEQ     2$
  853  004102  104002                          ERROR   2               ;CSRB ERROR
  854  004104  010237  001122            2$:   MOV     R2,$BDADR       ;CHECK CSRC
  855  004110  012737  000200  001124          MOV     #BIT7,$GDDAT
  856  004116  011215                          MOV     (R2),(R5)
 *857  004120  042715  000007                  BIC     #7,(R5)
  858  004124  023715  001124                  CMP     $GDDAT,(R5)
  859  004130  001401                          BEQ     3$
  860  004132  104002                          ERROR   2               ;CSRC ERROR
  861  004134  010337  001122            3$:   MOV     R3,$BDADR
  862  004140  005037  001124                  CLR     $GDDAT
  863  004144  011315                          MOV     (R3),(R5)
  864  004146  023715  001124                  CMP     $GDDAT,(R5)
  865  004152  001401                          BEQ     TST16           ;;BR IF EQUAL
  866  004154  104002                          ERROR   2               ;CSRD ERROR
  867
  868
  869                                  ;;*****************************************************************
  870                                  ;*TEST 16       TEST PORT A TO PORT C INTERACTION
  871                                  ;;*****************************************************************
  872  004156  000004                  TST16:  SCOPE
  873  004160  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
  874  004164  012703  013110                  MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
  875  004170  012737  004220  001110          MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
  876  004176  013700  001342                  MOV     DRCSA,R0        ;GET PORT A, CSRA ADDRESS
  877  004202  013701  001352                  MOV     DRCSC,R1        ;GET PORT C, CSRC ADDRESS
  878  004206  013702  001346                  MOV     DRCSB,R2        ;CSRB ADDRESS
  879  004212  112762  000001  000001          MOVB    #BIT0,1(R2)     ;CSRB IN OUTPUT MODE
  880  004220  010037  001122            1$:   MOV     R0,$BDADR       ;STORE CSRA ADDRESS
  881  004224  005062  000002                  CLR     2(R2)           ;CLEAR DBRB
  882  004230  105061  000001                  CLRB    1(R1)           ;PORT C,CSRC,INPUT MODE
  883  004234  112760  000001  000001          MOVB    #BIT0,1(R0)     ;SET CSRA IN OUTPUT MODE
  884  004242  011360  000002                  MOV     (R3),2(R0)      ;WRITE INTO DBRA
  885  004246  016104  000002                  MOV     2(R1),R4        ;READ DBRC
  886  004252  012737  100700  001124          MOV     #RDY!DIR!BIT7!BIT6,$GDDAT
  887                                                                  ;CSRA DIR SHOULD STAY SET
  888  004260  011015                          MOV     (R0),(R5)       ;READ CSRA
  889  004262  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
  890  004266  023715  001124                  CMP     $GDDAT,(R5)
  891  004272  001401                          BEQ     100$
  892  004274  104002                          ERROR   2               ;CSRA ERROR
  893  004276  012737  000200  001124  100$:   MOV     #BIT7,$GDDAT
  894  004304  010137  001122                  MOV     R1,$BDADR       ;CSRC ADDRESS
  895  004310  011115                          MOV     (R1),(R5)       ;READ CSRC
  896  004312  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
  897  004316  023715  001124                  CMP     $GDDAT,(R5)
  898  004322  001401                          BEQ     101$
  899  004324  104002                          ERROR   2               ;CSRC ERROR
  900  004326  013737  001354  001122  101$:   MOV     DRDBC,$BDADR    ;STORE DBRC ADDRESS
  901  004334  011337  001124                  MOV     (R3),$GDDAT     ;SAVE EXPECTED
  902  004340  010415                          MOV     R4,(R5)         ;DBRC CONTENTS
```

```
 903  004342  023715  001124                  CMP     $CDDAT,(R5)     ;CHECK PORT C,DBRC
 904  004346  001401                          BEQ     2$              ;BEQ TO NEXT SUBTEST
 905  004350  104002                          ERROR   2               ;DBRC REG ERROR
 906  004352  005137  001124          2$:     COM     $GDDAT          ;SET UP TO WRITE COM DATA FROM
 907                                                                  ;PORT C TO PORT A
 908  004356  013737  001344  001122          MOV     DRDBA,$BDADR    ;STORE DBRA ADDRESS
 909  004364  013761  001124  000002  3$:     MOV     $GDDAT,2(R1)    ;WRITE PORT C,INPUT MODE
 910  004372  105060  000001                  CLRB    1(R0)           ;MAKE PORT A INPUT MODE
 911  004376  112761  000001  000001          MOVB    #BIT0,1(R1)     ;MAKE PORT C OUTPUT MODE
 912  004404  016015  000002                  MOV     2(R0),(R5)      ;READ PORT A,DBRA
 913  004410  023715  001124                  CMP     $GDDAT,(R5)     ;PORT A =PORT C?
 914  004414  001401                          BEQ     4$              ;CHECK DBRB FOR NO DATA CHANGE
 915  004416  104002                          ERROR   2               ;PORT A,DBRA REG ERROR
 916  004420  013737  001350  001122  4$:     MOV     DRDBB,$BDADR    ;STORE DBRB ADDRESS
 917  004426  005037  001124                  CLR     $GDDAT          ;CONTENTS SHOULD STAY ZERO
 918  004432  016215  000002                  MOV     2(R2),(R5)      ;READ DBRB FOR CLEAR
 919  004436  023715  001124                  CMP     $GDDAT,(R5)     ;CHECK FOR DBRB CLEAR
 920  004442  001401                          BEQ     5$              ;YES,CONTINUE
 921  004444  104002                          ERROR   2               ;DBRB INTERACTION ERROR
 922  004446  005723                  5$:     TST     (R3)+           ;INC FOR NEXT PATTERN
 923  004450  020327  013410                  CMP     R3,#ENDDAT      ;CHECK FOR END
 924  004454  001261                          BNE     1$              ;DO NEXT PATTERN
 925
 926
 927                                  ;;*****************************************************************
 928                                  ;*TEST 17       TEST PORT B TO PORT D INTERACTION
 929                                  ;;*****************************************************************
 930  004456  000004                  TST17:  SCOPE
 931  004460  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
 932  004464  012703  013110                  MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
 933  004470  012737  004520  001110          MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
 934  004476  013700  001346                  MOV     DRCSB,R0        ;GET PORT B, CSRB ADDRESS
 935  004502  013701  001356                  MOV     DRCSD,R1        ;GET PORT D, CSRD ADDRESS
 936  004506  013702  001342                  MOV     DRCSA,R2        ;STORE CSRA ADDRESS
 937  004512  112762  000001  000001          MOVB    #BIT0,1(R2)     ;CSRA IN OUTPUT MODE
 938  004520  010037  001122          1$:     MOV     R0,$BDADR       ;STORE CSRB ADDRESS
 939  004524  005062  000002                  CLR     2(R2)           ;CLEAR DBRA
 940  004530  105061  000001                  CLRB    1(R1)           ;PORT D,CSRD,INPUT MODE
 941  004534  112760  000001  000001          MOVB    #BIT0,1(R0)     ;SET CSRB IN OUTPUT MODE
 942  004542  011360  000002                  MOV     (R3),2(R0)      ;WRITE INTO DBRB
 943  004546  016104  000002                  MOV     2(R1),R4        ;DBRD CONTENTS
 944  004552  012737  100400  001124          MOV     #RDY!DIR,$GDDAT ;SAVE EXPECTED
 945  004560  011015                          MOV     (R0),(R5)       ;READ CSRB
 946  004562  023715  001124                  CMP     $GDDAT,(R5)
 947  004566  001401                          BEQ     100$
 948  004570  104002                          ERROR   2               ;CSRB ERROR
 949  004572  005037  001124          100$:   CLR     $GDDAT          ;SAVE EXPECTED
 950  004576  010137  001122                  MOV     R1,$BDADR       ;SAVE CSRD ADDRESS
 951  004602  011115                          MOV     (R1),(R5)
 952  004604  023715  001124                  CMP     $GDDAT,(R5)
 953  004610  001401                          BEQ     101$
 954  004612  104002                          ERROR   2               ;CSRD ERROR
 955  004614  013737  001360  001122  101$:   MOV     DRDBD,$BDADR    ;SAVE DBRD ADDRESS
 956  004622  011337  001124                  MOV     (R3),$GDDAT     ;SAVE EXPECTED
 957  004626  010415                          MOV     R4,(R5)         ;DBRD CONTENTS
 958  004630  023715  001124                  CMP     $GDDAT,(R5)     ;CHECK PORT D,DBRD
```

```
 959  004634  001401                          BEQ     2$              ;BEQ TO NEXT SUBTEST
 960  004636  104002                          ERROR   2               ;DBRD REG ERROR
 961  004640  005137  001124          2$:     COM     $GDDAT          ;SET UP TO WRITE COM DATA FROM
 962                                                                  ;PORT D TO PORT B
 963  004644  013737  001350  001122          MOV     DRDBB,$BDADR    ;STORE DBRB ADDRESS
 964  004652  013761  001124  000002  3$:     MOV     $GDDAT,2(R1)    ;WRITE PORT D,INPUT MODE
 965  004660  105060  000001                  CLRB    1(R0)           ;MAKE PORT B INPUT MODE
 966  004664  112761  000001  000001          MOVB    #BIT0,1(R1)     ;MAKE PORT D OUTPUT MODE
 967  004672  016015  000002                  MOV     2(R0),(R5)      ;READ PORT B,DBRB
 968  004676  023715  001124                  CMP     $GDDAT,(R5)     ;PORT B =PORT D?
 969  004702  001401                          BEQ     4$              ;CHECK DBRA FOR NO DATA CHANGE
 970  004704  104002                          ERROR   2               ;PORT B,DBRB REG ERROR
 971  004706  013737  001344  001122  4$:     MOV     DRDBA,$BDADR    ;STORE DBRA ADDRESS
 972  004714  005037  001124                  CLR     $GDDAT          ;CONTENTS = 0
 973  004720  016215  000002                  MOV     2(R2),(R5)      ;READ DBRA FOR CLEAR
 974  004724  023715  001124                  CMP     $GDDAT,(R5)     ;DBRA CLEAR?
 975  004730  001401                          BEQ     5$              ;YES,CONTINUE
 976  004732  104002                          ERROR   2               ;DBRA INTERACTION ERROR
 977  004734  005723                  5$:     TST     (R3)+           ;INC FOR NEXT PATTERN
 978  004736  020327  013410                  CMP     R3,#ENDDAT      ;CHECK FOR END
 979  004742  001266                          BNE     1$              ;DO NEXT PATTERN
 980
 981
 982                                  ;;*******************************************************************
 983                                  ;*TEST 20        TEST PORT C TO PORT A INTERACTION
 984                                  ;;*******************************************************************
 985  004744  000004                  TST20:  SCOPE
 986  004746  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
 987  004752  012703  013110                  MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
 988  004756  012737  005006  001110          MOV     #1$,$LPERR      ;SET UP SCOPE ADDRESS
 989  004764  013700  001352                  MOV     DRCSC,R0        ;GET PORT C, CSRC ADDRESS
 990  004770  013701  001342                  MOV     DRCSA,R1        ;GET PORT A, CSRA ADDRESS
 991  004774  013702  001356                  MOV     DRCSD,R2        ;STORE CSRD ADDRESS
 992  005000  112762  000001  000001          MOVB    #BIT0,1(R2)     ;CSRD IN OUTPUT MODE
 993  005006  010037  001122          1$:     MOV     R0,$BDADR       ;STORE CSRC ADDRESS
 994  005012  005062  000002                  CLR     2(R2)           ;CLEAR DBRD
 995  005016  105061  000001                  CLRB    1(R1)           ;PORT A,CSRA,INPUT MODE
 996  005022  112760  000001  000001          MOVB    #BIT0,1(R0)     ;SET CSRC IN OUTPUT MODE
 997  005030  011360  000002                  MOV     (R3),2(R0)      ;WRITE INTO DBRC
 998  005034  016104  000002                  MOV     2(R1),R4        ;READ DBRA
 999  005040  012737  100600  001124          MOV     #RDY!DIR!BIT7,$GDDAT
1000                                                                  ;CSRC DIR SHOULD BE SET
1001  005046  011015                          MOV     (R0),(R5)       ;READ CSRC
1002  005050  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
1003  005054  023715  001124                  CMP     $GDDAT,(R5)
1004  005060  001401                          BEQ     100$
1005  005062  104002                          ERROR   2               ;CSRC ERROR
1006  005064  012737  000300  001124  100$:   MOV     #BIT7!BIT6,$GDDAT
1007  005072  010137  001122                  MOV     R1,$BDADR       ;CSRA ADDRESS
1008  005076  011115                          MOV     (R1),(R5)       ;READ CSRA
1009  005100  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
1010  005104  023715  001124                  CMP     $GDDAT,(R5)
1011  005110  001401                          BEQ     101$
1012  005112  104002                          ERROR   2               ;CSRA ERROR
1013  005114  013737  001344  001122  101$:   MOV     DRDBA,$BDADR    ;STORE DBRA ADDRESS
1014  005122  011337  001124                  MOV     (R3),$GDDAT     ;SAVE EXPECTED
```

```
1015  005126  010415                          MOV     R4,(R5)          ;DBRA CONTENTS
1016  005130  023715  001124                  CMP     $GDDAT,(R5)      ;CHECK PORT A,DBRA
1017  005134  001401                          BEQ     2$               ;BEQ TO NEXT SUBTEST
1018  005136  104002                          ERROR   2                ;DBRA REG ERROR
1019  005140  005137  001124          2$:     COM     $GDDAT           ;SET UP TO WRITE COM DATA FROM
1020                                                                   ;PORT A TO PORT C
1021  005144  013737  001354  001122          MOV     DRDBC,$BDADR     ;STORE DBRC ADDRESS
1022  005152  013761  001124  000002  3$:     MOV     $GDDAT,2(R1)     ;WRITE PORT A,INPUT MODE
1023  005160  105060  000001                  CLRB    1(R0)            ;MAKE PORT C INPUT MODE
1024  005164  112761  000001  000001          MOVB    #BIT0,1(R1)      ;MAKE PORT A OUTPUT MODE
1025  005172  016015  000002                  MOV     2(R0),(R5)       ;READ PORT C,DBRC
1026  005176  023715  001124                  CMP     $GDDAT,(R5)      ;PORT C =PORT A?
1027  005202  001401                          BEQ     4$               ;CHECK DBRD FOR NO DATA CHANGE
1028  005204  104002                          ERROR   2                ;PORT C,DBRC REG ERROR
1029  005206  013737  001360  001122  4$:     MOV     DRDBD,$BDADR     ;STORE DBRD ADDRESS
1030  005214  005037  001124                  CLR     $GDDAT           ;DBRD = 0 EXPECTED
1031  005220  016215  000002                  MOV     2(R2),(R5)       ;READ DBRD FOR CLEAR
1032  005224  023715  001124                  CMP     $GDDAT,(R5)      ;IS DBRD CLEAR?
1033  005230  001401                          BEQ     5$               ;YES,CONTINUE
1034  005232  104002                          ERROR   2                ;DBRD INTERACTION ERROR
1035  005234  005723                  5$:     TST     (R3)+            ;INC FOR NEXT PATTERN
1036  005236  020327  013410                  CMP     R3,#ENDDAT       ;CHECK FOR END
1037  005242  001261                          BNE     1$               ;DO NEXT PATTERN
1038
1039                                  ;;*******************************************************************
1040                                  ;*TEST 21        TEST PORT D TO PORT B INTERACTION
1041                                  ;;*******************************************************************
1042  005244  000004                  TST21:  SCOPE
1043  005246  004737  013002                  JSR     PC,CLRCSR        ;CLEAR ALL CSRS
1044  005252  012703  013110                  MOV     #BEGPAT,R3       ;GET DATA PATTERN TABLE
1045  005256  012737  005306  001110          MOV     #1$,$LPERR       ;SET UP SCOPE ADDRESS
1046  005264  013700  001356                  MOV     DRCSD,R0         ;GET PORT D, CSRD ADDRESS
1047  005270  013701  001346                  MOV     DRCSB,R1         ;GET PORT B, CSRB ADDRESS
1048  005274  013702  001352                  MOV     DRCSC,R2         ;STORE CSRC ADDRESS
1049  005300  112762  000001  000001          MOVB    #BIT0,1(R2)      ;CSRC IN OUTPUT MODE
1050  005306  010037  001122          1$:     MOV     R0,$BDADR        ;STORE CSRD ADDRESS
1051  005312  005062  000002                  CLR     2(R2)            ;CLEAR DBRC
1052  005316  105061  000001                  CLRB    1(R1)            ;PORT B,CSRB,INPUT MODE
1053  005322  112760  000001  000001          MOVB    #BIT0,1(R0)      ;SET CSRD IN OUTPUT MODE
1054  005330  011360  000002                  MOV     (R3),2(R0)       ;WRITE INTO DBRD
1055  005334  016104  000002                  MOV     2(R1),R4         ;READ DBRB
1056  005340  012737  100400  001124          MOV     #RDY!DIR,$GDDAT  ;SAVE EXPECTED
1057  005346  011015                          MOV     (R0),(R5)        ;READ CSRD
1058  005350  023715  001124                  CMP     $GDDAT,(R5)
1059  005354  001401                          BEQ     100$
1060  005356  104002                          ERROR   2                ;CSRD ERROR
1061  005360  005037  001124          100$:   CLR     $GDDAT           ;SAVE EXPECTED
1062  005364  010137  001122                  MOV     R1,$BDADR        ;SAVE CSRB ADDRESS
1063  005370  011115                          MOV     (R1),(R5)
1064  005372  023715  001124                  CMP     $GDDAT,(R5)
1065  005376  001401                          BEQ     101$
1066  005400  104002                          ERROR   2                ;CSRB ERROR
1067  005402  013737  001350  001122  101$:   MOV     DRDBB,$BDADR     ;SAVE DBRB ADDRESS
1068  005410  011337  001124                  MOV     (R3),$GDDAT      ;SAVE EXPECTED
1069  005414  010415                          MOV     R4,(R5)          ;DBRB CONTENTS
1070  005416  023715  001124                  CMP     $GDDAT,(R5)      ;CHECK PORT B,DBRB
```

```
1071  005422  001401                        BEQ     2$              ;BEQ TO NEXT SUBTEST
1072  005424  104002                        ERROR   2               ;DBRB REG ERROR
1073  005426  005137  001124          2$:   COM     $GDDAT          ;SET UP TO WRITE COM DATA FROM
1074                                                                ;PORT B TO PORT D
1075  005432  013737  001360  001122        MOV     DRDBD,$BDADR    ;STORE  DBRD ADDRESS
1076  005440  013761  001124  000002  3$:   MOV     $GDDAT,2(R1)    ;WRITE PORT B,INPUT MODE
1077  005446  105060  000001                CLRB    1(R0)           ;MAKE PORT D INPUT MODE
1078  005452  112761  000001  000001        MOVB    #BIT0,1(R1)     ;MAKE PORT B OUTPUT MODE
1079  005460  016015  000002                MOV     2(R0),(R5)      ;READ PORT D,DBRD
1080  005464  023715  001124                CMP     $GDDAT,(R5)     ;PORT B =PORT D?
1081  005470  001401                        BEQ     4$              ;CHECK DBRC FOR NO DATA CHANGE
1082  005472  104002                        ERROR   2               ;PORT D,DBRD REG ERROR
1083  005474  013737  001354  001122  4$:   MOV     DRDBC,$BDADR    ;STORE DBRC ADDRESS
1084  005502  005037  001124                CLR     $GDDAT          ;EXPECT DBRC = 0
1085  005506  016215  000002                MOV     2(R2),(R5)      ;READ DBRC
1086  005512  023715  001124                CMP     $GDDAT,(R5)     ;IS DBRC = 0
1087  005516  001401                        BEQ     5$              ;YES,CONTINUE
1088  005520  104002                        ERROR   2               ;DBRC INTERACTION ERROR
1089  005522  005723                  5$:   TST     (R3)+           ;INC FOR NEXT PATTERN
1090  005524  020327  013410                CMP     R3,#ENDDAT      ;CHECK FOR END
1091  005530  001266                        BNE     1$              ;DO NEXT PATTERN
1092
1093                                        ;;**************************************************************
1094                                        ;*TEST 22       TEST GROUP 1,2 IMR,IRR,ACR WITH CHIP RESET
1095                                        ;;**************************************************************
1096  005532  000004                  TST22: SCOPE
1097  005534  004737  013002                JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1098  005540  012702  000002                MOV     #2,R2           ;TWO GROUPS TO TEST
1099  005544  013700  001342                MOV     DRCSA,R0        ;STORE CSRA
1100  005550  013701  001346                MOV     DRCSB,R1        ;STORE CSRB
1101  005554  004737  013052                JSR     PC,CLRIRR       ;CLEAR IRR REGISTERS
1102  005560  010137  001122          111$: MOV     R1,$BDADR       ;STORE ADDRESS
1103  005564  012737  000377  001124        MOV     #377,$GDDAT     ;STORE EXPECTED DATA
1104  005572  112710  000244                MOVB    #MIMR,(R0)      ;LOAD MODE BITS FOR IMR
1105  005576  111115                        MOVB    (R1),(R5)       ;READ IMR REGISTER
1106  005600  023715  001124                CMP     $GDDAT,(R5)
1107  005604  001401                        BEQ     1$
1108  005606  104005                        ERROR   5               ;IMR REG ERROR
1109  005610  005037  001124            1$: CLR     $GDDAT          ;STORE EXPECTED
1110  005614  112710  000250                MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
1111  005620  111115                        MOVB    (R1),(R5)       ;READ IRR REGISTER
1112  005622  023715  001124                CMP     $GDDAT,(R5)
1113  005626  001401                        BEQ     2$
1114  005630  104003                        ERROR   3               ;IRR REG ERROR
1115  005632  112710  000254            2$: MOVB    #MACR,(R0)      ;LOAD MODE BITS FOR ACR
1116  005636  111115                        MOVB    (R1),(R5)       ;READ ACR REGISTER
1117  005640  023715  001124                CMP     $GDDAT,(R5)
1118  005644  001401                        BEQ     3$
1119  005646  104004                        ERROR   4               ;ACR REG ERROR
1120  005650  005302                    3$: DEC     R2              ;FINISHED BOTH GROUPS?
1121  005652  001405                        BEQ     TST23           ;;BR IF EQUAL
1122  005654  013700  001352                MOV     DRCSC,R0
1123  005660  013701  001356                MOV     DRCSD,R1
1124  005664  000735                        BR      111$
1125
1126                                        ;;**************************************************************
```

```
1127                                        ;*TEST 23       TEST GROUPS 1 AND 2 ACR UNIQUENESS
1128                                        ;;*****************************************************************
1129  005666  000004                        TST23:  SCOPE
1130  005670  004737  013002                        JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1131  005674  013700  001342                        MOV     DRCSA,R0        ;CSRA ADDRESS
1132  005700  013701  001346                        MOV     DRCSB,R1        ;CSRB ADDRESS
1133  005704  012703  000002                        MOV     #2,R3           ;COUNTER FOR TESTING TWO GROUPS
1134  005710  004737  013052                        JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1135  005714  010137  001122                111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1136  005720  012737  000252  001124                MOV     #252,$GDDAT     ;STORE EXPECTED
1137  005726  112710  000254                        MOVB    #MACR,(R0)      ;LOAD MODE BITS FOR ACR
1138  005732  112710  000300                        MOVB    #PACR,(R0)      ;PRESELECT ACR FOR WRITING
1139  005736  113711  001124                        MOVB    $GDDAT,(R1)     ;WRITE INTO DATA PORT
1140  005742  111115                              MOVB    (R1),(R5)       ;STORE DATA FOR COMPARE
1141  005744  023715  001124                        CMP     $GDDAT,(R5)     ;CHECK ACR RESULTS
1142  005750  001401                              BEQ     1$              ;
1143  005752  104004                              ERROR   4               ;ACR ERROR
1144  005754  112710  000244                1$:    MOVB    #MIMR,(R0)      ;CHANGE TO IMR REGISTER
1145  005760  012737  000377  001124                MOV     #377,$GDDAT     ;STORE EXPECTED
1146  005766  111115                              MOVB    (R1),(R5)       ;READ IMR
1147  005770  023715  001124                        CMP     $GDDAT,(R5)     ;SHOULD STILL BE ALL 1'S
1148  005774  001401                              BEQ     2$
1149  005776  104005                              ERROR   5               ;IMR REG ERROR
1150  006000  112710  000240                2$:    MOVB    #MISR,(R0)      ;LOAD MODE BITS FOR ISR
1151  006004  005037  001124                        CLR     $GDDAT          ;STORE EXPECTED
1152  006010  111115                              MOVB    (R1),(R5)       ;READ ISR
1153  006012  023715  001124                        CMP     $GDDAT,(R5)     ;ISR SHOULD BE CLEARED
1154  006016  001401                              BEQ     3$
1155  006020  104006                              ERROR   6               ;ISR REG ERROR
1156  006022  112710  000250                3$:    MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
1157  006026  111115                              MOVB    (R1),(R5)       ;READ IRR
1158  006030  023715  001124                        CMP     $GDDAT,(R5)     ;IRR SHOULD BE CLEARED
1159  006034  001401                              BEQ     4$
1160  006036  104003                              ERROR   3               ;IRR REG ERROR
1161  006040  105010                          4$:    CLRB    (R0)            ;CHIP RESET GROUP 1
1162  006042  112710  000254                        MOVB    #MACR,(R0)      ;LOAD MODE BITS FOR ACR.
1163  006046  111115                              MOVB    (R1),(R5)       ;READ ACR
1164  006050  023715  001124                        CMP     $GDDAT,(R5)     ;ACR SHOULD BE CLEARED
1165  006054  001401                              BEQ     5$
1166  006056  104004                              ERROR   4               ;ACR REG ERROR
1167  006060  005303                          5$:    DEC     R3              ;TEST GROUPS 1 AND 2
1168  006062  001405                              BEQ     TST24           ;;BR IF BOTH GROUPS TESTED
1169  006064  013700  001352                        MOV     DRCSC,R0        ;GROUP 2 CONTROL PORT
1170  006070  013701  001356                        MOV     DRCSD,R1        ;GROUP 2 DATA PORT
1171  006074  000707                              BR      111$            ;TEST GROUP 2 ACR UNIQUENESS
1172
1173                                        ;;*****************************************************************
1174                                        ;*TEST 24       TEST GROUPS 1 AND 2 IMR UNIQUENESS
1175                                        ;;*****************************************************************
1176  006076  000004                        TST24:  SCOPE
1177  006100  004737  013002                        JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1178  006104  013700  001342                        MOV     DRCSA,R0        ;CSRA ADDRESS
1179  006110  013701  001346                        MOV     DRCSB,R1        ;CSRB ADDRESS
1180  006114  012703  000002                        MOV     #2,R3           ;COUNTER FOR TWO GROUP TESTING
1181  006120  010137  001122                111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1182  006124  012737  000252  001124                MOV     #252,$GDDAT     ;STORE EXPECTED
```

```
 1183  006132  112710  000244                    MOVB    #MIMR,(R0)      ;LOAD MODE BITS FOR IMR
 1184  006136  112710  000260                    MOVB    #PIMR,(R0)      ;PRESELECT IMR FOR WRITING
 1185  006142  113711  001124                    MOVB    $GDDAT,(R1)     ;WRITE INTO DATA PORT
 1186  006146  111115                            MOVB    (R1),(R5)       ;STORE DATA FOR COMPARE
 1187  006150  023715  001124                    CMP     $GDDAT,(R5)     ;CHECK IMR RESULTS
 1188  006154  001401                            BEQ     1$              ;
 1189  006156  104005                            ERROR   5               ;IMR ERROR
 1190  006160  112710  000254            1$:     MOVB    #MACR,(R0)      ;CHANGE TO ACR REGISTER
 1191  006164  005037  001124                    CLR     $GDDAT          ;STORE EXPECTED
 1192  006170  111115                            MOVB    (R1),(R5)       ;READ ACR
 1193  006172  023715  001124                    CMP     $GDDAT,(R5)     ;SHOULD STILL BE CLEARED
 1194  006176  001401                            BEQ     2$
 1195  006200  104004                            ERROR   4               ;ACR REG ERROR
 1196  006202  112710  000240            2$:     MOVB    #MISR,(R0)      ;LOAD MODE BITS FOR ISR
 1197  006206  111115                            MOVB    (R1),(R5)       ;READ ISR
 1198  006210  023715  001124                    CMP     $GDDAT,(R5)     ;ISR SHOULD BE CLEARED
 1199  006214  001401                            BEQ     3$
 1200  006216  104006                            ERROR   6               ;ISR REG ERROR
 1201  006220  112710  000250            3$:     MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
 1202  006224  111115                            MOVB    (R1),(R5)       ;READ IRR
 1203  006226  023715  001124                    CMP     $GDDAT,(R5)     ;IRR SHOULD BE CLEARED
 1204  006232  001401                            BEQ     4$
 1205  006234  104003                            ERROR   3               ;IRR REG ERROR
 1206  006236  105010                    4$:     CLRB    (R0)            ;CHIP RESET GROUP 1
 1207  006240  112710  000244                    MOVB    #MIMR,(R0)      ;LOAD MODE BITS FOR IMR
 1208  006244  012737  000377  001124            MOV     #377,$GDDAT     ;STORE EXPECTED
 1209  006252  111115                            MOVB    (R1),(R5)       ;READ IMR
 1210  006254  023715  001124                    CMP     $GDDAT,(R5)     ;IMR SHOULD BE ALL ONES
 1211  006260  001401                            BEQ     5$              ;DO NEXT GROUP
 1212  006262  104005                            ERROR   5               ;IMR REG ERROR
 1213  006264  005303                    5$:     DEC     R3              ;DO GROUPS 1 AND GROUP 2
 1214  006266  001405                            BEQ     TST25           ;;BR IF BOTH GROUPS TESTED
 1215  006270  013700  001352                    MOV     DRCSC,R0        ;SET UP TO TEST GROUP 2
 1216  006274  013701  001356                    MOV     DRCSD,R1        ;GROUP 2 DATA PORT
 1217  006300  000707                            BR      111$            ;DO GROUP 2 IMR UNIQUENESS
 1218
 1219                                    ;;*******************************************************************
 1220                                    ;*TEST 25        TEST GROUPS 1 AND 2 IRR UNIQUENESS
 1221                                    ;;*******************************************************************
 1222  006302  000004                    TST25:  SCOPE
 1223  006304  004737  013002                    JSR     PC,CLRCSR       ;CLEAR ALL CSRS
 1224  006310  013700  001342                    MOV     DRCSA,R0        ;CSRA ADDRESS
 1225  006314  013701  001346                    MOV     DRCSB,R1        ;CSRB ADDRESS
 1226  006320  012703  000002                    MOV     #2,R3           ;COUNTER FOR TESTING TWO GROUPS
 1227  006324  004737  013052                    JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
 1228  006330  010137  001122            111$:   MOV     R1,$BDADR       ;STORE ADDRESS
 1229  006334  012737  000200  001124            MOV     #200,$GDDAT     ;STORE EXPECTED
 1230  006342  112710  000250                    MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
 1231  006346  112710  000137                    MOVB    #137,(R0)       ;SET SINGLE IRR BIT7
 1232  006352  111115                            MOVB    (R1),(R5)       ;STORE DATA FOR COMPARE
 1233  006354  023715  001124                    CMP     $GDDAT,(R5)     ;CHECK IRR RESULTS
 1234  006360  001401                            BEQ     1$              ;
 1235  006362  104003                            ERROR   3               ;IRR ERROR
 1236  006364  112710  000244            1$:     MOVB    #MIMR,(R0)      ;CHANGE TO IMR REGISTER
 1237  006370  012737  000377  001124            MOV     #377,$GDDAT     ;STORE EXPECTED
 1238  006376  111115                            MOVB    (R1),(R5)       ;READ IMR
```

```
1239  006400  023715  001124              CMP     $GDDAT,(R5)     ;SHOULD STILL BE ALL 1'S
1240  006404  001401                      BEQ     2$
1241  006406  104005                      ERROR   5               ;IMR REG ERROR
1242  006410  112710  000240      2$:     MOVB    #MISR,(R0)      ;LOAD MODE BITS FOR ISR
1243  006414  005037  001124              CLR     $GDDAT          ;STORE EXPECTED
1244  006420  111115                      MOVB    (R1),(R5)       ;READ ISR
1245  006422  023715  001124              CMP     $GDDAT,(R5)     ;ISR SHOULD BE CLEARED
1246  006426  001401                      BEQ     3$
1247  006430  104006                      ERROR   6               ;ISR REG ERROR
1248  006432  112710  000254      3$:     MOVB    #MACR,(R0)      ;LOAD MODE BITS FOR ACR
1249  006436  111115                      MOVB    (R1),(R5)       ;READ ACR
1250  006440  023715  001124              CMP     $GDDAT,(R5)     ;ACR SHOULD BE CLEARED
1251  006444  001401                      BEQ     4$
1252  006446  104004                      ERROR   4               ;ACR REG ERROR
1253  006450  105010      4$:     CLRB    (R0)            ;CHIP RESET GROUP 1
1254  006452  112710  000250              MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
1255  006456  111115                      MOVB    (R1),(R5)       ;READ IRR
1256  006460  023715  001124              CMP     $GDDAT,(R5)     ;IRR SHOULD BE CLEARED
1257  006464  001401                      BEQ     5$
1258  006466  104003                      ERROR   3               ;IRR REG ERROR
1259  006470  005303      5$:     DEC     R3              ;TEST GROUPS 1 AND 2
1260  006472  001405                      BEQ     TST26           ;;BR IF BOTH GROUPS TESTED
1261  006474  013700  001352              MOV     DRCSC,R0        ;GROUP 2 CONTROL PORT
1262  006500  013701  001356              MOV     DRCSD,R1        ;GROUP 2 DATA PORT
1263  006504  000711                      BR      111$            ;TEST GROUP 2 ACR UNIQUENESS
1264
1265
1266                                  ;;**********************************************************
1267                                  ;*TEST 26       TEST GROUPS 1,2 ACR WITH PATTERNS
1268                                  ;;**********************************************************
1269  006506  000004              TST26:  SCOPE
1270  006510  004737  013002              JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1271  006514  012702  000002              MOV     #2,R2           ;TEST TWO GROUPS
1272  006520  012737  006546  001110      MOV     #1$,$LPERR      ;SET UP LOOP RETURN
1273  006526  013700  001342              MOV     DRCSA,R0        ;STORE CSRA ADDRESS
1274  006532  013701  001346              MOV     DRCSB,R1        ;STORE CSRB ADDRESS
1275  006536  012703  013110      111$:   MOV     #BEGPAT,R3      ;SET UP PATTERN TABLE
1276  006542  010137  001122              MOV     R1,$BDADR       ;STORE ADDRESS
1277  006546  112710  000254      1$:     MOVB    #MACR,(R0)      ;LOAD MODE BITS FOR ACR
1278  006552  112710  000300              MOVB    #PACR,(R0)      ;PRESELECT ACR FOR WRITING
1279  006556  111337  001124              MOVB    (R3),$GDDAT     ;STORE EXPECTED
1280  006562  111311                      MOVB    (R3),(R1)       ;WRITE INTO ACR
1281  006564  111115                      MOVB    (R1),(R5)       ;READ OUT OF ACR
1282  006566  023715  001124              CMP     $GDDAT,(R5)
1283  006572  001401                      BEQ     2$
1284  006574  104004                      ERROR   4               ;ACR REGISTER ERROR
1285  006576  005723      2$:     TST     (R3)+           ;INC FOR NEXT PATTERN
1286  006600  020327  013410              CMP     R3,#ENDDAT      ;CHECK FOR TABLE END
1287  006604  001360                      BNE     1$              ;W/R NEXT PATTERN
1288  006606  005302                      DEC     R2              ;FINISHED BOTH GROUPS?
1289  006610  001405                      BEQ     TST27           ;;BR IF EQUAL
1290  006612  013700  001352              MOV     DRCSC,R0
1291  006616  013701  001356              MOV     DRCSD,R1
1292  006622  000745                      BR      111$            ;DO NEXT GROUP
1293
1294                                  ;;**********************************************************
```

```
1295                                    ;*TEST 27       TEST GROUPS 1,2 IMR WITH PATTERNS
1296                                    ;;**************************************************************
1297  006624  000004                    TST27:  SCOPE
1298  006626  004737  013002                    JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1299  006632  012702  000002                    MOV     #2,R2           ;TEST TWO GROUPS
1300  006636  012737  006664  001110          MOV     #1$,$LPERR      ;SET UP LOOP RETURN
1301  006644  013700  001342                    MOV     DRCSA,R0        ;STORE CSRA ADDRESS
1302  006650  013701  001346                    MOV     DRCSB,R1        ;STORE CSRB ADDRESS
1303  006654  012703  013110          111$:   MOV     #BEGPAT,R3      ;SET UP PATTERN TABLE
1304  006660  010137  001122                    MOV     R1,$BDADR       ;STORE ADDRESS
1305  006664  112710  000244          1$:     MOVB    #MIMR,(R0)      ;LOAD MODE BITS FOR IMR
1306  006670  112710  000260                    MOVB    #PIMR,(R0)      ;PRESELECT IMR FOR WRITING
1307  006674  111337  001124                    MOVB    (R3),$GDDAT     ;STORE EXPECTED
1308  006700  111311                          MOVB    (R3),(R1)       ;WRITE INTO IMR
1309  006702  111115                          MOVB    (R1),(R5)       ;READ OUT OF IMR
1310  006704  023715  001124                    CMP     $GDDAT,(R5)
1311  006710  001401                          BEQ     2$
1312  006712  104005                          ERROR   5               ;IMR REGISTER ERROR
1313  006714  005723                  2$:     TST     (R3)+           ;INC FOR NEXT PATTERN
1314  006716  020327  013410                    CMP     R3,#ENDDAT      ;CHECK FOR TABLE END
1315  006722  001360                          BNE     1$              ;W/R NEXT PATTERN
1316  006724  005302                          DEC     R2              ;FINISHED BOTH GROUPS?
1317  006726  001405                          BEQ     TST30           ;;BR IF EQUAL
1318  006730  013700  001352                    MOV     DRCSC,R0
1319  006734  013701  001356                    MOV     DRCSD,R1
1320  006740  000745                          BR      111$            ;DO NEXT GROUP
1321
1322                                    ;;**************************************************************
1323                                    ;*TEST 30       TEST GROUP 1,2 CLEAR IMR INSTR.
1324                                    ;;**************************************************************
1325  006742  000004                    TST30:  SCOPE
1326  006744  004737  013002                    JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1327  006750  012704  000002                    MOV     #2,R4           ;COUNTER FOR TWO GROUPS
1328  006754  013700  001342                    MOV     DRCSA,R0        ;CSRA ADDRESS
1329  006760  013701  001346                    MOV     DRCSB,R1        ;CSRB ADDRESS
1330  006764  010137  001122          111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1331  006770  005037  001124                    CLR     $GDDAT          ;EXPECTED DATA
1332  006774  112710  000244                    MOVB    #MIMR,(R0)      ;LOAD MODE BITS TO READ IMR
1333  007000  112710  000040                    MOVB    #CIMR,(R0)      ;CLEAR IMR COMMAND
1334  007004  111115                          MOVB    (R1),(R5)       ;READ DATA PORT
1335  007006  023715  001124                    CMP     $GDDAT,(R5)
1336  007012  001401                          BEQ     1$
1337  007014  104005                          ERROR   5               ;ERROR ,IMR SHOULD BE CLEARED
1338  007016  005304                  1$:     DEC     R4              ;FINISHED BOTH GROUPS?
1339  007020  001405                          BEQ     TST31           ;;BR IF BOTH GROUPS TESTED
1340  007022  013700  001352                    MOV     DRCSC,R0        ;SETUP FOR GROUP 2
1341  007026  013701  001356                    MOV     DRCSD,R1
1342  007032  000754                          BR      111$            ;DO IMR TEST WITH GROUP 2
1343
1344                                    ;;**************************************************************
1345                                    ;*TEST 31       TEST GROUP 1,2 SET IMR INSTR.
1346                                    ;;**************************************************************
1347  007034  000004                    TST31:  SCOPE
1348  007036  004737  013002                    JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1349  007042  012704  000002                    MOV     #2,R4           ;COUNTER FOR TWO GROUPS
1350  007046  013700  001342                    MOV     DRCSA,R0        ;CSRA ADDRESS
```

```
1351  007052  013701  001346                  MOV     DRCSB,R1        ;CSRB ADDRESS
1352  007056  010137  001122          111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1353  007062  012737  000377  001124          MOV     #377,$GDDAT     ;EXPECTED DATA
1354  007070  112710  000244                  MOVB    #MIMR,(R0)      ;LOAD MODE BITS TO READ IMR
1355  007074  112710  000040                  MOVB    #CIMR,(R0)      ;CLEAR IMR
1356  007100  112710  000060                  MOVB    #SIMR,(R0)      ;SET IMR COMMAND
1357  007104  111115                          MOVB    (R1),(R5)       ;READ DATA PORT
1358  007106  023715  001124                  CMP     $GDDAT,(R5)
1359  007112  001401                          BEQ     1$
1360  007114  104005                          ERROR   5               ;ERROR ,IMR SHOULD BE SET
1361  007116  005304                  1$:     DEC     R4              ;FINISHED BOTH GROUPS?
1362  007120  001405                          BEQ     TST32           ;;BR IF BOTH GROUPS TESTED
1363  007122  013700  001352                  MOV     DRCSC,R0        ;SETUP FOR GROUP 2
1364  007126  013701  001356                  MOV     DRCSD,R1
1365  007132  000751                          BR      111$            ;DO IMR TEST WITH GROUP 2
1366
1367                                  ;;*******************************************************************
1368                                  ;*TEST 32        TEST GROUP 1,2  CLEAR SINGLE IMR BIT INSTR.
1369                                  ;;*******************************************************************
1370  007134  000004                  TST32:  SCOPE
1371  007136  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1372  007142  012704  000002                  MOV     #2,R4           ;GROUP COUNTER
1373  007146  012737  007204  001110          MOV     #1$,$LPERR      ;SCOPE RETURN ADDRESS
1374  007154  013700  001342                  MOV     DRCSA,R0        ;CSRA ADDRESS
1375  007160  013701  001346                  MOV     DRCSB,R1        ;CSRB ADDRESS
1376  007164  010137  001122          111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1377  007170  012703  013254                  MOV     #BGCHP4,R3      ;GOOD DATA PATTERN TABLE
1378  007174  112710  000244                  MOVB    #MIMR,(R0)      ;LOAD MODE BITS FOR IMR
1379  007200  012702  000050                  MOV     #CSIMR,R2       ;CLEAR SINGLE IMR BIT VALUE
1380  007204  111337  001124          1$:     MOVB    (R3),$GDDAT     ;STORE EXPECTED
1381  007210  112710  000060                  MOVB    #SIMR,(R0)      ;SET ALL IMR BITS
1382  007214  110210                          MOVB    R2,(R0)         ;CLEAR SINGLE IMR BIT
1383  007216  111115                          MOVB    (R1),(R5)       ;READ DATA PORT
1384  007220  023715  001124                  CMP     $GDDAT,(R5)
1385  007224  001401                          BEQ     2$
1386  007226  104005                          ERROR   5               ;IMR REG ERROR
1387  007230  005723                  2$:     TST     (R3)+           ;INC EXPECTED DATA TABLE
1388  007232  020327  013274                  CMP     R3,#EDCHP4      ;CHECK FOR END
1389  007236  001402                          BEQ     3$
1390  007240  005202                          INC     R2              ;SET UP TO CLEAR NEXT IMR BIT
1391  007242  000760                          BR      1$              ;CLEAR NEXT IMR BIT
1392  007244  005304                  3$:     DEC     R4              ;DO GROUPS 1 AND 2
1393  007246  001405                          BEQ     TST33           ;;BR IF BOTH GROUPS TESTED
1394  007250  013700  001352                  MOV     DRCSC,R0        ;CSRC ADDRESS
1395  007254  013701  001356                  MOV     DRCSD,R1        ;CSRD ADDRESS
1396  007260  000741                          BR      111$            ;DO GROUP 2
1397
1398
1399                                  ;;*******************************************************************
1400                                  ;*TEST 33        TEST GROUP 1,2  SET SINGLE IMR BIT INSTR.
1401                                  ;;*******************************************************************
1402  007262  000004                  TST33:  SCOPE
1403  007264  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1404  007270  012704  000002                  MOV     #2,R4           ;GROUP COUNTER
1405  007274  012737  007332  001110          MOV     #1$,$LPERR      ;SCOPE RETURN ADDRESS
1406  007302  013700  001342                  MOV     DRCSA,R0        ;CSRA ADDRESS
```

```
1407  007306  013701  001346                MOV     DRCSB,R1        ;CSRB ADDRESS
1408  007312  010137  001122        111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1409  007316  012703  013212                MOV     #BGCHP3,R3      ;GOOD DATA PATTERN TABLE
1410  007322  112710  000244                MOVB    #MIMR,(R0)      ;LOAD MODE BITS FOR IMR
1411  007326  012702  000070                MOV     #SSIMR,R2       ;SET SINGLE IMR BIT VALUE
1412  007332  111337  001124        1$:     MOVB    (R3),$GDDAT     ;STORE EXPECTED
1413  007336  112710  000040                MOVB    #CIMR,(R0)      ;CLEAR ALL IMR BITS
1414  007342  110210                        MOVB    R2,(R0)         ;SET SINGLE IMR BIT
1415  007344  111115                        MOVB    (R1),(R5)       ;READ DATA PORT
1416  007346  023715  001124                CMP     $GDDAT,(R5)
1417  007352  001401                        BEQ     2$
1418  007354  104005                        ERROR   5               ;IMR REG ERROR
1419  007356  005723        2$:     TST     (R3)+           ;INC EXPECTED DATA TABLE
1420  007360  020327  013232                CMP     R3,#EDCHP3      ;CHECK FOR END
1421  007364  001402                        BEQ     3$
1422  007366  005202                        INC     R2              ;SET UP TO SET NEXT IMR BIT
1423  007370  000760                        BR      1$              ;SET NEXT IMR BIT
1424  007372  005304        3$:     DEC     R4              ;DO GROUPS 1 AND 2
1425  007374  001405                        BEQ     TST34           ;;BR IF BOTH GROUPS TESTED
1426  007376  013700  001352                MOV     DRCSC,R0        ;CSRC ADDRESS
1427  007402  013701  001356                MOV     DRCSD,R1        ;CSRD ADDRESS
1428  007406  000741                        BR      111$            ;DO GROUP 2
1429
1430                          ;;*************************************************************
1431                          ;*TEST 34       TEST GROUP 1,2 SET IRR INSTR.
1432                          ;;*************************************************************
1433  007410  000004        TST34:  SCOPE
1434  007412  004737  013002                JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1435  007416  012704  000002                MOV     #2,R4           ;COUNTER FOR TWO GROUPS
1436  007422  013700  001342                MOV     DRCSA,R0        ;CSRA ADDRESS
1437  007426  013701  001346                MOV     DRCSB,R1        ;CSRB ADDRESS
1438  007432  004737  013052                JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1439  007436  010137  001122        111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1440  007442  012737  000377  001124        MOV     #377,$GDDAT     ;EXPECTED DATA
1441  007450  112710  000250                MOVB    #MIRR,(R0)      ;LOAD MODE BITS TO READ IRR
1442  007454  112710  000120                MOVB    #SIRR,(R0)      ;SET IRR COMMAND
1443  007460  111115                        MOVB    (R1),(R5)       ;READ DATA PORT
1444  007462  023715  001124                CMP     $GDDAT,(R5)
1445  007466  001401                        BEQ     1$
1446  007470  104003                        ERROR   3               ;ERROR ,IRR SHOULD BE SET
1447  007472  005304        1$:     DEC     R4              ;FINISHED BOTH GROUPS?
1448  007474  001405                        BEQ     TST35           ;;BR IF BOTH GROUPS TESTED
1449  007476  013700  001352                MOV     DRCSC,R0        ;SETUP FOR GROUP 2
1450  007502  013701  001356                MOV     DRCSD,R1
1451  007506  000753                        BR      111$            ;DO IRR TEST WITH GROUP 2
1452
1453
1454                          ;;*************************************************************
1455                          ;*TEST 35       TEST GROUP 1,2 CLEAR IRR INSTR.
1456                          ;;*************************************************************
1457  007510  000004        TST35:  SCOPE
1458  007512  004737  013002                JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1459  007516  012704  000002                MOV     #2,R4           ;COUNTER FOR TWO GROUPS
1460  007522  013700  001342                MOV     DRCSA,R0        ;CSRA ADDRESS
1461  007526  013701  001346                MOV     DRCSB,R1        ;CSRB ADDRESS
1462  007532  004737  013052                JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
```

```
1463  007536  010137  001122          111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1464  007542  005037  001124                  CLR     $GDDAT          ;EXPECTED DATA
1465  007546  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD MODE BITS TO READ IRR
1466  007552  112710  000120                  MOVB    #SIRR,(R0)      ;SET IRR BITS
1467  007556  112710  000100                  MOVB    #CIRR,(R0)      ;CLEAR IRR COMMAND
1468  007562  111115                          MOVB    (R1),(R5)       ;READ DATA PORT
1469  007564  023715  001124                  CMP     $GDDAT,(R5)
1470  007570  001401                          BEQ     1$
1471  007572  104003                          ERROR   3               ;ERROR ,IRR SHOULD BE CLEARED
1472  007574  005304                  1$:     DEC     R4              ;FINISHED BOTH GROUPS?
1473  007576  001405                          BEQ     TST36           ;;BR IF BOTH GROUPS TESTED
1474  007600  013700  001352                  MOV     DRCSC,R0        ;SETUP FOR GROUP 2
1475  007604  013701  001356                  MOV     DRCSD,R1
1476  007610  000752                          BR      111$            ;DO IRR TEST WITH GROUP 2
1477
1478                                  ;;**********************************************************************
1479                                  ;*TEST 36        TEST GROUP 1,2  CLEAR SINGLE IRR BIT INSTR.
1480                                  ;;**********************************************************************
1481  007612  000004                  TST36:  SCOPE
1482  007614  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1483  007620  012704  000002                  MOV     #2,R4           ;GROUP COUNTER
1484  007624  012737  007666  001110          MOV     #1$,$LPERR      ;SCOPE RETURN ADDRESS
1485  007632  013700  001342                  MOV     DRCSA,R0        ;CSRA ADDRESS
1486  007636  013701  001346                  MOV     DRCSB,R1        ;CSRB ADDRESS
1487  007642  004737  013052                  JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1488  007646  010137  001122          111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1489  007652  012703  013254                  MOV     #BGCHP4,R3      ;GOOD DATA PATTERN TABLE
1490  007656  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
1491  007662  012702  000110                  MOV     #CSIRR,R2       ;CLEAR SINGLE IRR BIT VALUE
1492  007666  111337  001124          1$:     MOVB    (R3),$GDDAT     ;STORE EXPECTED
1493  007672  112710  000120                  MOVB    #SIRR,(R0)      ;SET ALL IRR BITS
1494  007676  110210                          MOVB    R2,(R0)         ;CLEAR SINGLE IRR BIT
1495  007700  111115                          MOVB    (R1),(R5)       ;READ DATA PORT
1496  007702  023715  001124                  CMP     $GDDAT,(R5)
1497  007706  001401                          BEQ     2$
1498  007710  104003                          ERROR   3               ;IRR REG ERROR
1499  007712  005723                  2$:     TST     (R3)+           ;INC EXPECTED DATA TABLE
1500  007714  020327  013274                  CMP     R3,#EDCHP4      ;CHECK FOR END
1501  007720  001402                          BEQ     3$
1502  007722  005202                          INC     R2              ;SET UP TO CLEAR NEXT IRR BIT
1503  007724  000760                          BR      1$              ;CLEAR NEXT IRR BIT
1504  007726  005304                  3$:     DEC     R4              ;DO GROUPS 1 AND 2
1505  007730  001405                          BEQ     TST37           ;;BR IF BOTH GROUPS TESTED
1506  007732  013700  001352                  MOV     DRCSC,R0        ;CSRC ADDRESS
1507  007736  013701  001356                  MOV     DRCSD,R1        ;CSRD ADDRESS
1508  007742  000741                          BR      111$            ;DO GROUP 2
1509
1510                                  ;;**********************************************************************
1511                                  ;*TEST 37        TEST GROUP 1,2  SET SINGLE IRR BIT INSTR.
1512                                  ;;**********************************************************************
1513  007744  000004                  TST37:  SCOPE
1514  007746  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1515  007752  012704  000002                  MOV     #2,R4           ;GROUP COUNTER
1516  007756  012737  010020  001110          MOV     #1$,$LPERR      ;SCOPE RETURN ADDRESS
1517  007764  013700  001342                  MOV     DRCSA,R0        ;CSRA ADDRESS
1518  007770  013701  001346                  MOV     DRCSB,R1        ;CSRB ADDRESS
```

```
1519  007774  004737  013052          JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1520  010000  010137  001122  111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1521  010004  012703  013212          MOV     #BGCHP3,R3      ;GOOD DATA PATTERN TABLE
1522  010010  112710  000250          MOVB    #MIRR,(R0)      ;LOAD MODE BITS FOR IRR
1523  010014  012702  000130          MOV     #SSIRR,R2       ;SET SINGLE IRR BIT VALUE
1524  010020  111337  001124  1$:     MOVB    (R3),$GDDAT     ;STORE EXPECTED
1525  010024  112710  000100          MOVB    #CIRR,(R0)      ;CLEAR ALL IRR BITS
1526  010030  110210                  MOVB    R2,(R0)         ;SET SINGLE IRR BIT
1527  010032  111115                  MOVB    (R1),(R5)       ;READ DATA PORT
1528  010034  023715  001124          CMP     $GDDAT,(R5)
1529  010040  001401                  BEQ     2$
1530  010042  104003                  ERROR   3               ;IRR REG ERROR
1531  010044  005723          2$:     TST     (R3)+           ;INC EXPECTED DATA TABLE
1532  010046  020327  013232          CMP     R3,#EDCHP3      ;CHECK FOR END
1533  010052  001402                  BEQ     3$
1534  010054  005202                  INC     R2              ;SET UP TO SET NEXT IRR BIT
1535  010056  000760                  BR      1$              ;SET NEXT IRR BIT
1536  010060  005304          3$:     DEC     R4              ;DO GROUPS 1 AND 2
1537  010062  001405                  BEQ     TST40           ;;BR IF BOTH GROUPS TESTED
1538  010064  013700  001352          MOV     DRCSC,R0        ;CSRC ADDRESS
1539  010070  013701  001356          MOV     DRCSD,R1        ;CSRD ADDRESS
1540  010074  000741                  BR      111$            ;DO GROUP 2
1541
1542                                  ;;*****************************************************************
1543                                  ;*TEST 40       TEST GROUP 1,2 CLEAR IRR+IMR INSTR.
1544                                  ;;*****************************************************************
1545  010076  000004          TST40:  SCOPE
1546  010100  004737  013002          JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1547  010104  012704  000002          MOV     #2,R4           ;COUNTER FOR TWO GROUPS
1548  010110  013700  001342          MOV     DRCSA,R0        ;CSRA ADDRESS
1549  010114  013701  001346          MOV     DRCSB,R1        ;CSRB ADDRESS
1550  010120  004737  013052          JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1551  010124  010137  001122  111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1552  010130  005037  001124          CLR     $GDDAT          ;EXPECTED DATA
1553  010134  112710  000250          MOVB    #MIRR,(R0)      ;LOAD MODE BITS TO READ IRR
1554  010140  112710  000120          MOVB    #SIRR,(R0)      ;SET IRR BITS
1555  010144  112710  000060          MOVB    #SIMR,(R0)      ;SET IMR BITS
1556  010150  112710  000020          MOVB    #CIRMR,(R0)     ;CLEAR IRR+IMR COMMAND
1557  010154  111115                  MOVB    (R1),(R5)       ;READ DATA PORT FOR IRR
1558  010156  023715  001124          CMP     $GDDAT,(R5)
1559  010162  001401                  BEQ     1$
1560  010164  104003                  ERROR   3               ;ERROR ,IRR SHOULD BE CLEARED
1561  010166  112710  000244  1$:     MOVB    #MIMR,(R0)      ;LOAD MODE BITS TO READ IMR
1562  010172  111115                  MOVB    (R1),(R5)       ;READ IMR REG
1563  010174  023715  001124          CMP     $GDDAT,(R5)
1564  010200  001401                  BEQ     2$
1565  010202  104005                  ERROR   5               ;IRR+IMR COMMAND DID NOT CLEAR IMR
1566  010204  005304          2$:     DEC     R4              ;FINISHED BOTH GROUPS?
1567  010206  001405                  BEQ     TST41           ;;BR IF BOTH GROUPS TESTED
1568  010210  013700  001352          MOV     DRCSC,R0        ;SETUP FOR GROUP 2
1569  010214  013701  001356          MOV     DRCSD,R1
1570  010220  000741                  BR      111$            ;DO IRR+IMR TEST WITH GROUP 2
1571
1572                                  ;;*****************************************************************
1573                                  ;*TEST 41       TEST GROUP 1,2  CLEAR SINGLE IRR+IMR BIT INSTR.
1574                                  ;;*****************************************************************
```

```
1575  010222  000004                  TST41:  SCOPE
1576  010224  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1577  010230  012704  000002                  MOV     #2,R4           ;GROUP COUNTER
1578  010234  012737  010272  001110          MOV     #1$,$LPERR      ;SCOPE RETURN ADDRESS
1579  010242  013700  001342                  MOV     DRCSA,R0        ;CSRA ADDRESS
1580  010246  013701  001346                  MOV     DRCSB,R1        ;CSRB ADDRESS
1581  010252  004737  013052                  JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1582  010256  010137  001122          111$:   MOV     R1,$BDADR       ;STORE ADDRESS
1583  010262  012703  013254                  MOV     #BGCHP4,R3      ;GOOD DATA PATTERN TABLE
1584  010266  012702  000030                  MOV     #CSIRMR,R2      ;CLEAR SINGLE IRR+IMR BIT VALUE
1585  010272  111337  001124          1$:     MOVB    (R3),$GDDAT     ;STORE EXPECTED
1586  010276  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD MODE BITS TO READ IRR
1587  010302  112710  000120                  MOVB    #SIRR,(R0)      ;SET ALL IRR BITS
1588  010306  112710  000060                  MOVB    #SIMR,(R0)      ;SET ALL IMR BITS
1589  010312  110210                          MOVB    R2,(R0)         ;CLEAR SINGLE IRR+IMR BIT
1590  010314  111115                          MOVB    (R1),(R5)       ;READ DATA PORT FOR IRR
1591  010316  023715  001124                  CMP     $GDDAT,(R5)
1592  010322  001401                          BEQ     2$
1593  010324  104003                          ERROR   3               ;IRR REG ERROR
1594  010326  112710  000244          2$:     MOVB    #MIMR,(R0)      ;SET UP TO READ IMR
1595  010332  111115                          MOVB    (R1),(R5)       ;READ IMR REGISTER
1596  010334  023715  001124                  CMP     $GDDAT,(R5)
1597  010340  001401                          BEQ     3$
1598  010342  104005                          ERROR   5               ;IMR REG ERROR
1599  010344  005723                  3$:     TST     (R3)+           ;INC EXPECTED DATA TABLE
1600  010346  020327  013274                  CMP     R3,#EDCHP4      ;CHECK FOR END
1601  010352  001402                          BEQ     4$
1602  010354  005202                          INC     R2              ;SET UP TO CLEAR NEXT IRR+IMR BIT
1603  010356  000745                          BR      1$              ;CLEAR NEXT IRR+IMR BIT
1604  010360  005304                  4$:     DEC     R4              ;DO GROUPS 1 AND 2
1605  010362  001405                          BEQ     TST42           ;;BR IF BOTH GROUPS TESTED
1606  010364  013700  001352                  MOV     DRCSC,R0        ;CSRC ADDRESS
1607  010370  013701  001356                  MOV     DRCSD,R1        ;CSRD ADDRESS
1608  010374  000730                          BR      111$            ;DO GROUP 2
1609
1610                                  ;;*******************************************************************
1611                                  ;*TEST 42        TEST GROUPS 1,2 FOR GROUP UNIQUENESS
1612                                  ;;*******************************************************************
1613  010376  000004                  TST42:  SCOPE
1614  010400  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1615                                                                   ;CSRA = R0
1616                                                                   ;CSRB = R1
1617                                                                   ;CSRC = R2
1618                                                                   ;CSRD = R3
1619  010404  012704  000002                  MOV     #2,R4           ;COUNTER FOR TESTING TWO GROUPS
1620  010410  004737  013052                  JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1621  010414  010337  001122          111$:   MOV     R3,$BDADR       ;STORE ADDRESS
1622  010420  112710  000300                  MOVB    #PACR,(R0)      ;PRESELECT ACR FOR WRITING
1623  010424  112711  000377                  MOVB    #377,(R1)       ;WRITE INTO DATA PORT FOR ACR
1624  010430  112710  000120                  MOVB    #SIRR,(R0)      ;SET IRR TO ALL 1'S
1625  010434  112710  000040                  MOVB    #CIMR,(R0)      ;CLEAR IMR
1626  010440  112712  000254                  MOVB    #MACR,(R2)      ;LOAD MODE TO READ ACR
1627  010444  005037  001124                  CLR     $GDDAT          ;EXPECTED
1628  010450  111315                          MOVB    (R3),(R5)       ;STORE DATA FOR COMPARE
1629  010452  023715  001124                  CMP     $GDDAT,(R5)     ;CHECK ACR RESULTS
1630  010456  001401                          BEQ     1$              ;
```

```
1631  010460  104004                          ERROR   4               ;ERROR,OTHER GROUP ACR SHOULD BE CLEARED
1632  010462  112712  000244          1$:     MOVB    #MIMR,(R2)      ;CHANGE TO IMR REGISTER
1633  010466  012737  000377  001124          MOV     #377,$GDDAT     ;STORE EXPECTED
1634  010474  111315                          MOVB    (R3),(R5)       ;READ IMR
1635  010476  023715  001124                  CMP     $GDDAT,(R5)     ;SHOULD  BE ALL 1'S
1636  010502  001401                          BEQ     2$
1637  010504  104005                          ERROR   5               ;ERROR,OTHER GROUP IMR SHOULD BE SET
1638  010506  112712  000240          2$:     MOVB    #MISR,(R2)      ;LOAD MODE BITS FOR ISR
1639  010512  005037  001124                  CLR     $GDDAT          ;STORE EXPECTED
1640  010516  111315                          MOVB    (R3),(R5)       ;READ ISR
1641  010520  023715  001124                  CMP     $GDDAT,(R5)     ;ISR SHOULD BE CLEARED
1642  010524  001401                          BEQ     3$
1643  010526  104006                          ERROR   6               ;ISR REG ERROR
1644  010530  112712  000250          3$:     MOVB    #MIRR,(R2)      ;LOAD MODE BITS FOR IRR
1645  010534  111315                          MOVB    (R3),(R5)       ;READ IRR
1646  010536  023715  001124                  CMP     $GDDAT,(R5)     ;IRR SHOULD BE CLEARED
1647  010542  001401                          BEQ     4$
1648  010544  104003                          ERROR   3               ;ERROR,OTHER GROUP IRR SHOULD BE CLEARED
1649  010546  005304                  4$:     DEC     R4              ;TEST GROUPS 1 AND 2
1650  010550  001415                          BEQ     TST43           ;;BR IF BOTH GROUPS TESTED
1651  010552  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1652  010556  013700  001352                  MOV     DRCSC,R0        ;GROUP 2 CONTROL PORT
1653  010562  013701  001356                  MOV     DRCSD,R1        ;GROUP 2 DATA PORT
1654  010566  013702  001342                  MOV     DRCSA,R2        ;GROUP 1 CONTROL PORT
1655  010572  013703  001346                  MOV     DRCSB,R3        ;GROUP 1 DATA PORT
1656  010576  004737  013052                  JSR     PC,CLRIRR       ;CLEAR IRR REGS WITH CHIP RESET
1657  010602  000704                          BR      111$            ;TEST GROUP 2 ACR UNIQUENESS
1658
1659                          ;;*****************************************************************
1660                          ;*TEST 43       TEST STATUS BITS GINT,S2,S1,S0,GP1,2
1661                          ;;*****************************************************************
1662  010604  000004          TST43:  SCOPE
1663  010606  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1664  010612  013700  001342                  MOV     DRCSA,R0
1665  010616  013701  001346                  MOV     DRCSB,R1
1666  010622  012703  000002                  MOV     #2,R3           ;DO TWO GROUPS
1667  010626  012704  000120          111$:   MOV     #120,R4         ;EXPECTED STATUS BITS
1668  010632  005077  170504                  CLR     @DRCSA          ;INIT CSRA
1669  010636  005077  170510                  CLR     @DRCSC          ;INIT CSRC
1670  010642  012702  013112                  MOV     #BGPAT1,R2      ;EXPECTED IRR PATTERN
1671  010646  112760  000001  000001          MOVB    #BIT0,1(R0)     ;CSR TO OUTPUT MODE
1672  010654  112777  000204  170460          MOVB    #204,@DRCSA     ;POLLED MODE FOR CSRA,GROUP 1
1673  010662  112777  000204  170462          MOVB    #204,@DRCSC     ;POLLED MODE FOR CSRC,GROUP 2
1674  010670  112710  000020                  MOVB    #CIRMR,(R0)     ;CLEAR IMR + IRR
1675  010674  012737  000070  001372          MOV     #SSIMR,IMRLOC   ;STORE CODE FOR SINGLE IMR
1676  010702  012737  000130  001376          MOV     #SSIRR,IRRLOC   ;STORE CODE FOR SINGLE IRR
1677  010710  010037  001122          112$:   MOV     R0,$BDADR       ;CSR CHIP COMMAND  ADDRESS
1678  010714  010437  001124                  MOV     R4,$GDDAT       ;EXPECTED DATA
1679  010720  113710  001376                  MOVB    IRRLOC,(R0)     ;SET SINGLE IRR BIT
1680  010724  111015                          MOVB    (R0),(R5)       ;CHECK STATUS
1681  010726  023715  001124                  CMP     $GDDAT,(R5)
1682  010732  001401                          BEQ     1$
1683  010734  104007                          ERROR   7               ;CHIP STATUS ERROR
1684  010736  005037  001124          1$:     CLR     $GDDAT
1685  010742  010137  001122                  MOV     R1,$BDADR       ;CSR CHIP DATA ADDRESS
1686  010746  111237  001124                  MOVB    (R2),$GDDAT
```

```
1687  010752  112710  000250          MOVB    #MIRR,(R0)      ;READ IRR
1688  010756  111115                  MOVB    (R1),(R5)
1689  010760  023715  001124          CMP     $GDDAT,(R5)     ;CHECK IRR
1690  010764  001401                  BEQ     2$
1691  010766  104003                  ERROR   3               ;IRR ERROR
1692  010770  010037  001122  2$:     MOV     R0,$BDADR
1693  010774  113710  001372          MOVB    IMRLOC,(R0)     ;SET IMR BIT
1694  011000  012737  000320  001124  MOV     #320,$GDDAT     ;CSR EXPECTED DATA
1695  011006  111015                  MOVB    (R0),(R5)
1696  011010  042715  000007          BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
1697  011014  023715  001124          CMP     $GDDAT,(R5)
1698  011020  001401                  BEQ     3$
1699  011022  104007                  ERROR   7               ;CHIP STATUS ERROR
1700  011024  010137  001122  3$:     MOV     R1,$BDADR
1701  011030  011237  001124          MOV     (R2),$GDDAT     ;EXPECTED DATA
1702  011034  112710  000244          MOVB    #MIMR,(R0)      ;READ IMR BITS
1703  011040  111115                  MOVB    (R1),(R5)       ;SAVE IMR READ
1704  011042  023715  001124          CMP     $GDDAT,(R5)
1705  011046  001401                  BEQ     4$
1706  011050  104005                  ERROR   5               ;IMR ERROR
1707  011052  005722          4$:     TST     (R2)+           ;NEXT EXPECTED FOR IMR + IRR
1708  011054  020227  013132          CMP     R2,#EDCHP1      ;CHECK FOR END
1709  011060  001407                  BEQ     5$
1710  011062  005237  001376          INC     IRRLOC          ;NEXT IRR BIT
1711  011066  005237  001372          INC     IMRLOC          ;NEXT IMR BIT
1712  011072  005204                  INC     R4              ;INDEX EXPECTED STATUS
1713  011074  000137  010710          JMP     112$            ;DO NEXT STATUS CHECK
1714  011100  005303          5$:     DEC     R3              ;FINISHED BOTH GROUPS?
1715  011102  001406                  BEQ     TST44           ;;BR IF EQUAL
1716  011104  013700  001352          MOV     DRCSC,R0
1717  011110  013701  001356          MOV     DRCSD,R1
1718  011114  000137  010626          JMP     111$            ;DO NEXT GROUP
1719
1720
1721                                  ;;*********************************************************************
1722                                  ;*TEST 44        TEST POLLED MODE;CSRS A,B=OUT C,D=IN,ACTIVE LOW
1723                                  ;;*********************************************************************
1724  011120  000004          TST44:  SCOPE
1725  011122  004737  013002          JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1726                                                          ;R0 = CSRA-GROUP 1 CONTROL
1727                                                          ;R1 = CSRB-GROUP 1 DATA
1728                                                          ;R2 = CSRC-GROUP 2 CONTROL
1729                                                          ;R3 = CSRD-GROUP 2 DATA
1730  011126  012737  011170  001110  MOV     #1$,$LPERR      ;SET FOR SCOPE RETURN
1731  011134  012704  013110          MOV     #BEGPAT,R4      ;START OF PATTERN TABLE
1732  011140  112760  000001  000001  MOVB    #BIT0,1(R0)     ;SET CSRA TO OUTPUT MODE
1733  011146  112761  000001  000001  MOVB    #BIT0,1(R1)     ;SET CSRB TO OUTPUT MODE
1734
1735  011154  105010                  CLRB    (R0)            ;CHIP RESET GROUP 1 CSRA
1736  011156  105012                  CLRB    (R2)            ;CHIP RESET GROUP 2 CSRC
1737  011160  112710  000204          MOVB    #204,(R0)       ;LOAD MODE BITS FOR POLLED MODE,GR 1
1738  011164  112712  000204          MOVB    #204,(R2)       ;LOAD MODE BITS FOR POLLED MODE,GR2
1739  011170  011460  000002  1$:     MOV     (R4),2(R0)      ;SET PATTERN DBRA FROM H TO L
1740  011174  112710  000020          MOVB    #CIRMR,(R0)     ;CLEAR IMR+IRR GROUP 1
1741  011200  112712  000020          MOVB    #CIRMR,(R2)     ;CLEAR IMR+IRR GROUP 2
1742  011204  005060  000002          CLR     2(R0)           ;CLEAR DBRA,ACTIVE LOW WILL SET RPLY C
```

```
1743  011210  010137  001122          2$:     MOV     R1,$BDADR       ; GROUP 1 DATA PORT
1744  011214  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD BITS TO READ IRR
1745  011220  111437  001124                  MOVB    (R4),$GDDAT
1746  011224  111115                          MOVB    (R1),(R5)       ;READ IRR,GROUP 1
1747  011226  023715  001124                  CMP     $GDDAT,(R5)
1748  011232  001401                          BEQ     3$
1749  011234  104003                          ERROR   3               ;IRR ERROR,GROUP 1
1750  011236  010337  001122          3$:     MOV     R3,$BDADR       ;GROUP 2 DATA PORT
1751  011242  112712  000250                  MOVB    #MIRR,(R2)      ;LOAD MODE BITS TO READ IRR GROUP2
1752  011246  116437  000001  001124          MOVB    1(R4),$GDDAT    ;BUILD EXPECTED DATA
1753  011254  042737  000360  001124          BIC     #360,$GDDAT     ;SAVE BITS 0-3
1754  011262  052737  000100  001124          BIS     #100,$GDDAT     ;EXPECTED 0-3,URPLY 6(C)
1755  011270  111315                          MOVB    (R3),(R5)       ;READ IRR BITS,GROUP 2
1756  011272  023715  001124                  CMP     $GDDAT,(R5)
1757  011276  001401                          BEQ     4$
1758  011300  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1759  011302  005762  000002          4$:     TST     2(R2)           ;READ DBRC FOR RPLY 4(A)
1760  011306  052737  000020  001124          BIS     #20,$GDDAT      ;STORE EXPECTED
1761  011314  111315                          MOVB    (R3),(R5)       ;READ IRR BITS,GROUP 2
1762  011316  023715  001124                  CMP     $GDDAT,(R5)
1763  011322  001401                          BEQ     5$
1764  011324  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1765  011326  005061  000002          5$:     CLR     2(R1)           ;CLEAR DBRB FOR RPY 7(D)
1766  011332  052737  000200  001124          BIS     #200,$GDDAT     ;EXPECTED
1767  011340  111315                          MOVB    (R3),(R5)       ;READ IRR BITS GROUP 2
1768  011342  023715  001124                  CMP     $GDDAT,(R5)
1769  011346  001401                          BEQ     6$
1770  011350  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1771  011352  005763  000002          6$:     TST     2(R3)           ;READ DBRD FOR RPLY 5(B)
1772  011356  052737  000040  001124          BIS     #40,$GDDAT
1773  011364  111315                          MOVB    (R3),(R5)       ;READ IRR BITS GROUP2
1774  011366  023715  001124                  CMP     $GDDAT,(R5)
1775  011372  001401                          BEQ     7$              ;GET NEXT PATTERN
1776  011374  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1777  011376  005724                  7$:     TST     (R4)+           ;INDEX DATA TABLE
1778  011400  020427  013410                  CMP     R4,#ENDDAT      ;CHECK FOR END
1779  011404  001271                          BNE     1$              ;DO NEXT PATTERN
1780
1781
1782                                  ;;*********************************************************************
1783                                  ;*TEST 45        TEST GROUPS 1,2 IN POLLED MODE,NO REPLY
1784                                  ;;*********************************************************************
1785  011406  000004          TST45:  SCOPE
1786  011410  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1787                                                                   ;R0 = CSRA-GROUP 1 CONTROL
1788                                                                   ;R1 = CSRB-GROUP 1 DATA
1789                                                                   ;R2 = CSRC-GROUP 2 CONTROL
1790                                                                   ;R3 = CSRD-GROUP 2 DATA
1791  011414  012704  000002                  MOV     #2,R4           ;TWO PASSES
1792                                                                   ;FIRST PASS CSRA,CSRB = OUTPUT
1793                                                                   ;           CSRC,CSRD = INPUT
1794                                                                   ;SEC PASS   CSRC,CSRD = OUTPUT
1795                                                                   ;           CSRA,CSRB = INPUT
1796  011420  112760  000001  000001  111$:   MOVB    #BIT0,1(R0)     ;SET CSR TO OUTPUT MODE
1797  011426  112761  000001  000001          MOVB    #BIT0,1(R1)     ;SET CSR TO OUTPUT MODE
1798  011434  012760  177777  000002          MOV     #-1,2(R0)       ;SET ALL ONES DBR FOR H TO L
```

```
1799  011442  105010                          CLRB    (R0)            ;CHIP RESET GROUP 1 CSR
1800  011444  105012                          CLRB    (R2)            ;CHIP RESET GROUP 2 CSR
1801  011446  112710  000204                  MOVB    #204,(R0)       ;LOAD MODE BITS FOR POLLED MODE
1802  011452  112712  000204                  MOVB    #204,(R2)       ;LOAD MODE BITS FOR POLLED MODE
1803  011456  112710  000020                  MOVB    #CIRMR,(R0)     ;CLEAR IMR+IRR
1804  011462  112712  000020                  MOVB    #CIRMR,(R2)     ;CLEAR IMR+IRR
1805  011466  005760  000002                  TST     2(R0)           ;READ DBR IN OUTPUT MODE,NO REPLY
1806  011472  012737  000320  001124          MOV     #320,$GDDAT     ;STORE EXPECTED
1807  011500  010037  001122                  MOV     R0,$BDADR       ;STORE ADDRESS
1808  011504  111015                          MOVB    (R0),(R5)       ;READ STATUS
1809  011506  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
1810  011512  023715  001124                  CMP     $GDDAT,(R5)     ;CHECK STATUS
1811  011516  001401                          BEQ     1$
1812  011520  104007                          ERROR   7               ;CHIP STATUS ERROR
1813  011522  010237  001122          1$:     MOV     R2,$BDADR       ;STORE ADDRESS
1814  011526  111215                          MOVB    (R2),(R5)       ;READ STATUS
1815  011530  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
1816  011534  023715  001124                  CMP     $GDDAT,(R5)
1817  011540  001401                          BEQ     2$
1818  011542  104007                          ERROR   7               ;CHIP STATUS ERROR
1819  011544  012762  000000  000002  2$:     MOV     #0,2(R2)        ;WRITE ONLY, DBR INPUT MODE FOR NO REPLY
1820  011552  005761  000002                  TST     2(R1)           ;READ DBR OUTPUT MODE,NO REPLY
1821  011556  012763  000000  000002          MOV     #0,2(R3)        ;WRITE ONLY, DBR INPUT MODE,NO REPLY
1822  011564  010137  001122                  MOV     R1,$BDADR       ;CHIP DATA PORT
1823  011570  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD BITS TO READ IRR
1824  011574  005037  001124                  CLR     $GDDAT          ;IRR SHOULD BE ZERO
1825  011600  111115                          MOVB    (R1),(R5)       ;READ IRR
1826  011602  023715  001124                  CMP     $GDDAT,(R5)
1827  011606  001401                          BEQ     3$
1828  011610  104003                          ERROR   3               ;IRR ERROR
1829  011612  010337  001122          3$:     MOV     R3,$BDADR       ;CHIP DATA PORT
1830  011616  112712  000250                  MOVB    #MIRR,(R2)      ;LOAD MODE BITS TO READ IRR
1831  011622  111315                          MOVB    (R3),(R5)       ;READ IRR BITS
1832  011624  023715  001124                  CMP     $GDDAT,(R5)
1833  011630  001401                          BEQ     4$
1834  011632  104003                          ERROR   3               ;IRR ERROR
1835  011634  005304                  4$:     DEC     R4              ;FINISHED TWO PASSES
1836  011636  001413                          BEQ     TST46           ;BR IF EQUAL
1837  011640  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1838  011644  013700  001352                  MOV     DRCSC,R0        ;SET UP FOR NEXT PASS
1839  011650  013701  001356                  MOV     DRCSD,R1
1840  011654  013702  001342                  MOV     DRCSA,R2
1841  011660  013703  001346                  MOV     DRCSB,R3
1842  011664  000655                          BR      111$            ;DO NEXT PASS
1843
1844                                  ;;*****************************************************************
1845                                  ;*TEST 46       TEST POLLED MODE;CSRS C,D=OUT A,B=IN,ACTIVE LOW
1846                                  ;;*****************************************************************
1847  011666  000004                  TST46:  SCOPE
1848  011670  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1849                                                                   ;R0 = CSRA-GROUP 1 CONTROL
1850                                                                   ;R1 = CSRB-GROUP 1 DATA
1851                                                                   ;R2 = CSRC-GROUP 2 CONTROL
1852                                                                   ;R3 = CSRD-GROUP 2 DATA
1853  011674  012737  011736  001110          MOV     #1$,$LPERR      ;SCOPE LOOP RETURN
1854  011702  012704  013110                  MOV     #BEGPAT,R4      ;PATTERN TABLE
```

```
1855  011706  112762  000001  000001          MOVB    #BIT0,1(R2)     ;SET CSRC TO OUTPUT MODE
1856  011714  112763  000001  000001          MOVB    #BIT0,1(R3)     ;SET CSRD TO OUTPUT MODE
1857  011722  105010                          CLRB    (R0)            ;CHIP RESET GROUP 1 CSRA
1858  011724  105012                          CLRB    (R2)            ;CHIP RESET GROUP 2 CSRC
1859  011726  112710  000204                  MOVB    #204,(R0)       ;LOAD MODE BITS FOR POLLED MODE,GR 1
1860  011732  112712  000204                  MOVB    #204,(R2)       ;LOAD MODE BITS FOR POLLED MODE,GR2
1861  011736  011462  000002          1$:     MOV     (R4),2(R2)      ;SET PATTERN DBRC FOR H TO L
1862  011742  112710  000020                  MOVB    #CIRMR,(R0)     ;CLEAR IMR+IRR GROUP 1
1863  011746  112712  000020                  MOVB    #CIRMR,(R2)     ;CLEAR IMR+IRR GROUP 2
1864  011752  005062  000002                  CLR     2(R2)           ;CLEAR DBRC,ACTIVE LOW WILL SET RPLY A
1865  011756  010137  001122          2$:     MOV     R1,$BDADR       ; GROUP 1 DATA PORT
1866  011762  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD BITS TO READ IRR
1867  011766  111437  001124                  MOVB    (R4),$GDDAT
1868  011772  111115                          MOVB    (R1),(R5)       ;READ IRR,GROUP 1
1869  011774  023715  001124                  CMP     $GDDAT,(R5)
1870  012000  001401                          BEQ     3$
1871  012002  104003                          ERROR   3               ;IRR ERROR,GROUP 1
1872  012004  010337  001122          3$:     MOV     R3,$BDADR       ;GROUP 2 DATA PORT
1873  012010  112712  000250                  MOVB    #MIRR,(R2)      ;LOAD MODE BITS TO READ IRR GROUP2
1874  012014  116437  000001  001124          MOVB    1(R4),$GDDAT
1875  012022  042737  000360  001124          BIC     #360,$GDDAT     ;SAVE BITS 0-3
1876  012030  052737  000020  001124          BIS     #20,$GDDAT      ;EXPECTED 0-3,URPLY 4(A)
1877  012036  111315                          MOVB    (R3),(R5)       ;READ IRR BITS,GROUP 2
1878  012040  023715  001124                  CMP     $GDDAT,(R5)
1879  012044  001401                          BEQ     4$
1880  012046  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1881  012050  005760  000002          4$:     TST     2(R0)           ;READ DBRA FOR RPLY 6(C)
1882  012054  052737  000100  001124          BIS     #100,$GDDAT     ;STORE EXPECTED
1883  012062  111315                          MOVB    (R3),(R5)       ;READ IRR BITS,GROUP 2
1884  012064  023715  001124                  CMP     $GDDAT,(R5)
1885  012070  001401                          BEQ     5$
1886  012072  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1887  012074  005063  000002          5$:     CLR     2(R3)           ;CLEAR DBRD FOR RPY 5(B)
1888  012100  052737  000040  001124          BIS     #40,$GDDAT      ;EXPECTED
1889  012106  111315                          MOVB    (R3),(R5)       ;READ IRR BITS GROUP 2
1890  012110  023715  001124                  CMP     $GDDAT,(R5)
1891  012114  001401                          BEQ     6$
1892  012116  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1893  012120  005761  000002          6$:     TST     2(R1)           ;READ DBRB FOR RPLY 7(D)
1894  012124  052737  000200  001124          BIS     #200,$GDDAT
1895  012132  111315                          MOVB    (R3),(R5)       ;READ IRR BITS GROUP2
1896  012134  023715  001124                  CMP     $GDDAT,(R5)
1897  012140  001401                          BEQ     7$
1898  012142  104003                          ERROR   3               ;IRR ERROR,GROUP 2
1899  012144  005724                  7$:     TST     (R4)+           ;INDEX DATA TABLE
1900  012146  020427  013410                  CMP     R4,#ENDDAT      ;CHECK FOR END
1901  012152  001271                          BNE     1$              ;DO NEXT PATTERN
1902
1903
1904
1905                                  ;;*******************************************************************
1906                                  ;*TEST 47       TEST IRR BITS WITH DATA PAT.,POLLED MODE,ACT. HIGH
1907                                  ;*******************************************************************
1908  012154  000004                  TST47:  SCOPE
1909  012156  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1910  012162  012703  013110                  MOV     #BEGPAT,R3      ;GET DATA PATTERN TABLE
```

```
1911  012166  012737  000001  001110          MOV     #1,$LPERR       ;SET UP SCOPE ADDRESS
1912  012174  013700  001342                  MOV     DRCSA,R0
1913  012200  013701  001346                  MOV     DRCSB,R1
1914  012204  013702  001356                  MOV     DRCSD,R2
1915  012210  112760  000001  000001          MOVB    #BIT0,1(R0)     ;CSRA OUTPUT MODE
1916  012216  112761  000001  000001          MOVB    #BIT0,1(R1)     ;CSRB OUTPUT MODE
1917  012224  112710  000224                  MOVB    #224,(R0)       ;LOAD MODE BITS FOR POLLED MODE,GP1
1918  012230  112760  000224  000010          MOVB    #224,10(R0)     ;LOAD MODE BITS FOR POLLED MODE,GP2
1919  012236  005060  000002          1$:     CLR     2(R0)           ;CLEAR DBRA
1920  012242  112710  000020                  MOVB    #CIRMR,(R0)     ;CLEAR IMR +IRR GROUP 1
1921  012246  112760  000020  000010          MOVB    #CIRMR,10(R0)   ;CLEAR IMR + IRR GROUP 2
1922  012254  011360  000002                  MOV     (R3),2(R0)      ;STORE PATTERN INTO DBRA
1923  012260  010137  001122                  MOV     R1,$BDADR       ;CSRB ADDRESS
1924  012264  112710  000250                  MOVB    #MIRR,(R0)      ;LOAD BITS TO READ IRR1
1925  012270  111337  001124                  MOVB    (R3),$GDDAT
1926  012274  111115                          MOVB    (R1),(R5)       ;READ IRR,GP1
1927  012276  023715  001124                  CMP     $GDDAT,(R5)
1928  012302  001401                          BEQ     2$
1929  012304  104003                          ERROR   3               ;IRR ERROR,GP1
1930  012306  010237  001122          2$:     MOV     R2,$BDADR       ;CSRD ADDRESS
1931  012312  112760  000250  000010          MOVB    #MIRR,10(R0)    ;SET MODE TO READ IRR BITS GROUP 2
1932  012320  116337  000001  001124          MOVB    1(R3),$GDDAT    ;BUILD EXPECTED DATA
1933  012326  042737  000360  001124          BIC     #360,$GDDAT     ;BITS 0-3 EXPECTED
1934  012334  052737  000100  001124          BIS     #100,$GDDAT     ;"A" REPLY EXPECTED
1935  012342  111215                          MOVB    (R2),(R5)       ;READ IRR2
1936  012344  023715  001124                  CMP     $GDDAT,(R5)
1937  012350  001401                          BEQ     3$
1938  012352  104003                          ERROR   3               ;IRR GROUP 2
1939  012354  005723                  3$:     TST     (R3)+           ;INDEX DATA TABLE
1940  012356  020327  013410                  CMP     R3,#ENDDAT      ;CHECK FOR PATTERN END
1941  012362  001325                          BNE     1$              ;DO NEXT PATTERN
1942
1943
1944                                  ;;*********************************************************************
1945                                  ;*TEST 50        TEST CSRA AND CSRB WITH RESET
1946                                  ;;*********************************************************************
1947  012364  000004                  TST50:  SCOPE
1948  012366  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
1949  012374  004737  013002                  JSR     PC,CLRCSR       ;CLEAR ALL CSRS
1950  012400  013700  001342                  MOV     DRCSA,R0
1951  012404  013701  001346                  MOV     DRCSB,R1
1952  012410  112760  001001  000001          MOVB    #IE!BIT0,1(R0)  ;CSRA TO OUTPUT MODE
1953  012416  112761  000001  000001          MOVB    #BIT0,1(R1)     ;SET CSRB TO OUTPUT MODE
1954  012424  010037  001122                  MOV     R0,$BDADR       ;STORE CSRA ADDRESS
1955  012430  012737  100300  001124          MOV     #100300,$GDDAT  ;RDY + CHIP STATUS EXP'D
1956  012436  000005                          RESET                   ;INITIALIZE
1957  012440  011015                          MOV     (R0),(R5)       ;STORE REC'D
1958  012442  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
1959  012446  023715  001124                  CMP     $GDDAT,(R5)     ;MAKE COMPARE
1960  012452  001401                          BEQ     1$
1961  012454  104002                          ERROR   2               ;CSRA ERROR ON RESET
1962  012456  012737  100000  001124  1$:     MOV     #100000,$GDDAT  ;STORE EXPECTED
1963  012464  010137  001122                  MOV     R1,$BDADR       ;CHECK CSRB
1964  012470  011115                          MOV     (R1),(R5)       ;STORE IN $BDDAT
1965  012472  023715  001124                  CMP     $GDDAT,(R5)     ;MAKE COMPARE
1966  012476  001401                          BEQ     TST51           ;;BR IF EQUAL
```

```
 1967  012500  104002                          ERROR   2               ;CSRB ERROR WITH RESET
 1968
 1969
 1970                                  ;;*************************************************************************
 1971                                  ;*TEST 51        TEST CSRC AND CSRD WITH RESET
 1972                                  ;;*************************************************************************
 1973  012502  000004                  TST51:  SCOPE
 1974  012504  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
 1975  012512  004737  013002                  JSR     PC,CLRCSR
 1976  012516  013702  001352                  MOV     DRCSC,R2
 1977  012522  013703  001356                  MOV     DRCSD,R3
 1978  012526  112762  000001  000001         MOVB    #BIT0,1(R2)     ;SET CSRC TO OUTPUT MODE
 1979  012534  112763  000001  000001         MOVB    #BIT0,1(R3)     ;SET CSRD TO OUTPUT MODE
 1980  012542  010237  001122                  MOV     R2,$BDADR       ;STORE ADDRESS
 1981  012546  012737  100200  001124          MOV     #100200,$GDDAT  ;RDY + CHIP STATUS
 1982  012554  000005                          RESET                   ;INITIALIZE
 1983  012556  011215                          MOV     (R2),(R5)       ;STORE CSRC
 1984  012560  042715  000007                  BIC     #7,(R5)         ;CLEAR UNDEFINED BITS
 1985  012564  023715  001124                  CMP     $GDDAT,(R5)     ;MAKE COMPARE
 1986  012570  001401                          BEQ     1$
 1987  012572  104002                          ERROR   2               ;CSRC ERROR WITH RESET
 1988  012574  010337  001122          1$:     MOV     R3,$BDADR       ;CHECK CSRD
 1989  012600  012737  100000  001124          MOV     #100000,$GDDAT  ;STORE EXPECTED
 1990  012606  011315                          MOV     (R3),(R5)       ;READ CSRD
 1991  012610  023715  001124                  CMP     $GDDAT,(R5)     ;MAKE COMPARE
 1992  012614  001401                          BEQ     2$
 1993  012616  104002                          ERROR   2               ;CSRD ERROR ON RESET
 1994  012620                          2$:
 1995
```

```
 1996
 1997
 1998
 1999                                   ;DON'T REPORT 'EOP'' UNTIL ALL SELECTED DRV11-J'S HAVE BEEN TESTED.
 2000  012620  013701  001342           NXDEV:  MOV     DRCSA,R1        ;INIT TO SETUP NEXT DRV11J ADDRESSES
 2001  012624  000241                   NXDEV1: CLC                     ;CLEAR CARRY FOR DEVICE MAP
 2002  012626  006037  001364                   ROR     DMAP            ;LOOK FOR NEXT DRV11J
 2003  012632  001412                           BEQ     $EOP            ;BR IF ALL TESTED
 2004  012634  162701  000020                   SUB     #20,R1          ;NEXT DRV11-J STARTS -20
 2005  012640  005237  001202                   INC     $UNIT           ;UPDATE UNIT NUMBER
 2006  012644  032737  000001  001364         BIT     #1,DMAP         ;IS UNIT SELECTED?
 2007  012652  001764                           BEQ     NXDEV1          ;NEXT,IF NOT SELECTED
 2008  012654  000137  002032                   JMP     NEXPAS          ;TEST NEXT DRV11-J
 2009                                  .SBTTL  END OF PASS ROUTINE
 2010
 2011                                  ;;**********************************************************************
 2012                                  ;*INCREMENT THE PASS NUMBER ($PASS)
 2013                                  ;*TYPE 'END PASS #XXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER)
 2014                                  ;*IF THERES A MONITOR GO TO IT
 2015                                  ;*IF THERE ISN'T JUMP TO START1
 2016
 2017  012660                          $EOP:
 2018  012660  000240                           NOP
 2019  012662  005037  001102                   CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 2020  012666  005037  001160                   CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 2021  012672  005237  001176                   INC     $PASS           ;;INCREMENT THE PASS NUMBER
 2022  012676  042737  100000  001176         BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 2023  012704  005327                           DEC     (PC)+           ;;LOOP?
 2024  012706  000001                  $EOPCT: .WORD   1
 2025  012710  003022                           BGT     $DOAGN          ;;YES
 2026  012712  012737                           MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 2027  012714  000001                  $ENDCT: .WORD   1
 2028  012716  012706                           $EOPCT
 2029  012720  104401  012765                   TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
 2030  012724  013746  001176                   MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 2031  012730  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 2032  012732  104401  012762                   TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 2033  012736  013700  000042           $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 2034  012742  001405                           BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 2035  012744  000005                           RESET                   ;;CLEAR THE WORLD
 2036  012746  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 2037  012750  000240                           NOP                     ;;SAVE ROOM
 2038  012752  000240                           NOP                     ;;FOR
 2039  012754  000240                           NOP                     ;;ACT11
 2040  012756                          $DOAGN:
 2041  012756  000137                           JMP     @(PC)+          ;;RETURN
 2042  012760  001766                  $RTNAD: .WORD   START1
 2043  012762     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
 2044  012765     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
 2045  012772  050040  051501  020123
 2046  013000  000043
```

```
2047                                    .SBTTL  PROGRAM SUBROUTINES
2048
2049
2050                                    ;;*****************************************************************
2051                                    ;CLEAR ALL CONTROL/STATUS REGISTERS
2052                                    ;INIT REGISTERS R0-R4 WITH CSRA-CSRD
2053                                    ;AND STORE $BDDAT INTO R5.
2054                                    ;;*****************************************************************
2055  013002  012705  001126           CLRCSR: MOV     #$BDDAT,R5      ;BAD DATA STORAGE IN R5
2056  013006  013700  001342                   MOV     DRCSA,R0        ;CSRA ADDRESS TO R0
2057  013012  013701  001346                   MOV     DRCSB,R1        ;CSRB ADDRESS TO R1
2058  013016  013702  001352                   MOV     DRCSC,R2        ;CSRC ADDRESS TO R2
2059  013022  013703  001356                   MOV     DRCSD,R3        ;CSRD ADDRESS TO R3
2060  013026  005037  001124                   CLR     $GDDAT          ;CLEAR EXPECTED
2061  013032  005015                           CLR     (R5)            ;CLEAR REC'D
2062  013034  005010                           CLR     (R0)            ;CLEAR CSRA;CHIP RESET GROUP 1
2063  013036  105061  000001                   CLRB    1(R1)           ;CLEAR HIGH BYTE CSRB
2064  013042  005012                           CLR     (R2)            ;CLEAR CSRC;CHIP RESET GROUP 2
2065  013044  105063  000001                   CLRB    1(R3)           ;CLEAR HIGH BYTE CSRD
2066  013050  000207                           RTS     PC
2067
2068
2069                                    ;CLEAR IRR REGISTERS,GROUP 1,GROUP 2 WITH CHIP RESET
2070  013052  010046                   CLRIRR: MOV     R0,-(SP)
2071  013054  013700  001342                   MOV     DRCSA,R0        ;START OF CSR ADDRESS
2072  013060  105060  000011                   CLRB    11(R0)          ;CSRC TO INPUT MODE
2073  013064  112760  000001  000001          MOVB    #BIT0,1(R0)     ;CSRA TO OUTPUT MODE
2074  013072  005060  000002                   CLR     2(R0)           ;CLEAR DBRA
2075  013076  105010                           CLRB    (R0)            ;CHIP RESET OF GROUP1
2076  013100  105060  000010                   CLRB    10(R0)          ;CHIP RESET OF GROUP 2
2077  013104  012600                           MOV     (SP)+,R0        ;RESTORE REGISTER
2078  013106  000207                           RTS     PC              ;EXIT
2079
```

```
2080
2081                                    .SBTTL  PATTERNS FOR REGISTER R/W
2082                                    ;
2083                                    ;PATTERNS USED FOR LOADING/READING REGISTERS
2084
2085  013110  000000                    BEGPAT: 0                       ;GROWING 1
2086  013112  000001                    BGPAT1: 1
2087  013114  000003                            3
2088  013116  000007                            7
2089  013120  000017                            17
2090  013122  000037                            37
2091  013124  000077                            77
2092  013126  000177                            177
2093  013130  000377                            377
2094  013132  000777                    EDCHP1: 777
2095  013134  001777                            1777
2096  013136  003777                            3777
2097  013140  007777                            7777
2098  013142  017777                            17777
2099  013144  037777                            37777
2100  013146  077777                            77777
2101  013150  177777                            177777
2102  013152  177776                    BGCHP2: 177776                  ;GROWING 0
2103  013154  177774                            177774
2104  013156  177770                            177770
2105  013160  177760                            177760
2106  013162  177740                            177740
2107  013164  177700                            177700
2108  013166  177600                            177600
2109  013170  177400                    EDCHP2: 177400
2110  013172  177000                            177000
2111  013174  176000                            176000
2112  013176  174000                            174000
2113  013200  170000                            170000
2114  013202  160000                            160000
2115  013204  140000                            140000
2116  013206  100000                            100000
2117
2118  013210  000000                            000000
2119  013212  000001                    BGCHP3: 1                       ;WALKING 1
2120  013214  000002                            2
2121  013216  000004                            4
2122  013220  000010                            10
2123  013222  000020                            20
2124  013224  000040                            40
2125  013226  000100                            100
2126  013230  000200                            200
2127  013232  000400                    EDCHP3: 400
2128  013234  001000                            1000
2129  013236  002000                            2000
2130  013240  004000                            4000
2131  013242  010000                            10000
2132  013244  020000                            20000
2133  013246  040000                            40000
2134  013250  100000                            100000
2135  013252  177777                            177777                  ;WALKING 0
```

```
2136  013254  177776                  BGCHP4: 177776
2137  013256  177775                          177775
2138  013260  177773                          177773
2139  013262  177767                          177767
2140  013264  177757                          177757
2141  013266  177737                          177737
2142  013270  177677                          177677
2143  013272  177577                          177577
2144  013274  177377                  EDCHP4: 177377
2145  013276  176777                          176777
2146  013300  175777                          175777
2147  013302  173777                          173777
2148  013304  167777                          167777
2149  013306  157777                          157777
2150  013310  137777                          137777
2151  013312  077777                          077777
2152  013314  177777                          177777
2153  013316  000000                  ENDPAT: 000000
2154
2155                                  ;DATA  PATTERNS
2156  013320  155555                  PATDAT: 155555
2157  013322  133333                          133333
2158  013324  066666                          066666
2159  013326  125252                          125252
2160  013330  052525                          052525
2161  013332  177777                          177777
2162  013334  000000                          000000
2163  013336  107070                          107070
2164  013340  070707                          070707
2165  013342  144444                          144444
2166  013344  033333                          033333
2167  013346  011111                          011111
2168  013350  022222                          022222
2169  013352  044444                          044444
2170  013354  111111                          111111
2171  013356  166666                          166666
2172  013360  010421                          010421
2173  013362  021042                          021042
2174  013364  031463                          031463
2175  013366  042104                          042104
2176  013370  063146                          063146
2177  013372  073567                          073567
2178  013374  104210                          104210
2179  013376  114631                          114631
2180  013400  135673                          135673
2181  013402  146314                          146314
2182  013404  156735                          156735
2183  013406  167356                          167356
2184  013410  000000                  ENDDAT: 000000
2185
```

```
                                        .SBTTL  SYSMAC ROUTINES

                                .SBTTL  TYPE ROUTINE

                                ;;*********************************************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*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 ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*

013412  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
013416  100002                          BPL     1$              ;;BR IF YES
013420  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
013422  000430                          BR      3$              ;;LEAVE
013424  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
013426  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
013432  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
013440  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
013442  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
013450  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
013452  010037  013462                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
013456  004737  013702                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
013462  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
013464  132737  000040  001211  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
013472  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
013474  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
013476  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
013500  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
013502  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
013504  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
013510  000002                          RTI                     ;;RETURN
013512  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
013516  001430                          BEQ     8$
013520  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
013524  001006                          BNE     5$
013526  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
013530  104401                          TYPE                    ;;TYPE A CR AND LF
013532  001165                          $CRLF
013534  105037  013670                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
013540  000755                          BR      2$              ;;GET NEXT CHARACTER
013542  004737  013624          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
013546  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
013552  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
013554  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
013560  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
013564  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
2242  013566  004737  013624                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
2243  013572  105337  013670                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
2244  013576  000770                          BR      7$              ;;LOOP
2245
2246                                  ;HORIZONTAL TAB PROCESSOR
2247
2248  013600  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
2249  013604  004737  013624          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
2250  013610  132737  000007  013670          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
2251  013616  001372                          BNE     9$              ;;TAB STOP
2252  013620  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
2253  013622  000724                          BR      2$              ;;GET NEXT CHARACTER
2254  013624  105777  165320          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
2255  013630  100375                          BPL     $TYPEC
2256  013632  116677  000002  165312          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
2257  013640  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
2258  013646  001003                          BNE     1$              ;;BRANCH IF NO
2259  013650  105037  013670                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
2260  013654  000406                          BR      $TYPEX          ;;EXIT
2261  013656  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
2262  013664  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
2263  013666  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
2264  013670  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
2265  013672  000207          $TYPEX: RTS     PC
2266
2267                                  .SBTTL  APT COMMUNICATIONS ROUTINE
2268
2269                                  ;;**************************************************************************
2270  013674  112737  000001  014140  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
2271  013702  112737  000001  014136  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
2272  013710  000403                          BR      $ATYC
2273  013712  112737  000001  014140  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
2274  013720          $ATYC:
2275  013720  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2276  013722  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2277  013724  105737  014136                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
2278  013730  001450                          BEQ     5$              ;;IF NOT:  BR
2279  013732  122737  000001  001210          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
2280  013740  001031                          BNE     3$              ;;IF NOT:  BR
2281  013742  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
2282  013750  001425                          BEQ     3$              ;;IF NOT:  BR
2283  013752  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
2284  013756  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
2285  013764  005737  001170          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
2286  013770  001375                          BNE     1$              ;;IF NOT:  WAIT
2287  013772  010037  001204                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
2288  013776  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
2289  014000  001376                          BNE     2$
2290  014002  163700  001204                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
2291  014006  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
2292  014010  010037  001206                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
2293  014014  012737  000004  001170          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
2294  014022  000413                          BR      5$
2295  014024  017637  000004  014050  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
2296  014032  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
2297  014040  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
```

```
2298  014044  004737  013412              JSR     PC,$TYPE        ;;CALL TYPE MACRO
2299  014050  000000              4$:     .WORD   0
2300  014052                      5$:
2301  014052  105737  014140      10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
2302  014056  001416              12$     BEQ     12$             ;;IF NOT:  BR
2303  014060  005737  001210              TST     $ENV            ;;RUNNING UNDER APT?
2304  014064  001413              12$     BEQ     12$             ;;IF NOT:  BR
2305  014066  005737  001170      11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
2306  014072  001375              11$     BNE     11$             ;;IF NOT:  WAIT
2307  014074  017637  000004  001172      MOV     @4(SP),$FATAL   ;;GET ERROR #
2308  014102  062766  000002  000004      ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
2309  014110  005237  001170              INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
2310  014114  105037  014140      12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
2311  014120  105037  014137              CLRB    $LFLG           ;;CLEAR LOG FLAG
2312  014124  105037  014136              CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
2313  014130  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
2314  014132  012600                      MOV     (SP)+,R0        ;;POP STACK INTO R0
2315  014134  000207                      RTS     PC              ;;RETURN
2316  014136     000      $MFLG:  .BYTE   0               ;;MESSG. FLAG
2317  014137     000      $LFLG:  .BYTE   0               ;;LOG FLAG
2318  014140     000      $FFLG:  .BYTE   0               ;;FATAL FLAG
2319          014142              .EVEN
2320          000200      APTSIZE=200
2321          000001      APTENV=001
2322          000100      APTSPOOL=100
2323          000040      APTCSUP=040
2324                      .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
2325
2326                      ;;**********************************************************************
2327                      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
2328                      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
2329                      ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
2330                      ;*CALL:
2331                      ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2332                      ;*      TYPOS                   ;;CALL FOR TYPEOUT
2333                      ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
2334                      ;*      .BYTE   M               ;;M=1 OR 0
2335                      ;*                                      ;;1=TYPE LEADING ZEROS
2336                      ;*                                      ;;0=SUPPRESS LEADING ZEROS
2337                      ;*
2338                      ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
2339                      ;*$TYPOS OR $TYPOC
2340                      ;*CALL:
2341                      ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2342                      ;*      TYPON                   ;;CALL FOR TYPEOUT
2343                      ;*
2344                      ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
2345                      ;*CALL:
2346                      ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
2347                      ;*      TYPOC                   ;;CALL FOR TYPEOUT
2348
2349  014142  017646  000000      $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
2350  014146  116637  000001  014365      MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
2351  014154  112637  014367              MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
2352  014160  062716  000002              ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
2353  014164  000406                      BR      $TYPON
```

```
2354  014166  112737  000001  014365  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
2355  014174  112737  000006  014367          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
2356  014202  112737  000005  014364  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
2357  014210  010346                          MOV     R3,-(SP)        ;;SAVE R3
2358  014212  010446                          MOV     R4,-(SP)        ;;SAVE R4
2359  014214  010546                          MOV     R5,-(SP)        ;;SAVE R5
2360  014216  113704  014367                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
2361  014222  005404                          NEG     R4
2362  014224  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
2363  014230  110437  014366                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
2364  014234  113704  014365                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
2365  014240  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
2366  014244  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
2367  014246  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
2368  014250  000404                          BR      3$              ;;GO DO MSB
2369  014252  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
2370  014254  006105                          ROL     R5
2371  014256  006105                          ROL     R5
2372  014260  010503                          MOV     R5,R3
2373  014262  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
2374  014264  105337  014366                  DECB    $OMODE          ;;TYPE THIS DIGIT?
2375  014270  100016                          BPL     7$              ;;BR IF NO
2376  014272  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
2377  014276  001002                          BNE     4$              ;;TEST FOR 0
2378  014300  005704                          TST     R4              ;;SUPPRESS THIS 0?
2379  014302  001403                          BEQ     5$              ;;BR IF YES
2380  014304  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
2381  014306  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
2382  014312  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
2383  014316  110337  014362                  MOVB    R3,8$           ;;SAVE FOR TYPING
2384  014322  104401  014362                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
2385  014326  105337  014364          7$:     DECB    $OCNT           ;;COUNT BY 1
2386  014332  003347                          BGT     2$              ;;BR IF MORE TO DO
2387  014334  002402                          BLT     6$              ;;BR IF DONE
2388  014336  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
2389  014340  000744                          BR      2$              ;;GO DO THE LAST DIGIT
2390  014342  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
2391  014344  012604                          MOV     (SP)+,R4        ;;RESTORE R4
2392  014346  012603                          MOV     (SP)+,R3        ;;RESTORE R3
2393  014350  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
2394  014356  012616                          MOV     (SP)+,(SP)
2395  014360  000002                          RTI                     ;;RETURN
2396  014362     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
2397  014363     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
2398  014364     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
2399  014365     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
2400  014366  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
2401                          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
2402
2403                          ;;**************************************************************************
2404                          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
2405                          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
2406                          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
2407                          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
2408                          ;*REPLACED WITH SPACES.
2409                          ;*CALL
```

```
2410                                            MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
2411                                            TYPDS                   ;;GO TO THE ROUTINE
2412
2413  014370                                $TYPDS:
2414  014370  010046                               MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2415  014372  010146                               MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2416  014374  010246                               MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2417  014376  010346                               MOV     R3,-(SP)        ;;PUSH R3 ON STACK
2418  014400  010546                               MOV     R5,-(SP)        ;;PUSH R5 ON STACK
2419  014402  012746  020200                       MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
2420  014406  016605  000020                       MOV     20(SP),R5       ;;GET THE INPUT NUMBER
2421  014412  100004                               BPL     1$              ;;BR IF INPUT IS POS.
2422  014414  005405                               NEG     R5              ;;MAKE THE BINARY NUMBER POS.
2423  014416  112766  000055  000001               MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
2424  014424  005000                        1$:    CLR     R0              ;;ZERO THE CONSTANTS INDEX
2425  014426  012703  014604                       MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
2426  014432  112723  000040                       MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
2427  014436  005002                        2$:    CLR     R2              ;;CLEAR THE BCD NUMBER
2428  014440  016001  014574                       MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
2429  014444  160105                        3$:    SUB     R1,R5           ;;FORM THIS BCD DIGIT
2430  014446  002402                               BLT     4$              ;;BR IF DONE
2431  014450  005202                               INC     R2              ;;INCREASE THE BCD DIGIT BY 1
2432  014452  000774                               BR      3$
2433  014454  060105                        4$:    ADD     R1,R5           ;;ADD BACK THE CONSTANT
2434  014456  005702                               TST     R2              ;;CHECK IF BCD DIGIT=0
2435  014460  001002                               BNE     5$              ;;FALL THROUGH IF 0
2436  014462  105716                               TSTB    (SP)            ;;STILL DOING LEADING 0'S?
2437  014464  100407                               BMI     7$              ;;BR IF YES
2438  014466  106316                        5$:    ASLB    (SP)            ;;MSD?
2439  014470  103003                               BCC     6$              ;;BR IF NO
2440  014472  116663  000001  177777               MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
2441  014500  052702  000060                6$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
2442  014504  052702  000040                7$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
2443  014510  110223                               MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
2444  014512  005720                               TST     (R0)+           ;;JUST INCREMENTING
2445  014514  020027  000010                       CMP     R0,#10          ;;CHECK THE TABLE INDEX
2446  014520  002746                               BLT     2$              ;;GO DO THE NEXT DIGIT
2447  014522  003002                               BGT     8$              ;;GO TO EXIT
2448  014524  010502                               MOV     R5,R2           ;;GET THE LSD
2449  014526  000764                               BR      6$              ;;GO CHANGE TO ASCII
2450  014530  105726                        8$:    TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
2451  014532  100003                               BPL     9$              ;;BR IF NO
2452  014534  116663  177777  177776               MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
2453  014542  105013                        9$:    CLRB    (R3)            ;;SET THE TERMINATOR
2454  014544  012605                               MOV     (SP)+,R5        ;;POP STACK INTO R5
2455  014546  012603                               MOV     (SP)+,R3        ;;POP STACK INTO R3
2456  014550  012602                               MOV     (SP)+,R2        ;;POP STACK INTO R2
2457  014552  012601                               MOV     (SP)+,R1        ;;POP STACK INTO R1
2458  014554  012600                               MOV     (SP)+,R0        ;;POP STACK INTO R0
2459  014556  104401  014604                       TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
2460  014562  016666  000002  000004               MOV     2(SP),4(SP)     ;;ADJUST THE STACK
2461  014570  012616                               MOV     (SP)+,(SP)
2462  014572  000002                               RTI                     ;;RETURN TO USER
2463  014574  023420                        $DTBL:: 10000.
2464  014576  001750                               1000.
2465  014600  000144                               100.
```

```
2466   014602  000012                          10.
2467   014604  000004                  $DBLK:  .BLKW   4
2468                                   .SBTTL  ERROR HANDLER ROUTINE
2469
2470                                   ;;*******************************************************************
2471                                   ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT
2472                                   ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
2473                                   ;*AND GO TO SWRCK ON ERROR
2474                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
2475                                   ;*SW15=1         HALT ON ERROR
2476                                   ;*SW13=1         INHIBIT ERROR TYPEOUTS
2477                                   ;*SW09=1         LOOP ON ERROR
2478                                   ;*CALL
2479                                   ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
2480
2481   014614                          $ERROR:
2482   014614  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
2483   014616  104407                          CKSWR           ;GO LOOK FOR SWR CHANGE
2484   014620  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
2485   014624  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
2486   014626  013777  001102  164306          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
2487   014634  005237  001112                  INC     $ERTTL          ;;INC THE ERROR COUNT
2488   014640  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
2489   014644  162737  000002  001116          SUB     #2,$ERRPC
2490   014652  117737  164240  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
2491   014660  032777  020000  164252          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
2492   014666  001004                          BNE     20$             ;;SKIP TYPEOUTS
2493   014670  004737  014770                  JSR     PC,SWRCK        ;;GO TO USER ERROR ROUTINE
2494   014674  104401  001165                  TYPE    ,$CRLF
2495   014700                          20$:
2496   014700  122737  000001  001210          CMPB    #APTENV,$ENV.   ;RUNNING IN APT MODE
2497   014706  001007                          BNE     2$              ;NO SKIP APT ERROR REPORT
2498   014710  113737  001114  014722          MOVB    $ITEMB,21$      ;SET ITEM NUMBER AS ERROR-NUMBER
2499   014716  004737  013712                  JSR     PC,$ATY4        ;REPORT FATAL ERROR TO APT
2500   014722  000                     21$:    .BYTE   0
2501   014723  000                             .BYTE   0
2502   014724  000777                  22$:    BR      22$             ;;APT ERROR LOOP
2503   014726  005777  164206          2$:     TST     @SWR            ;;HALT ON ERROR
2504   014732  100002                          BPL     3$              ;;SKIP IF CONTINUE
2505   014734  000000                          HALT                    ;;HALT ON ERROR
2506   014736  104407                          CKSWR                   ;TEST FOR CHANGE IN SOFT-SWR
2507   014740  032777  001000  164172  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
2508   014746  001402                          BEQ     4$              ;;BR IF NO
2509   014750  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
2510   014754  005737  001162          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
2511   014760  001402                          BEQ     5$              ;;BR IF NONE
2512   014762  013716  001162                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
2513   014766                          5$:
2514   014766  000002                          RTI                     ;;RETURN
2515                                   ;*******************************************************************
2516                                   ;GO TYPE ERROR
2517                                   ;GO UPDATE SOFTWARE SWR IF 'CNTRL/G'
2518                                   ;;*******************************************************************
2519   014770  113737  001102  001362  SWRCK:  MOVB    $TSTNM,TSTNUM   ;SET UP TEST # ON ER
2520   014776  004737  015006                  JSR     PC,$ERRTYP      ;GO TYPE ERROR
2521   015002  104407                          CKSWR                   ;GO LOOK FOR SWR CHANGE
```

```
                                ERROR HANDLER ROUTINE

 2522  015004  000207                        RTS     PC              ;RETURN TO ERROR HANDLER
 2523                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 2524
 2525                                ;;*************************************************************
 2526                                ;*THIS ROUTINE USES THE ''ITEM CONTROL BYTE'' ($ITEMB) TO DETERMINE WHICH
 2527                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE ''ERROR TABLE'' ($ERRTB),
 2528                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 2529
 2530  015006                        $ERRTYP:
 2531  015006  104401  001165                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 2532  015012  010046                        MOV     R0,-(SP)        ;;SAVE R0
 2533  015014  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
 2534  015016  153700  001114                BISB    @#$ITEMB,R0
 2535  015022  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
 2536                                                                ;;TYPE THE PC OF THE ERROR
 2537  015024  013746  001116                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 2538                                                                ;;ERROR ADDRESS
 2539  015030  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2540  015032  000426                        BR      6$              ;;GET OUT
 2541  015034  005300                1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 2542  015036  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
 2543  015040  006300                        ASL     R0
 2544  015042  006300                        ASL     R0
 2545  015044  062700  001252                ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 2546  015050  012037  015060                MOV     (R0)+,2$        ;;PICKUP ''ERROR MESSAGE'' POINTER
 2547  015054  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 2548  015056  104401                        TYPE                    ;;TYPE THE ''ERROR MESSAGE''
 2549  015060  000000                2$:     .WORD   0               ;;''ERROR MESSAGE'' POINTER GOES HERE
 2550  015062  104401  001165                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 2551  015066  012037  015076        3$:     MOV     (R0)+,4$        ;;PICKUP ''DATA HEADER'' POINTER
 2552  015072  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
 2553  015074  104401                        TYPE                    ;;TYPE THE ''DATA HEADER''
 2554  015076  000000                4$:     .WORD   0               ;;''DATA HEADER'' POINTER GOES HERE
 2555  015100  104401  001165                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 2556  015104  011000                5$:     MOV     (R0),R0         ;;PICKUP ''DATA TABLE'' POINTER
 2557  015106  001004                        BNE     7$              ;;GO TYPE THE DATA
 2558  015110  012600                6$:     MOV     (SP)+,R0        ;;RESTORE R0
 2559  015112  104401  001165                TYPE    ,$CRLF          ;;''CARRIAGE RETURN'' & 'LINE FEED''
 2560  015116  000207                        RTS     PC              ;;RETURN
 2561  015120                        7$:
 2562  015120  013046                        MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
 2563  015122  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 2564  015124  005710                        TST     (R0)            ;;IS THERE ANOTHER NUMBER?
 2565  015126  001770                        BEQ     6$              ;;BR IF NO
 2566  015130  104401  015136                TYPE    ,8$             ;;TYPE TWO(2) SPACES
 2567  015134  000771                        BR      7$              ;;LOOP
 2568  015136  020040      000       8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
 2569          015142                        .EVEN
 2570                                .SBTTL  SCOPE HANDLER ROUTINE
 2571
 2572                                ;;*************************************************************
 2573                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 2574                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 2575                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 2576                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 2577                                ;*SW14=1         LOOP ON TEST
```

```
2578                                    ;*SW11=1        INHIBIT ITERATIONS
2579                                    ;*SW09=1        LOOP ON ERROR
2580                                    ;*SW08=1        LOOP ON TEST IN SWR<7:0>
2581                                    ;*CALL
2582                                    ;*      SCOPE           ;;SCOPE=IOT
2583
2584  015142                            $SCOPE:
2585  015142  104407                           CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
2586  015144  032777  040000  163766    1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
2587  015152  001114                            BNE     $OVER           ;;YES IF SW14=1
2588                                    ;#####START OF CODE FOR THE XOR TESTER#####
2589  015154  000416                    $XTSTR: BR      6$              ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
2590                                                                    ;;THIS INSTRUCTION TO A 'NOP' (NOP=240)
2591  015156  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
2592  015162  012737  015202  000004            MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
2593  015170  005737  177060                    TST     @#177060        ;;TIME OUT ON XOR?
2594  015174  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
2595  015200  000463                            BR      $SVLAD          ;;GO TO THE NEXT TEST
2596  015202  022626                    5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
2597  015204  012637  000004                    MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
2598  015210  000423                            BR      7$              ;;LOOP ON THE PRESENT TEST
2599  015212                            6$:;#####END OF CODE FOR THE XOR TESTER#####
2600  015212  032777  000400  163720            BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
2601  015220  001404                            BEQ     2$              ;;BR IF NO
2602  015222  127737  163712  001102            CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
2603  015230  001465                            BEQ     $OVER           ;;BR IF YES
2604  015232  105737  001103            2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
2605  015236  001421                            BEQ     3$              ;;BR IF NO
2606  015240  123737  001115  001103            CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
2607  015246  101015                            BHI     3$              ;;BR IF NO
2608  015250  032777  001000  163662            BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
2609  015256  001404                            BEQ     4$              ;;BR IF NO
2610  015260  013737  001110  001106    7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
2611  015266  000446                            BR      $OVER
2612  015270  105037  001103            4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
2613  015274  005037  001160                    CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
2614  015300  000415                            BR      1$              ;;ESCAPE TO THE NEXT TEST
2615  015302  032777  004000  163630    3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
2616  015310  001011                            BNE     1$              ;;BR IF YES
2617  015312  005737  001176                    TST     $PASS           ;;IF FIRST PASS OF PROGRAM
2618  015316  001406                            BEQ     1$              ;;      INHIBIT ITERATIONS
2619  015320  005237  001104                    INC     $ICNT           ;;INCREMENT ITERATION COUNT
2620  015324  023737  001160  001104            CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
2621  015332  002024                            BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
2622  015334  012737  000001  001104    1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
2623  015342  013737  015420  001160            MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
2624  015350  105237  001102            $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
2625  015354  113737  001102  001174            MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
2626  015362  011637  001106                    MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
2627  015366  011637  001110                    MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
2628  015372  005037  001162                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
2629  015376  112737  000001  001115            MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
2630  015404  013777  001102  163530    $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
2631  015412  013716  001106                    MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
2632  015416  000002                            RTI                     ;;FIXES PS
2633  015420  000062                    $MXCNT: 50.                     ;;MAX. NUMBER OF ITERATIONS
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
                                        ;;*****************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
015422  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
015430  001074                        BNE     15$             ;;BRANCH IF NO
015432  105777  163506                TSTB    @$TKS           ;;CHAR THERE?
015436  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
015440  117746  163502                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
015444  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
015450  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
015454  001062                        BNE     15$             ;;NO, RETURN TO USER
015456  123727  001134  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
015464  001456                        BEQ     15$             ;;BRANCH IF YES
                                
015466  104401  016147                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
015472  104401  016154        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
015476  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
015502  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
015504  104401  016165                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
015510  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
015512  005046                        CLR     -(SP)           ;;THE NEW SWR
015514  105777  163424        7$:     TSTB    @$TKS           ;;CHAR THERE?
015520  100375                        BPL     7$              ;;IF NOT TRY AGAIN
                                
015522  117746  163420                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
015526  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                
                                
                                
015532  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
015536  001005                        BNE     10$             ;;BRANCH IF NOT
015540  104401  016142                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
015544  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
015550  000757                        BR      19$             ;;LET'S TRY IT AGAIN
                                
                                
015552  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
015556  001022                        BNE     16$             ;;BRANCH IF NO
015560  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
015564  001403                        BEQ     11$             ;;BRANCH IF YES
015566  016677  000002  163344        MOV     2(SP),@SWR      ;;SAVE NEW SWR
015574  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
015600  104401  001165        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
015604  123727  001135  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
015612  001003                        BNE     15$             ;;BRANCH IF NOT
015614  012777  000100  163322        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
015622  000002                15$:    RTI                     ;;RETURN
015624  004737  013624        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
015630  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
```

```
2690  015634  002420                          BLT     18$             ;;BRANCH IF YES
2691  015636  021627  000067                  CMP     (SP),#67        ;;CHAR > 77
2692  015642  003015                          BGT     18$             ;;BRANCH IF YES
2693  015644  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
2694  015650  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
2695  015654  001403                          BEQ     17$             ;;BRANCH IF YES
2696  015656  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
2697  015660  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
2698  015662  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
2699  015664  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
2700  015670  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
2701  015674  000707                          BR      7$              ;;GET THE NEXT ONE
2702  015676  104401  001164          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
2703  015702  000720                          BR      20$             ;;SIMULATE CONTROL-U
2704                                .DSABL  LSB
2705
2706
2707                                ;;**************************************************************
2708                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
2709                                ;*CALL:
2710                                ;*      RDCHR   .               ;;INPUT A SINGLE CHARACTER FROM THE TTY
2711                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
2712                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
2713                                ;
2714
2715  015704  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
2716  015706  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
2717  015714  105777  163224          1$:   TSTB    @$TKS           ;;WAIT FOR
2718  015720  100375                        BPL     1$              ;;A CHARACTER
2719  015722  117766  163220  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
2720  015730  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
2721  015736  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
2722  015744  001013                        BNE     3$              ;;BRANCH IF NO
2723  015746  105777  163172          2$:   TSTB    @$TKS           ;;WAIT FOR A CHARACTER
2724  015752  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
2725  015754  117746  163166                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
2726  015760  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
2727  015764  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
2728  015770  001366                        BNE     2$              ;;IF NOT DISCARD IT
2729  015772  000750                        BR      1$              ;;YES, RESUME
2730  015774  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
2731  016002  002407                        BLT     4$              ;;BRANCH IF YES
2732  016004  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
2733  016012  003003                        BGT     4$              ;;BRANCH IF YES
2734  016014  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
2735  016022  000002                  4$:   RTI                     ;;GO BACK TO USER
2736                                ;;**************************************************************
2737                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
2738                                ;*CALL:
2739                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
2740                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
2741                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
2742
2743  016024  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
2744  016026  012703  016132          1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
2745  016032  022703  016142          2$:   CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
```

```
2746  016036  101405                          BLOS    4$                  ;;BR IF YES
2747  016040  104410                          RDCHR                       ;;GO READ ONE CHARACTER FROM THE TTY
2748  016042  112613                          MOVB    (SP)+,(R3)          ;;GET CHARACTER
2749  016044  122713  000177          10$:    CMPB    #177,(R3)           ;;IS IT A RUBOUT
2750  016050  001003                          BNE     3$                  ;;SKIP IF NOT
2751  016052  104401  001164          4$:     TYPE    ,$QUES              ;;TYPE A '?'
2752  016056  000763                          BR      1$                  ;;CLEAR THE BUFFER AND LOOP
2753  016060  111337  016130          3$:     MOVB    (R3),9$             ;;ECHO THE CHARACTER
2754  016064  104401  016130                  TYPE    ,9$
2755  016070  122723  000015                  CMPB    #15,(R3)+           ;;CHECK FOR RETURN
2756  016074  001356                          BNE     2$                  ;;LOOP IF NOT RETURN
2757  016076  105063  177777                  CLRB    -1(R3)              ;;CLEAR RETURN (THE 15)
2758  016102  104401  001166                  TYPE    ,$LF                ;;TYPE A LINE FEED
2759  016106  012603                          MOV     (SP)+,R3            ;;RESTORE R3
2760  016110  011646                          MOV     (SP),-(SP)          ;;ADJUST THE STACK AND PUT ADDRESS OF THE
2761  016112  016666  000004  000002          MOV     4(SP),2(SP)         ;;      FIRST ASCII CHARACTER ON IT
2762  016120  012766  016132  000004          MOV     #$TTYIN,4(SP)
2763  016126  000002                          RTI                         ;;RETURN
2764  016130     000                  9$:     .BYTE   0                   ;;STORAGE FOR ASCII CHAR. TO TYPE
2765  016131     000                          .BYTE   0                   ;;TERMINATOR
2766  016132  000010                  $TTYIN: .BLKB   8.                  ;;RESERVE 8 BYTES FOR TTY INPUT
2767  016142  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>        ;;CONTROL 'U'
2768  016147     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>        ;;CONTROL 'G'
2769  016154  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
2770  016162  020075     000
2771  016165     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
2772  016172  036440  000040
2773                                  .SBTTL  POWER DOWN AND UP ROUTINES
2774
2775                                  ;;*****************************************************************
2776                                  ;POWER DOWN ROUTINE
2777  016176  012737  016342  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
2778  016204  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
2779  016212  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
2780  016214  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2781  016216  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2782  016220  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
2783  016222  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
2784  016224  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
2785  016226  017746  162706                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
2786  016232  010637  016346                  MOV     SP,$SAVR6       ;;SAVE SP
2787  016236  012737  016250  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
2788  016244  000000                          HALT
2789  016246  000776                          BR      .-2             ;;HANG UP
2790
2791                                  ;;*****************************************************************
2792                                  ;POWER UP ROUTINE
2793  016250  012737  016342  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
2794  016256  013706  016346                  MOV     $SAVR6,SP       ;;GET SP
2795  016262  005037  016346                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
2796  016266  005237  016346          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
2797  016272  001375                          BNE     1$              ;;OF  WORD
2798  016274  012677  162640                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
2799  016300  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
2800  016302  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
2801  016304  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
2802  016306  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
2803  016310  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
2804  016312  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
2805  016314  012737  016176  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
2806  016322  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
2807  016330  104401                          TYPE                    ;;REPORT THE POWER FAILURE
2808  016332  016350                $PWRMG:   .WORD   PWRMSG          ;;POWER FAIL MESSAGE POINTER
2809  016334  012716                          MOV     (PC)+,(SP)      ;;RESTART AT START1
2810  016336  001766                $PWRAD:   .WORD   START1          ;;RESTART ADDRESS
2811  016340  000002                          RTI
2812  016342  000000                $ILLUP:   HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
2813  016344  000776                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
2814  016346  000000                $SAVR6:   0                       ;;PUT THE SP HERE
2815  016350  005015  042522  052123  PWRMSG: .ASCIZ  <15><12>/RESTARTED FROM PWR FAIL/
2816  016356  051101  042524  020104
2817  016364  051106  046517  050040
2818  016372  051127  043040  044501
2819  016400  000114
2820                                    .EVEN
2821                                    .SBTTL  TRAP DECODER
2822
2823                                    ;;**************************************************************************
2824                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
2825                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
2826                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
2827                                    ;*GO TO THAT ROUTINE.
2828
2829  016402  010046                $TRAP:    MOV     R0,-(SP)        ;;SAVE R0
2830  016404  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
2831  016410  005740                          TST     -(R0)           ;;BACKUP BY 2
2832  016412  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
2833  016414  006300                          ASL     R0              ;;POSITION FOR INDEXING
2834  016416  016000  016436                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
2835  016422  000200                          RTS     R0              ;;GO TO ROUTINE
2836
2837
2838                                    ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
2839
2840  016424  011646                $TRAP2:   MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
2841  016426  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
2842  016434  000002                          RTI                     ;;RESTORE THE PSW
2843
2844                                    .SBTTL  TRAP TABLE
2845
2846                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
2847                                    ;*BY THE ''TRAP'' INSTRUCTION.
2848
2849                                    ;       ROUTINE
2850                                    ;       -------
2851  016436  016424                $TRPAD:   .WORD   $TRAP2
2852  016440  013412                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
2853  016442  014166                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
2854  016444  014142                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
2855  016446  014202                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
2856  016450  014370                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
2857
```

```
                                        $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
                                        $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
                                        $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
                                        $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE

                                .SBTTL  ASCII MESSAGES
016462  005015  053103  051104  TITLED: .ASCIZ  <15><12>/CVDRCA  DRV11J DIAG TEST PART 1  /<15><12>
016470  040503  020040  051104
016476  030526  045061  042040
016504  040511  020107  042524
016512  052123  050040  051101
016520  020124  020061  006440
016526  000012
016530  005015  051104  030526  TLCABL: .ASCIZ  <15><12>/DRV11J CABLE REQ'D/<15><12>
016536  045061  041440  041101
016544  042514  051040  050505
016552  042047  005015     000

016557     122  043505  052040  EM1:    .ASCIZ  /REG TIMEOUT ER/
016564  046511  047505  052125
016572  042440  000122
016576  042522  020107  042522  EM2:    .ASCIZ  'REG READ/WRITE ER'
016604  042101  053457  044522
016612  042524  042440  000122
016620  051111  020122  042522  EM3:    .ASCIZ  /IRR REG ER/
016626  020107  051105     000
016633     101  051103  051040  EM4:    .ASCIZ  /ACR REG ER/
016640  043505  042440  000122
016646  046511  020122  042522  EM5:    .ASCIZ  /IMR REG ER/
016654  020107  051105     000
016661     111  051123  051040  EM6:    .ASCIZ  /ISR REG ER/
016666  043505  042440  000122
016674  044103  050111  051440  EM7:    .ASCIZ  /CHIP STAT ER/
016702  040524  020124  051105
016710     000
016711     105  051122  041520  DH1:    .ASCIZ  /ERRPC   TSTNUM  BUSADR  EXPCT   RCVD/
016716  020040  052040  052123
016724  052516  020115  041040
016732  051525  042101  020122
016740  042440  050130  052103
016746  020040  051040  053103
016754  000104
016756  051105  050122  020103  DH2:    .ASCIZ  /ERRPC   TSTNUM  BUSADR  ADRS    EXPCT   RCVD/
016764  020040  051524  047124
016772  046525  020040  052502
017000  040523  051104  020040
017006  042101  051522  020040
017014  020040  054105  041520
017022  020124  020040  041522
017030  042126     000

                017034
                                .EVEN
017034  001116  001362  001122  DT1:    $ERRPC,TSTNUM,$BDADR,$GDDAT,$BDDAT,0
017042  001124  001126  000000
017050  001116  001362  001122  DT2:    $ERRPC,TSTNUM,$BDADR,$GDADR,$GDDAT,$BDDAT,0
```

```
 2914  017056  001120  001124  001126
 2915  017064  000000
 2916
 2917          0000C1                          .END
```

```
ABASE  = 164160              135#    269     310     377     378     379     380     381     382     383     384
ACDW1  = 000000              269     312
ACDW2  = 000000              269
ACPUOP= 000000               269     284
ACRLOC  001400               396#
ADDW0  = 000000              269
ADDW1  = 000000              269
ADDW10= 000000               269
ADDW11= 000000               269
ADDW12= 000000               269
ADDW13= 000000               269
ADDW14= 000000               269
ADDW15= 000000               269
ADDW2  = 000000              269
ADDW3  = 000000              269
ADDW4  = 000000              269
ADDW5  = 000000              269
ADDW6  = 000000              269
ADDW7  = 000000              269
ADDW8  = 000000              269
ADDW9  = 000000              269
ADEVCT= 000000               269     275
ADEVM  = 000001              136#    269     311
AENV   = 000000              269     280
AENVM  = 000000              269     281
AFATAL= 000000               269     272
AMADR1= 000000               269     297
AMADR2= 000000               269     301
AMADR3= 000000               269     304
AMADR4= 000000               269     307
AMAMS1= 000000               269     291
AMAMS2= 000000               269     299
AMAMS3= 000000               269     302
AMAMS4= 000000               269     305
AMSGAD= 000000               269     277
AMSGLG= 000000               269     278
AMSGTY= 000000               269     271
AMTYP1= 000000               269     292
AMTYP2= 000000               269     300
AMTYP3= 000000               269     303
AMTYP4= 000000               269     306
APASS  = 000000              269     274
APRIOR= 000000               269
APTCSU= 000040              2218    2323#
APTENV= 000001              2211    2279    2321#   2496
APTSIZ= 000200               437    2320#
APTSPO= 000100              2213    2281    2322#
ASWREG= 000000               269     282
ATESTN= 000000               269     273
AUNIT  = 000000              269     276
AUSWR  = 000000              269     283
AVECT1= 000000               269     308
AVECT2= 000000               269     309
BEGPAT  013110               707     729     752     775     874     932     987    1044    1275    1303    1731    1854    1910
                            2085#
BGCHP2  013152              2102#
```

```
BGCHP3  013212            1409    1521    2119#
BGCHP4  013254            1377    1489    1583    2136#
BGPAT1  013112            1670    2086#
BIT0  = 000001             120#    519     541     555     580     605     624     639     661     677     696     712     734
                           757     780     803     841     879     883     911     937     941     966     992     996    1024
                          1049    1053    1078    1671    1732    1733    1796    1797    1855    1856    1915    1916    1952
                          1953    1978    1979    2073
BIT00 = 000001             110#    120
BIT01 = 000002             109#    119
BIT02 = 000004             108#    118
BIT03 = 000010             107#    117
BIT04 = 000020             106#    116
BIT05 = 000040             105#    115
BIT06 = 000100             104#    114
BIT07 = 000200             103#    113
BIT08 = 000400             102#    112    2600
BIT09 = 001000             101#    111    2507    2608
BIT1  = 000002             119#    580
BIT10 = 002000             100#
BIT11 = 004000              99#   2615
BIT12 = 010000              98#
BIT13 = 020000              97#   2491
BIT14 = 040000              96#   2586
BIT15 = 100000              95#    137
BIT2  = 000004             118#
BIT3  = 000010             117#
BIT4  = 000020             116#
BIT5  = 000040             115#
BIT6  = 000100             114#    518     533     540     557     564     587     646     886    1006
BIT7  = 000200             113#    518     526     533     540     557     564     587     638     646     653     660     824
                           855     886     893     999    1006
BIT8  = 000400             112#    138
BIT9  = 001000             111#    139     557
BPTVEC= 000014             127#
CHPISR= 000140             155#
CIMR  = 000040             145#   1333    1355    1413    1625
CIRMR = 000020             142#   1556    1674    1740    1741    1803    1804    1862    1863    1920    1921
CIRR  = 000100             150#   1467    1525
CISR  = 000160             156#
CKSWR = 104407            2482    2483    2506    2521    2585    2860#
CLRCSR  013002             552     576     600     634     672     706     728     751     774     797     835     873     931
                           986    1043    1097    1130    1177    1223    1270    1298    1326    1348    1371    1403    1434
                          1458    1482    1514    1546    1576    1614    1651    1663    1725    1786    1837    1848    1909
                          1949    1975    2055#
CLRIRR  013052            1101    1134    1227    1438    1462    1487    1519    1550    1581    1620    1656    2070#
CR    = 000015              35#   2257    2267
CRLF  = 000200              36#   2228    2267
CSIMR = 000050             146#   1379
CSIRMR= 000030             143#   1584
CSIRR = 000110             151#   1491
CSISR = 000170             157#
DDISP = 177570              42#    251     425
DH1     016711             332     338     344     350     356     362     369    2894#
DH2     016756            2901#
DIR   = 000400             138#    518     604     638     676     886     944     999    1056
DISPLA  001142             251#    425*    433*   2486*   2630*
```

```
DISPRE  000174            180#   433
DMAP    001364            390#   470*   471*   474   2002*  2006
DRCSA   001342            377#   473*   477    496    512    553    577    636    709    876    936    990   1099
                         1131   1178   1224   1273   1301   1328   1350   1374   1406   1436   1460   1485   1517
                         1548   1579   1654   1664   1668*  1672*  1840   1912   1950   2000   2056   2071
DRCSB   001346            379#   601    674    731    878    934   1047   1100   1132   1179   1225   1274   1302
                         1329   1351   1375   1407   1437   1461   1486   1518   1549   1580   1655   1665   1841
                         1913   1951   2057
DRCSC   001352            381#   513    554    635    754    877    989   1048   1122   1169   1215   1261   1290
                         1318   1340   1363   1394   1426   1449   1474   1506   1538   1568   1606   1652   1669*
                         1673*  1716   1838   1976   2058
DRCSD   001356            383#   602    673    777    935    991   1046   1123   1170   1216   1262   1291   1319
                         1341   1364   1395   1427   1450   1475   1507   1539   1569   1607   1653   1717   1839
                         1914   1977   2059
DRDBA   001344            378#   710    908    971   1013
DRDBB   001350            380#   732    916    963   1067
DRDBC   001354            382#   755    900   1021   1083
DRDBD   001360            384#   480    778    955   1029   1075
DSWR  = 177570             41#   250    424
DT1     017034            333    339    345    351    357    363    370   2911#
DT2     017050           2913#
EDCHP1  013132           1708   2094#
EDCHP2  013170           2109#
EDCHP3  013232           1420   1532   2127#
EDCHP4  013274           1388   1500   1600   2144#
EMTVEC= 000030            130#   409*   410*
EM1     016557            331   2877#
EM2     016576            337   2880#
EM3     016620            343   2883#
EM4     016633            349   2885#
EM5     016646            355   2887#
EM6     016661            361   2889#
EM7     016674            368   2891#
ENDDAT  013410            721    743    766    789    923    978   1036   1090   1286   1314   1778   1900   1940
                         2184#
ENDPAT  013316           2153#
ERRVEC= 000004            123#   422    423*   434*   495*   506*  2591   2592*  2594*  2597*
GNS   = ****** U          179   2852   2853   2854   2855   2856  2858   2860   2861    362
GTSWR = 104406            468   2858#
HT    = 000011             33#  2226   2267
IE    = 001000            139#   558   1952
IMRLOC  001372            393#  1675*  1693   1711*
INTFLG  001366            391#
IOTVEC= 000020            128#   407*   408*
IRRLOC  001376            395#  1676*  1679   1710*
ISRLOC  001374            394#
LF    = 000012             34#  2261   2267
LMD04 = 000200            159#
LMD57 = 000240            160#
MACR  = 000254            166#  1115   1137   1162   1190   1248   1277   1626
MIMR  = 000244            164#  1104   1144   1183   1207   1236   1305   1332   1354   1378   1410   1561   1594
                         1632   1702
MIRR  = 000250            165#  1110   1156   1201   1230   1254   1441   1465   1490   1522   1553   1586   1644
                         1687   1744   1751   1823   1830   1866   1873   1924   1931
MISR  = 000240            163#  1150   1196   1242   1638
NEXPAS  002032            475    477#  2008
```

```
NEXPA1  002036            478#   481
NXDEV   012620           2000#
NXDEV1  012624            476   2001#  2007
PACR  = 000300            169#  1138   1278   1622
PATDAT  013320           2156#
PIMR  = 000260            170#  1184   1306
PIRQ  = 177772             40#
PIRQVE= 000240            134#
PR0   = 000000             57#
PR1   = 000040             58#
PR2   = 000100             59#
PR3   = 000140             60#
PR4   = 000200             61#
PR5   = 000240             62#
PR6   = 000300             63#
PR7   = 000340             64#   485*
PS    = 177776             37#    38
PSW   = 177776             38#
PVMA  = 000340            171#
PWRMSG  016350           2808   2815#
PWRVEC= 000024            129#   413*   414*  2777*  2778*  2787*  2793*  2805*  2806*
RDCHR = 104410           2747   2861#
RDLIN = 104411           2862#
RDY   = 100000            137#   518    533    587    604    617    638    653    676    689    886    944    999
                         1056
RESVEC= 000010            124#
SIMR  = 000060            147#  1356   1381   1555   1588
SIRR  = 000120            152#  1442   1466   1493   1554   1587   1624
SSIMR = 000070            148#  1411   1675
SSIRR = 000130            153#  1523   1676
STACK = 001100             28#   482
START   001402            183    398#
START1  001766            467    469#  2042   2810
STKLMT= 177774             39#
SWR     001140            250#   403    424*   426    432*   439*  2491   2503   2507   2586   2600   2602   2608
                         2615   2644   2681*  2785   2798*
SWRCK   014770           2493   2519#
SWREG   000176            181#   432   2644   2657
SW0   = 000001             92#
SW00  = 000001             82#    92
SW01  = 000002             81#    91
SW02  = 000004             80#    90
SW03  = 000010             79#    89
SW04  = 000020             78#    88
SW05  = 000040             77#    87
SW06  = 000100             76#    86
SW07  = 000200             75#    85
SW08  = 000400             74#    84
SW09  = 001000             73#    83
SW1   = 000002             91#
SW10  = 002000             72#
SW11  = 004000             71#
SW12  = 010000             70#
SW13  = 020000             69#
SW14  = 040000             68#
SW15  = 100000             67#
```

```
SW2    = 000004              90#
SW3    = 000010              89#
SW4    = 000020              88#
SW5    = 000040              87#
SW6    = 000100              86#
SW7    = 000200              85#
SW8    = 000400              84#
SW9    = 001000              83#
TBITVE= 000014              125#
TITLED  016462              462    2865#
TKVEC  = 000060             132#
TLCABL  016530              463    2872#
TPVEC  = 000064             133#
TRAPVE= 000034              131#    411*    412*
TRTVEC= 000014              126#
TSTNUM  001362              389#   2519*   2911    2913
TST1    002074              492#
TST10   003260              699     705#
TST11   003354              727#
TST12   003450              750#
TST13   003544              773#
TST14   003640              796#
TST15   004010              828     834#
TST16   004156              865     872#
TST17   004456              930#
TST2    002152              511#
TST20   004744              985#
TST21   005244             1042#
TST22   005532             1096#
TST23   005666             1121    1129#
TST24   006076             1168    1176#
TST25   006302             1214    1222#
TST26   006506             1260    1269#
TST27   006624             1289    1297#
TST3    002344              545     551#
TST30   006742             1317    1325#
TST31   007034             1339    1347#
TST32   007134             1362    1370#
TST33   007262             1393    1402#
TST34   007410             1425    1433#
TST35   007510             1448    1457#
TST36   007612             1473    1481#
TST37   007744             1505    1513#
TST4    002454              569     575#
TST40   010076             1537    1545#
TST41   010222             1567    1575#
TST42   010376             1605    1613#
TST43   010604             1650    1662#
TST44   011120             1715    1724#
TST45   011406             1785#
TST46   011666             1836    1847#
TST47   012154             1908#
TST5    002552              592     599#
TST50   012364             1947#
TST51   012502             1966    1973#
TST6    002720              627     633#
```

```
TST7    003112              665    671#
TYPDS = 104405             2031   2856#
TYPE  = 104401              462    463   2029  2032  2231  2384  2459  2494  2531  2548  2550  2553  2555
                           2559   2566   2655  2656  2659  2672  2683  2702  2751  2754  2758  2807  2852#
TYPOC = 104402             2539   2563   2658  2853#
TYPON = 104404             2855#
TYPOS = 104403             2854#
XXDP    001370              392#   449*   460
$APTHD  001000              210    216#
$ASTAT= ****** U           2301   2316
$ATYC   013720             2272   2274#
$ATY1   013674             2270#
$ATY3   013702             2216   2271#
$ATY4   013712             2273#  2499
$AUTOB  001134              247#   453*   458   466   2652  2773
$BASE   001244              310#   472
$BDADR  001122              242#   498*   516*  525*  532*  556*  578*  603*  610*  616*  637*  645*  652*
                            675*   682*   688*  711*  733*  756*  779*  804*  811*  817*  823*  842*  848*
                            854*   861*   880*  894*  900*  908*  916*  938*  950*  955*  963*  971*  993*
                           1007*  1013*  1021* 1029* 1050* 1062* 1067* 1075* 1083* 1102* 1135* 1181* 1228*
                           1276*  1304*  1330* 1352* 1376* 1408* 1439* 1463* 1488* 1520* 1551* 1582* 1621*
                           1677*  1685*  1692* 1700* 1743* 1750* 1807* 1813* 1822* 1829* 1865* 1872* 1923*
                           1930*  1954*  1963* 1980* 1988* 2911  2913
$BDDAT  001126              244#   483   2055  2911  2913
$CDW1   001250              312#
$CHARC  013670             2233*  2243*  2250  2259* 2264#
$CKSWR  015422             2644#  2860
$CMTAG  001100              230#   400    401   409   415   416
$GM3  = 000000              260#
$CNTLG  016147             2655   2768#
$CNTLU  016142             2672   2767#
$CPUOP  001216              284#
$CRLF   001165              263#   2232   2267  2494  2515  2531  2550  2555  2559  2683  2767
$DBLK   014604             2425   2459   2467#
$DEVCT  001200              275#   484*
$DEVM   001246              311#   470
$DOAGN  012756             2025   2034   2040#
$DTBL   014574             2428   2463#
$ENDAD  012746              193   2036#
$ENDCT  012714             2027#
$ENDMG  012765             2029   2044#
$ENULL  012762             2032   2043#
$ENV    001210              280#   451   2211  2279  2303  2496
$ENVM   001211              281#   437   2213  2218  2281
$EOP    012660             2003   2017#
$EOPCT  012706             2024#  2028
$ERFLG  001103              233#  2484*  2515  2575  2604  2606  2612*  2634
$ERMAX  001115              239#   417*  2606  2629*  2634
$ERROR  014614              409   2481#
$ERRPC  001116              240#  2488*  2489*  2490  2515  2537  2911  2913
$ERRTB  001252              329#  2545
$ERRTY  015006             2520   2530#
$ERTTL  001112              237#  2487*  2515
$ESCAP  001162              261#   416*  2510  2512  2515  2628*
$ETABL  001210              279#
$ETEND  001252              222    313#
```

```
$FATAL  001172          272#   441*  2307*
$FFLG   014140         2270*  2273*  2301  2310*  2318#
$FILLC  001156          258#  2236  2267
$FILLS  001155          257#  2267
$GDADR  001120          241#  2913
$GDDAT  001124          243#   493*   518*   522   526*   529   533*   537   540*   544   557*   561   564*
                        568    579*   584   587*   591   604*   607   611*   613   617*   620   623*   626
                        638*   642*   646*   649   653*   657   660*   664   676*   679   683*   685   689*
                        692    695*   698   713*   717   735*   739   758*   762   781*   785   805*   808
                        812*   814    818*  820   824*   827   843*   846   849*   851   855*   858   862*
                        864    886*   890   893*  897   901*   903   906*   909   913   917*   919   944*
                        946    949*   952   956*  958   961*   964   968    972*  974   999*  1003  1006*
                       1010   1014*  1016  1019*  1022  1026  1030*  1032  1056*  1058  1061*  1064  1068*
                       1070   1073*  1076  1080   1084*  1086  1103*  1106  1109*  1112  1117   1136*  1139
                       1141   1145*  1147  1151*  1153  1158  1164   1182*  1185   1187  1191*  1193  1198
                       1203   1208*  1210  1229*  1233  1237*  1239  1243*  1245  1250  1256   1279*  1282
                       1307*  1310   1331*  1335  1353*  1358  1380*  1384   1412*  1416  1440*  1444  1464*
                       1469   1492*  1496  1524*  1528  1552*  1558  1563   1585*  1591  1596   1627*  1629
                       1633*  1635   1639*  1641  1646  1678*  1681  1684*  1686*  1689  1694*  1697  1701*
                       1704   1745*  1747  1752*  1753*  1754*  1756  1760*  1762  1766*  1768   1772*  1774
                       1806*  1810   1816  1824*  1826  1832  1867*  1869   1874*  1875*  1876*  1878  1882*
                       1884   1888*  1890  1894*  1896  1925*  1927  1932*  1933*  1934*  1936   1955*  1959
                       1962*  1965   1981*  1985  1989*  1991  2060*  2911   2913
$GET42  012736          2033#
$GTSWR  015472         2656#  2858
$HD   = 000003            11    12
$HIBTS  001000          217#
$ICNT   001104          234#  2619*  2620  2622*  2633
$ILLUP  016342         2777   2793   2812#
$INTAG  001135          248#  2684   2773
$ITEMB  001114          238#  2490*  2498  2515   2534
$LF     001166          264#  2267   2515  2758   2767
$LFLG   014137         2311*  2317#
$LPADR  001106          235#   418*  2610*  2626*  2631  2633
$LPERR  001110          236#   419*   708*   730*   753*   776*   875*   933*   988*  1045*  1272*  1300*  1373*
                       1405*  1484*  1516*  1578*  1730*  1853*  1911*  2509   2610   2627*  2633
$MADR1  001222          297#
$MADR2  001226          301#
$MADR3  001232          304#
$MADR4  001236          307#
$MAIL   001170          218    222    270#   436   2211   2496   2625
$MAMS1  001220          291#
$MAMS2  001224          299#
$MAMS3  001230          302#
$MAMS4  001234          305#
$MBADR  001002          218#
$MFLG   014136         2271*  2277   2312*  2316#
$MNEW   016165         2659   2771#
$MSGAD  001204          277#  2287*  2290
$MSGLG  001206          278#  2292*
$MSGTY  001170          271#   442*  2285   2293*  2305   2309*
$MSWR   016154         2656   2769#
$MTYP1  001221          292#
$MTYP2  001225          300#
$MTYP3  001231          303#
$MTYP4  001235          306#
```

```
$MXCNT  015420          2623    2633#
$NULL   001154           256#   2238    2267
$NWTST= 000001           489#    508#    548#    572#    596#    630#    668#    702#    724#    747#    770#    793#    831#
                         869#    927#    982#   1039#   1093#   1126#   1173#   1219#   1266#   1294#   1322#   1344#   1367#
                        1399#   1430#   1454#   1478#   1510#   1542#   1572#   1610#   1659#   1721#   1782#   1844#   1905#
                        1944#   1970#
$OCNT   014364          2356*   2385*   2398#
$OMODE  014366          2351*   2355*   2360    2363*   2374*   2400#
$OVER   015404          2587    2603    2611    2621    2630#
$PASS   001176           274#    436*   2021*   2022*   2030    2043    2617    2634
$PASTM  001006           220#
$PWRAD  016336          2810#
$PWRDN  016176           413    2777#   2805
$PWRMG  016332          2808#
$PWRUP  016250          2787    2793#
$QUES   001164           262#   2267    2515    2702    2751    2767
$RDCHR  015704          2715#   2861
$RDDEC= ****** U        2863
$RDLIN  016024          2743#   2862
$RDOCT= ****** U        2863
$RDSZ = 000010          2736#
$RTNAD  012760          2042#
$R2A  = ****** U        2863
$SAVRE= ****** U        2863
$SAVR6  016346          2786*   2794    2795*   2796*   2814#
$SCOPE  015142           407    2584#
$SETUP= 000117           397#    406     407     409     411     413     415     416     418    2019    2482    2506    2514
                        2585    2639    2773
$STUP = 177777           397#
$SVLAD  015350          2595    2624#
$SVPC = 000106           191#    196
$SWR  = 165400            11     12#     13#     19      20      21      22      23      24     260     261     262     415
                         416     418     419     493     512     552     576     600     634     672     706     728     751
                         774     797     835     873     931     986    1043    1097    1130    1177    1223    1270    1298
                        1326    1348    1371    1403    1434    1458    1482    1514    1546    1576    1614    1663    1725
                        1786    1848    1909    1948    1974    2014    2020    2035    2041    2043    2474    2475    2476
                        2477    2487    2491    2503    2507    2515    2576    2577    2578    2579    2580    2586    2598
                        2600    2601    2604    2605    2606    2613    2614    2615    2627    2630    2633    2811
$SWREG  001212           282#    439
$SWRMK= 000000            24      25    2580    2581    2602
$TESTN  001174           273#    443*   2625*
$TIMES  001160           260#    415*   1948*   1974*   2020*   2613*   2620    2623*   2633
$TKB    001146           253#   2637    2648    2665    2719    2725
$TKS    001144           252#   2637    2646    2662    2686*   2717    2723
$TN   = 000052            11#     14#     489     493#    508     512#    545     548     552#    569     572     576#    592
                         596     600#    627     630     634#    665     668     672#    699     702     706#    724     728#
                         747     751#    770     774#    793     797#    828     831     835#    865     869     873#    927
                         931#    982     986#   1039    1043#   1093    1097#   1121    1126    1130#   1168    1173    1177#
                        1214    1219    1223#   1260    1266    1270#   1289    1294    1298#   1317    1322    1326#   1339
                        1344    1348#   1362    1367    1371#   1393    1399    1403#   1425    1430    1434#   1448    1454
                        1458#   1473    1478    1482#   1505    1510    1514#   1537    1542    1546#   1567    1572    1576#
                        1605    1610    1614#   1650    1659    1663#   1715    1721    1725#   1782    1786#   1836    1844
                        1848#   1905    1909#   1944    1948#   1966    1970    1974#
$TPB    001152           255#   2256*   2267
$TPFLG  001157           259#   2205    2267
$TPS    001150           254#   2254    2267
```

```
$TRAP   016402            411    2829#
$TRAP2  016424           2840#   2851
$TRP  = 000012           2844#   2853#  2854#  2855#  2856#  2857#  2858   2859#  2860   2861#  2862#  2863#
$TRPAD  016436           2834    2851#
$TSTM   001004            219#
$TSTNM  001102            232#   2019*  2486   2515   2519   2575   2602   2624*  2625   2630   2634
$TTYIN  016132           2744    2745   2762   2766#
$TYPBN= ****** U         2857
$TYPDS  014370           2413#   2856
$TYPE   013412           2205#   2298   2844   2852
$TYPEC  013624           2235    2242   2249   2254#  2255   2688
$TYPEX  013672           2260    2262   2265#
$TYPOC  014166           2354#   2853
$TYPON  014202           2353    2356#  2855
$TYPOS  014142           2349#   2854
$UNIT   001202            276#    469*   484    2005*
$UNITM  001010            221#
$USWR   001214            283#
$VECT1  001240            308#
$VECT2  001242            309#
$XTSTR  015154           2589#
$$GET4= 000000           2035#
$OFILL  014365           2350*   2354*  2364   2399#
$40CAT= ****** U         2493    2586
.     = 017066            175#    179#   184#   191    192#   194#   196#   197#   206    207#   209#   211#   229#
                          265     404    418    419   2043   2047   2267   2319#  2467#  2515   2569#  2633   2634
                         2637    2766#  2767   2773   2789   2813   2910#
.$ASTA= ****** U         2271    2274
.$X   = 001000            206#    211
```

```
COMMEN      1#    135#
ENDCOM      1#    135#
ERROR      29#    505    524    531    539    546    563    570    586    593    609    615    622    628    644
          651    659    666    681    687    694    700    719    741    764    787    810    816    822    829
          853    860    866    892    899    905    915    921    948    954    960    970    976   1005   1012
         1018   1028   1034   1060   1066   1072   1082   1088   1108   1114   1119   1143   1149   1155   1160
         1166   1189   1195   1200   1205   1212   1235   1241   1247   1252   1258   1284   1312   1337   1360
         1386   1418   1446   1471   1498   1530   1560   1565   1593   1598   1631   1637   1643   1648   1683
         1691   1699   1706   1749   1758   1764   1770   1776   1812   1818   1828   1834   1871   1880   1886
         1892   1898   1929   1938   1961   1967   1987   1993
ESCAPE      1#    135#
GETPRI      1#    135#
GETSWR      1#    135#
MULT        1#    135#
NEWTST      1#    135#    489    508    548    572    596    630    668    702    724    747    770    793    831
          869    927    982   1039   1093   1126   1173   1219   1266   1294   1322   1344   1367   1399   1430
         1454   1478   1510   1542   1572   1610   1659   1721   1782   1844   1905   1944   1970
POP         1#    135#   2313   2314   2454   2798   2799
PUSH        1#    135#   2274   2276   2297   2413   2779   2785
REPORT      1#    135#
SCOPE      30#    492    511    551    575    599    633    671    705    727    750    773    796    834    872
          930    985   1042   1096   1129   1176   1222   1269   1297   1325   1347   1370   1402   1433   1457
         1481   1513   1545   1575   1613   1662   1724   1785   1847   1908   1947   1973
SETPRI      1#    135#
SETTRA   2844#   2853   2854   2855   2856   2858   2860   2861   2862
SETUP       1#    135#    398
SKIP        1#    135#    538    545    569    592    621    627    658    665    693    699    828    865   1121
         1168   1214   1260   1289   1317   1339   1362   1393   1425   1448   1473   1505   1537   1567   1605
         1650   1715   1836   1966
SLASH       1#    135#
SPACE     135#
STARS       1#    135#    189    203    205    212    225    265    268    489    491    508    510    548    550
          572    574    596    598    630    632    668    670    702    704    724    726    747    749    770
          772    793    795    831    833    869    871    927    929    982    984   1039   1041   1093   1095
         1126   1128   1173   1175   1219   1221   1266   1268   1294   1296   1322   1324   1344   1346   1367
         1369   1399   1401   1430   1432   1454   1456   1478   1480   1510   1512   1542   1544   1572   1574
         1610   1612   1659   1661   1721   1723   1782   1784   1844   1846   1905   1907   1944   1946   1970
         1972   2011   2050   2054   2190   2269   2326   2403   2470   2515   2518   2525   2572   2636   2639
         2707   2736   2775   2791   2823
SWRSU       1#    135#    420#
TRMTRP   2844#
TYPBIN      1#    135#
TYPDEC      1#    135#   2030
TYPNAM      1#    135#
TYPNUM      1#    135#
TYPOCS      1#    135#
TYPOCT      1#    135#   2537   2561   2657
TYPTXT      1#    135#
$$CMRE    223#
$$CMTM    223#
$$ESCA      1#    135#
$$NEWT      1#    135#    489    508    548    572    596    630    668    702    724    747    770    793    831
          869    927    982   1039   1093   1126   1173   1219   1266   1294   1322   1344   1367   1399   1430
         1454   1478   1510   1542   1572   1610   1659   1721   1782   1844   1905   1944   1970
$$SET    2844#   2853   2854   2855   2856   2858   2860   2861   2862
$$SETM    436#
```

```
$$SKIP      1#     135#    545     569     592     627     665     699     828     865    1121    1168    1214    1260    1289
         1317     1339    1362    1393    1425    1448    1473    1505    1537    1567    1605    1650    1715    1836    1966
.EQUAT      1#      25
.HEADE      1#
.KT11       1#
.SETUP      1#     397
.SWRHI      1#      15
.SWRLO     25#
.$ACT1      1#     187
.$APTB      1#     266#
.$APTH      1#     201
.$APTY      1#    2267
.$ASTA      1#
.$CATC      1#     173
.$CMTA      1#     223
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#    2009
.$ERRO      1#    2468
.$ERRT      1#    2523
.$MULT      1#
.$POWE      1#    2773
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#    2634
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    2570
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    2821
.$TYPB      1#
.$TYPD      1#    2401
.$TYPE      1#    2188
.$TYPO      1#    2324
.$40CA      1#
.1170       1#


. ABS.  017066      000

 ERRORS DETECTED:  0

 DSKZ:CVDRCA,DSKZ:CVDRCA/CRF/SOL=DSKZ:SYSMAC.SML,DSKZ:CVDRCA.P11
 RUN-TIME: 51 52 3 SECONDS
 RUN-TIME RATIO: 331/108=3.0
 CORE USED:  33K  (65 PAGES)
```