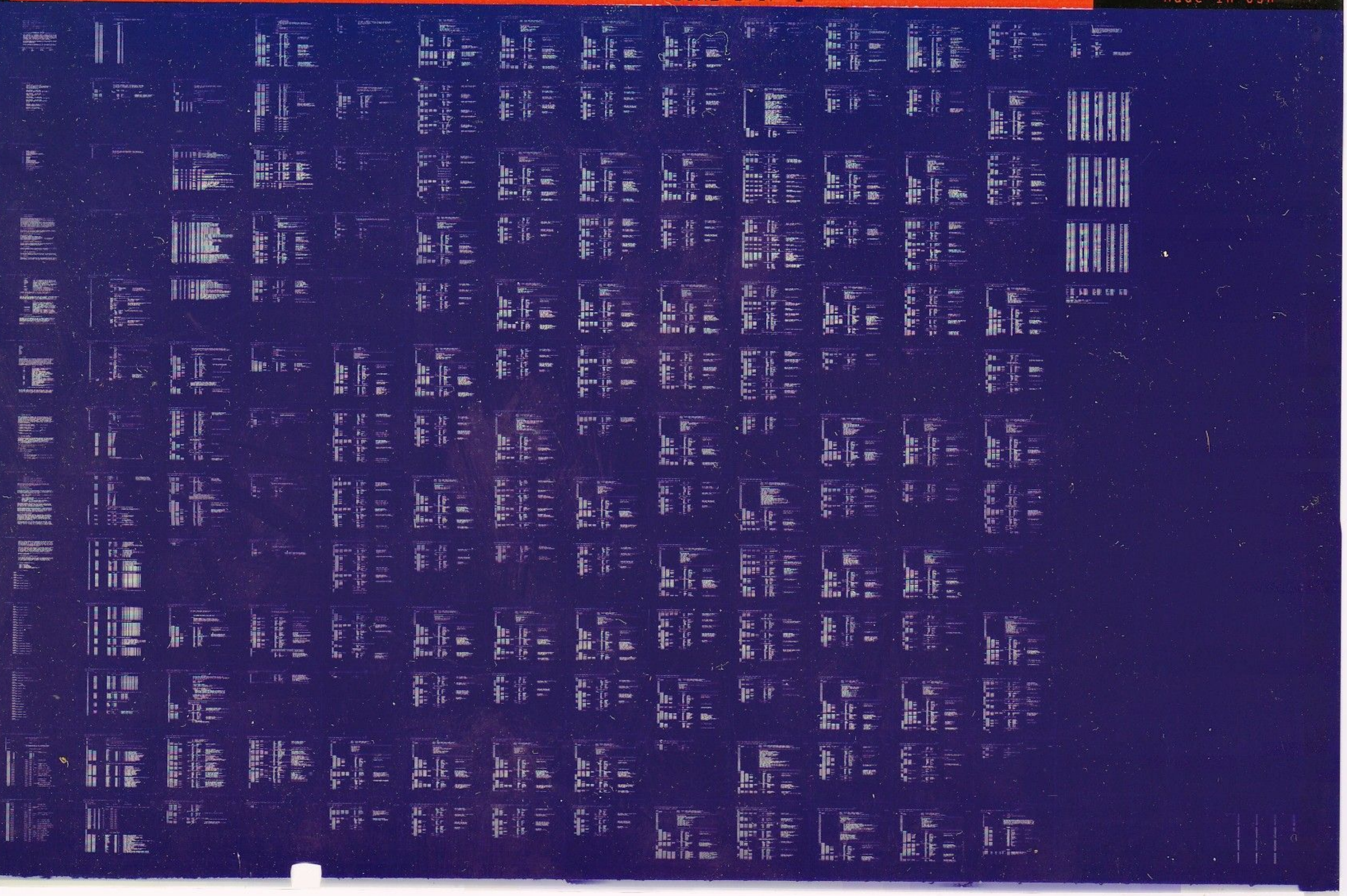
FEPCM

CTRL BRIDGE INTFC
CZFPAAQ

AH-T859A-MC
FICHE 1 OF 1

APR 1984
digital
Made In USA



.REM &

IDENTIFICATION

PRODUCT CODE: AC T858A-MC

PRODUCT NAME: CZFPAAO CTRL BRIDGE INTFC

PRODUCT DATE: MAY 1982

MAINTAINER: CSS GNG DIAGNOSTIC ENGINEERING

AUTHOR: DALE F. PROCTOR

REVISION HISTORY

A.0 DALE PROCTOR 22-DEC 1983
CHANGED THE NAME TO CZFPA AO AND RELEASED TO SDC. ALSO UPDATED ALL THE DOCUMENTATION, AND REMOVED THE HELP FILE QUESTIONS FROM THE INITIALIZATION SECTION.

D.2 DALE PROCTOR 4-NOV-1983
CHANGED THE INSTRUCTION MOV $PATCH,@Q8A TO MOV @$PATCH,@Q8A IN Z0791D5.MAC IN TEST 41.

D.1 DALE PROCTOR 30-SEP-1983
ADDED TO THE CLEANUP CODE TO ALWAYS DISABLE MIODIS.

D.0 DALE PROCTOR 15-JUNE-1983
RENAMED PROGRAM TO Z0791DX.MAC AND ARCHEVED.

C.1 DALE PROCTOR 10-JUNE 1983
ADDED STATST AND CKPNT MACROS TO PRINT OUT THE TEST SBTTL IF THE FLA:PNT FLAG IS SET.

C.0 MODIFIER- JOHN M. MARTIN 31-MAR-83
DISABLED UMR IN INITIALIZE.
CORRECTED ERRORS IN TESTS 3,36.

B.0 MODIFIER- PAUL HOLSINGER
ADDED 11/24 (UNIBUS) COMPATIBILITY.

A.0 INITIAL RELEASE

TABLE OF CONTENTS

1.0 GENERAL INFORMATION
1.1 PROGRAM ABSTRACT
1.2 SYSTEM REQUIREMENTS
1.3 RELATED DOCUMENTS AND STANDARDS
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
1.5 ASSUMPTIONS

2.0 OPERATING INSTRUCTIONS
2.1 COMMANDS
2.2 SWITCHES
2.3 FLAGS
2.4 HARDWARE QUESTIONS
2.5 SOFTWARE QUESTIONS
2.6 EXTENDED P-TABLE DIALOGUE
2.7 QUICK STARTUP PROCEDURE

3.0 ERROR INFORMATION

4.0 PERFORMANCE AND PROGRESS REPORTS

5.0 DEVICE INFORMATION TABLES

6.0 TEST SUMMARIES

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC CONTAINS CONTAINS REPAIR LEVEL TESTS FOR THE FRONT END PROCESSOR BRIDGE MODULE ON THE PDP-11 SIDE. IT CAN BE RUN EITHER WITH INTERNAL LOOPBACK LOGIC OR WITH A LOOPBACK CABLE INSTALLED.
THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

PDP-11/23 OR /24 CPU WITH AT LEAST 16K WORDS OF MEMORY, CONSOLE DEVICE AND TU-58 TAPE DRIVE. NORMAL CONFIGURATION FOR THE CONSOLE DEVICE IS THROUGH THE VAX SIDE OF THE FEP.

1.3 RELATED DOCUMENTS AND STANDARDS

FEPCM TECHNICAL MANUAL
AC-T860A-MC CZFPBA0 CTRL BRIDGE LINK
AC-T866AS-MC CXFPCA0 CTRL BRIDGE DEC/X11
EVDTE COMMAND BRIDGE INTERFACE DIAGNOSTIC. (VAX DIAGNOSTIC)
EVDTC COMMAND BRIDGE LINK DIAGNOSTIC. (VAX DIAGNOSTIC)
XXDP+ USERS MANUAL - CHQUS??.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE CPU, MEMORY, CONSOLE DEVICE AND THE TU-58 TAPE DRIVE ARE ASSUMED TO BE FUNCTIONING PRIOR TO RUNNING THIS TEST. NO RUNNING ORDER IS REQUIRED ON THE TESTS.

1.5 ASSUMPTIONS

EITHER A LOOPBACK CABLE IS INSTALLED OR THE DEFAULT IN THE HARDWARE QUESTION REFERING TO THE LOOPBACK CABLE IS ANSWERED.

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ↑C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

```
START/TESTS:1-5/PASS:1000/EOP:100
```

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBE*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP* USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

LSI BRIDGE MODULE BASE ADDRESS DEFAULT=160500
  ADDRESS OF Q-COMMAND REGISTER

LSI BRIDGE MODULE VECTOR DEFAULT=500
  ADDRESS OF INTERRUPT VECTOR

LOOPBACK CABLE INSTALLED DEFAULT=NO
  INTERNAL LOOPBACK OR PHYSICAL CABLE USED

BR LEVEL DEFAULT=4
  INTERRUPT PRIORITY LEVEL

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

THERE ARE NO SOFTWARE QUESTIONS IN THIS DIAGNOSTIC.

2.6 EXTENDED P-TABLE DIALOGUE

NOT APPLICABLE.

2.7 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM

4. TYPE "START"

5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"

6. ANSWER ALL THE HARDWARE QUESTIONS

7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME  TYPE  NUMBER  ON UNIT NUMBER  TST NUMBER PC:XXXXXX
ERROR MESSAGE
```

```
.WHERE; NAME = DIAGNOSTIC NAME
        TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
        NUMBER = ERROR NUMBER
        UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
        TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
        PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL
```

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBE" OR "IXE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

MOST OF THE ERROR MESSAGES ARE SELF EXPLAINATORY. THE GENERAL FORMAT IS THE BASIC ERROR LINE, FOLLOWED BY A DESCRIPTIVE LINE, FOLLOWED BY A LINE THAT CONTAINS A REGISTER NAME, IT'S ADDRESS, IT'S CONTENTS, AND WHAT WAS EXPECTED. UNDETERMINED BITS AND UNTESTED BITS ARE ZEROED OUT PRIOR TO TESTING AND DISPLAYING. IF THERE IS A BIT RANGE (EX., BIT 1-5 ONLY ARE TESTED,) THAT IS INCLUDED IN THE ERROR MESSAGE.

THREE OF THE TESTS, 28 - 30 DEAL WITH THE THE VIEW RAM. THERE IS A POSSIBILITY OF 1K OR 2K OF RAM INSALLED. IF THERE IS AN ERROR IN THIS RAM, YOU WILL GET A MAXIMUM OF 20 LINES OF ERROR MESSAGES. THIS IS TO PREVENT 1K OR 2K LINES OF MESSAGES PRODUCED.

HOWEVER, A TOTAL ERROR COUNT IS PROVIDED AT THE END OF EACH TEST. ALSO, TEST 29 AND 30 TELL THE OPERATOR THE RAM SIZE. THE THREE POSSIBILITIES ARE 1K, 2K OR UNDETERMINED RAM SIZE. THE 1K OR 2K RAM SIZE MESSAGE WILL OCCUR ONLY ONCE FROM THE POINT OF RUNNING THE DIAGNOSTIC, BUT UNDETERMINED RAM SIZE WILL PRINT EACH TIME TEST 29 OR 30 IS ENCOUNTERED IF NEEDED.

THERE IS A WATCH DOG TIMER CIRCUIT ON THE FEPCM, AND THE TESTING OF IT USES THE BELL ON THE CONSOLE. TEST 36 TESTS EACH TIME PATTERN 5 TIMES, AND WITH 4 TIMES PATTERNS THAT MEANS 20 BELLS ON THE CONSOLE. THE BELL'S RINGS SHOULD BE CLOSE TO THE TIMING OF THE WDT CIRCUIT. A MAXIMUM TIME OF 4 SECONDS IS PROVIDED BY THE CIRCUIT, BUT THE TEST ALLOWS APPROXIMATELY 8 SECONDS.

SEVERAL OF THE TESTS TEST THE INTERRUPTS. THEIR ERROR MESSAGES ARE SELF EXPLAINATORY.

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

5.0 DEVICE INFORMATION TABLES

P-TABLE DEFINITION-
- WORD 1- CSR ADDRESS
- WORD 2- VECTOR ADDRESS
- WORD 3- LOOKBACK CABLE INSTALLED (Y/N)
- WORD 4- BR INTERRUPT PRIORITY LEVEL

6.0 TEST SUMMARIES

TEST1:
RESET

TEST2:
CHECK QMT1 REGISTER

TEST3:
CHECK QCLR RESET

TEST4:
CHECK QCMD LO BITS (MOVING 1)

TEST5:
CHECK QCMD LOW BITS (MOVING 0)

TEST6:
CHECK QCMD HI BITS (MOVING 1)

TEST7:
CHECK QCMD HI BITS (MOVING 0)

```
TEST8:
CHECK QVNT LO BITS (MOVING 1)

TEST9:
CHECK QVNT LOW BITS (MOVING 0)

TEST10:
CHECK QVNT HI BITS (MOVING 1)

TEST11:
CHECK QVNT HI BITS (MOVING 0)

TEST12:
CHECK ACK - PRS LINK

TEST13:
CHECK QBA (MOVING 1)

TEST14:
CHECK QBA (MOVING 0)

TEST15:
CHECK QDMA HI BITS (8-13) (MOVING 1)

TEST16:
CHECK QDMA HI BITS (8-13) (MOVING 0)

TEST17:
CHECK QDMA UXAD BITS

TEST18:
CHECK QUBA (MOVING 1)

TEST19:
CHECK QUBA (MOVING 0)

TEST20:
CHECK QWC (MOVING 1)

TEST21:
CHECK QWC (MOVING 0)

TEST22:
CHECK QVAD (Q-REGISTER) (MOVING 1)

TEST23:
CHECK QVAD (Q-REGISTER) (MOVING 0)

TEST24:
CHECK QVAD (U-REGISTER) (MOVING 1)

TEST25:
CHECK QVAD (U-REGISTER) (MOVING 0)

TEST26:
CHECK QVAD INCREMENT (Q-REGISTER)
```

TEST27:
CHECK QVAD INCREMENT (U-REGISTER)

TEST28:
CHECK VIEW REGISTER RAM

TEST29:
CHECK RAM ADDRESSING (ADDRESS VALUE)

TEST30:
CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

TEST31:
CHECK QBA INCREMENT

TEST32:
CHECK QWC OVERFLOW (READ)

TEST33:
CHECK QWC OVERFLOW (WRITE)

TEST34:
DMA 1 WORD (WRITE)

TEST35:
DMA 1 WORD (READ)

TEST36:
DMA 1 WORD (WRITE)

TEST37:
CHECK WDTO TIME

TEST38:
CHECK WDTO RESET

TEST39:
CHECK VNTACK INTERRUPT

TEST40:
CHECK CMDPRS INTERRUPT

TEST41:
CHECK RDY INTERRUPT

TEST42:
CHECK ERR INTERRUPT (UNEX)

TEST43:
CHECK ERR INTERRUPT (QNEX)

TEST44:
CHECK ERR INTERRUPT (UPWRFL)

TEST45:
CHECK BR LEVEL &

```
                        .TITLE CZFPA-A0 CTRL BRIDGE INTFC DIAGNOSTIC
                        .SBTTL  PROGRAM HEADER

000000                          .ENABL  ABS,AMA
        002000                  .       =       2000

002000                          BGNMOD

                        ;++
                        ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                        ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                        ;--

002000                          POINTER BGNSETUP,BGNRPT


002000                          HEADER  CZFPA,A,0,8.,1
002000                  L$NAME::                ;DIAGNOSTIC NAME
002000      103                 .ASCII /C/
002001      132                 .ASCII /Z/
002002      106                 .ASCII /F/
002003      120                 .ASCII /P/
002004      101                 .ASCII /A/
002005      000                 .BYTE   0
002006      000                 .BYTE   0
002007      000                 .BYTE   0
002010                  L$REV::                 ;REVISION LEVEL
002010      101                 .ASCII  /A/
002011                  L$DEPO::                ;0
002011      060                 .ASCII  /0/
002012                  L$UNIT::                ;NUMBER OF UNITS
002012  000001                  .WORD   T$PTHV
002014                  L$TIML::                ;LONGEST TEST TIME
002014  000010                  .WORD   8.
002016                  L$HPCP::                ;POINTER TO H.W. QUES.
002016  041444                  .WORD   L$HARD
002020                  L$SPCP::                ;POINTER TO S.W. QUES.
002020  000000                  .WORD   0
002022                  L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002260                  .WORD   L$HW
002024                  L$SPTP::                ;PTR. TO S.W. PTABLE
002024  000000                  .WORD   0
002026                  L$LADP::                ;DIAG. END ADDRESS
002026  042064                  .WORD   L$LAST
002030                  L$STA::                 ;RESERVED FOR APT STATS
002030  000000                  .WORD   0
002032                  L$CO::
002032  000000                  .WORD   0
002034                  L$DTYP::                ;DIAGNOSTIC TYPE
002034  000001                  .WORD   1
002036                  L$APT::                 ;APT EXPANSION
002036  000000                  .WORD   0
002040                  L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  002124                  .WORD   L$DISPATCH
002042                  L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                  .WORD   0
002044                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
```

PROGRAM HEADER

```
002044  000000                  .WORD   0
002046                  L$EXP1::                ;EXPANSION WORD
002046  000000                  .WORD   0
002050                  L$MREV::                ;SVC REV AND EDIT #
002050     003                  .BYTE   C$REVISION
002051     003                  .BYTE   C$EDIT
002052                  L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  002554                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  014446                  .WORD   L$RPT
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  000000                  .WORD   0
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  000000                  .WORD   0
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  002560                  .WORD   L$DESC
002100                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                  L$ETP::                 ;POINTER TO ERRTBL
002102  000000                  .WORD   0
002104                  L$ICP::                 ;PTR. TO INIT CODE
002104  014466                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  015044                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  015040                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  014460                  .WORD   L$PROT
002114                  L$TEST::                ;TEST NUMBER
002114  000000                  .WORD   0
002116                  L$DLY::                 ;DELAY COUNT
002116  000000                  .WORD   0
002120                  L$HIME::                ;PTR. TO HIGH MEM
002120  000000                  .WORD   0
75
```

DISPATCH TABLE

```
                                .SBTTL  DISPATCH TABLE

                                ;++
                                ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
                                ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
                                ;--

                                        DISPATCH 45
002122  000055                         .WORD   45
002124                          L$DISPATCH::
002124  015152                          .WORD   T1
002126  016050                          .WORD   T2
002130  016302                          .WORD   T3
002132  017012                          .WORD   T4
002134  017320                          .WORD   T5
002136  017644                          .WORD   T6
002140  020140                          .WORD   T7
002142  020456                          .WORD   T8
002144  020760                          .WORD   T9
002146  021304                          .WORD   T10
002150  021600                          .WORD   T11
002152  022104                          .WORD   T12
002154  022502                          .WORD   T13
002156  022756                          .WORD   T14
002160  023266                          .WORD   T15
002162  023616                          .WORD   T16
002164  024174                          .WORD   T17
002166  024576                          .WORD   T18
002170  025062                          .WORD   T19
002172  025360                          .WORD   T20
002174  025634                          .WORD   T21
002176  026142                          .WORD   T22
002200  026444                          .WORD   T23
002202  027006                          .WORD   T24
002204  027322                          .WORD   T25
002206  027652                          .WORD   T26
002210  030064                          .WORD   T27
002212  030334                          .WORD   T28
002214  031570                          .WORD   T29
002216  032452                          .WORD   T30
002220  033306                          .WORD   T31
002222  033620                          .WORD   T32
002224  034062                          .WORD   T33
002226  034362                          .WORD   T34
002230  034674                          .WORD   T35
002232  035226                          .WORD   T36
002234  035702                          .WORD   T37
002236  036420                          .WORD   T38
002240  036646                          .WORD   T39
002242  037126                          .WORD   T40
002244  037404                          .WORD   T41
002246  037704                          .WORD   T42
002250  040260                          .WORD   T43
002252  040620                          .WORD   T44
002254  041152                          .WORD   T45
```

DEFAULT HARDWARE P TABLE

```
                                .SBTTL  DEFAULT HARDWARE P TABLE

                                ;**
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
                                ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
                                ; -

002256                                  BGNHW   DFPTBL
002256  000004                         .WORD   L10000-L$HW/2
002260                          L$HW::
002260                          DFPTBL::

002260  160500                         .WORD   QCMDAD                  ;HARDWARE UNIT 0 DEFAULT BASE ADDRESS
002262  000500                         .WORD   VEC                     ;HARDWARE UNIT 0 DEFAULT VECTOR
002264  000000                         .WORD   NO                      ;LOOPBACK CABLE INSTALLED DEFAULT
002266  000200                         .WORD   ^B0000000010000000      ;DEFAULT BR INTERRUPT LEVEL 4

002270                                  ENDHW
002270                          L10000:
```

SOFTWARE P TABLE

```
                                .SBTTL  SOFTWARE P-TABLE

                                ;++
                                ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                                ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                                ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                                ; AT RUN TIME.
                                ;--

                                ;SOFTWARE TABLE NOT USED


002270                                  ENDMOD
```

SOFTWARE P TABLE

```
                                        .ENABL  AMA
                                .SBTTL  GLOBAL EQUATES SECTION

002270                                  BGNMOD
```

MACRO DEFINITIONS

```
                        .SBTTL MACRO DEFINITIONS
                        ;*****************************************************************************
                        ;**
                        ;                       MACRO DEFINITION SECTION
                        ;--
                        ;*****************************************************************************

                        .MACRO  INIT
                        .LIST
                        ;INITIALIZE DEVICE
                                MOV     @#MQHALT,@QMT1          ;INITIALIZE Q-BRIDGE
                                MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
                                CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
                                CLR     ERROR1                  ;CLEAR ERROR FLAG
                        .NLIST
                        .ENDM
                        ;
                        ;       MACROS TO CALL BITS SUBROUTINE
                        ;
                        ;       MACRO SETBITS
                        ;       USED TO CALL ROUTINE TO PRINT OUT REGISTER BIT NAMES
                        ;
                        .MACRO  SETBITS NAME,POINTER,VALUE
                        .NARG   TE$                     ;GET NUMBER OF ARGUMENTS PASSED
                        .IF NE  TE$-3                   ;SEE IF RIGHT NUMBER
                        .ERROR  ;SETBITS MACRO REQUIRES 3 PARAMETERS
                        .ENDC
                        .NCHR   $L1 NAME                ;GET NUMBER OF CHARACTERS IN NAME PARAMETER
                        $L1 = $L1+1                     ;ALLOW FOR ZERO BYTE IN ASCIZ
                        $TEMP = $L1 & 1                 ;CHECK FOR ODD OR EVEN
                        .IF EQ $TEMP                    ;IF EVEN LENGTH OF NAME
                        $LEN = 4+$L1
                        .ENDC
                        .IF NE $TEMP                    ;IF ODD LENGTH OF NME
                        $LEN = 5+$L1
                        .ENDC
                                MOV     VALUE,R1        ;GET VALUE REGISTER TO READ
                                JSR     R5,BITS$        ;SUBROUTINE TO OUTPUT VALUES
                                .WORD   $LEN
                                .WORD   POINTER
                                .ASCIZ  /NAME/
                                .EVEN
                        .ENDM
                        ;
                        ;       MACRO S$BITS
                        ;       USED TO CALL ROUTINE TO FORMAT PRINT BUFFER OF REGISTER BIT NAMES
                        ;
                        .MACRO  S$BITS  POINTER,VALUE
                        .NARG   TE$                     ;GET NUMBER OF ARGUMENTS PASSED
                        .IF NE  TE$-2                   ;SEE IF RIGHT NUMBER
                        .ERROR  ;S$BITS MACRO REQUIRES 2 PARAMETERS
                        .ENDC
                                MOV     VALUE,R1        ;GET VALUE REGISTER TO READ
                                JSR     R5,BIT$         ;SUBROUTINE TO OUTPUT VALUES
                                .WORD   POINTER
                                .EVEN
                        .ENDM
```

MACRO DEFINITIONS

```
                                ;
                                ;       MACROS USED TO PRINT OUT THE TEST'S TITLE IF THE PNT FLAG IS SET
                                ;
                                .MACRO  STATST  P1
                                .MCALL  PASTST
                                        .IF EQ F$
                                        .ERROR  ;2 STATST MACRO'S WITHOUT CKPNT MACRO CALL IN BETWEEN
                                        .MEXIT
                                        .ENDC
                                        F$ = 0
                                        .RADIX  10
                                        T$NUM = T$NUM+1
                                        PASTST  \T$NUM,<P1>
                                        .RADIX  8
                                .ENDM   STATST
                                .MACRO  PASTST  Z,P2
                                        .LIST
                                T'Z'MSG::       .ASCIZ  / P2/
                                        .NLIST
                                        .LIST   MEB
                                        .EVEN
                                .ENDM   PASTST
                                .MACRO  CKPNT
                                .MCALL  PTPNT
                                        .IF NE F$
                                        .ERROR  ;CKPNT MACRO MUST FOLLOW STATST MACRO, NOT ANOTHER CKPNT CALL
                                        .MEXIT
                                        .ENDC
                                        .RADIX  10
                                        F$ = 1
                                        PTPNT   \T$NUM
                                        .RADIX  8
                                .ENDM   CKPNT
                                .MACRO  PTPNT   N,?A
                                        RFLAGS  R0                      ; READ CURRENT FLAGS
                                        CMP     @PNT,R0                 ; IS PNT FLAG SET?
                                        BNE     A                       ; BRANCH IF NOT SET
                                        PRINTF  @PNTMSG,@T'N'MSG        ; PRINT MESSAGE IF PNT SET
                                A:      CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                .ENDM   PTPNT
002270      045     124     000 PNTMSG::        .ASCIZ  /%T/
        000000                  T$NUM = 0
        000001                  F$ = 1
```

MACRO DEFINITIONS

```
                                ;
                                ;MACRO TO INTERFACE WITH $HLP DUMP HELP FILE SUBROUTINE
                                ;
                                .MACRO  HLPMAC  N
                                .LIST MEB
                                .NCHR   $SYM,N
                                        JSR     R5,$HLP                 ;JUMP TO SUBROUTINE
                                        .ASCII  /N/                     ;NAME OF FILE
                                        $TEMP = 10.-$SYM
                                        .REPT   $TEMP
                                        .BYTE   0
                                        .ENDR
                                .ENDM

002273                          STARS
                                ;;*****************************************************************
                                ;*
                                ;*                              GLOBAL EQUATE SECTION
                                ;*
002273                          STARS
                                ;;*****************************************************************

002273                          EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
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
                                ;
001000                          BIT9=  BIT09
000400                          BIT8=  BIT08
000200                          BIT7=  BIT07
000100                          BIT6=  BIT06
000040                          BIT5=  BIT05
000020                          BIT4=  BIT04
000010                          BIT3=  BIT03
000004                          BIT2=  BIT02
000002                          BIT1=  BIT01
000001                          BIT0=  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
```

MACRO DEFINITIONS

```
                000040          EF.START==      32.             ; START COMMAND WAS ISSUED
                000037          EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
                000036          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
                000035          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
                000034          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
                000340          PRI07== 340
                000300          PRI06== 300
                000240          PRI05== 240
                000200          PRI04== 200
                000140          PRI03== 140
                000100          PRI02== 100
                000040          PRI01== 40
                000000          PRI00== 0
                                ;
                                ;OPERATOR FLAG BITS
                                ;
                000004          EVL==     4
                000010          LOT==    10
                000020          ADR==    20
                000040          IDU==    40
                000100          ISR==   100
                000200          UAM==   200
                000400          BOE==   400
                001000          PNT==  1000
                002000          PRI==  2000
                004000          IXE==  4000
                010000          IBE== 10000
                020000          IER== 20000
                040000          LOE== 40000
                100000          HOE== 100000
176                             ;
177                             ; STANDARD REGISTER BASE ADDRESS CONFIGURATION
178                             ;
179             160500          QCMDAD  == 160500       ;STANDARD BASE ADDRESS FOR UNIT 0
180             000500          VEC     == 500          ;STANDARD VECTOR ADDRESS FOR UNIT 0
181
182                             ;
183                             ;BIT DEFINITIONS
184                             ;
185                             ;       QCMD - Q-BUS COMMAND REGISTER
186                             ;
187             000001          CMDACK  == BIT00        ; WO COMMAND ACKNOWLDEGE
188             000100          PRSIE   == BIT06        ; R/W INTERRUPT ENABLE
189                             ;
190                             ;       QVNT - Q-BUS EVENT REGISTER
191                             ;
192             000001          VNTPRS  == BIT00        ; WO EVENT PRESENT
193             000100          ACKIE   == BIT06        ; R/W INTERRUPT ENABLE
194                             ;
195                             ;       QINT - Q-BUS INTERRUPT SUMMARY REGISTER
196                             ;
197             000100          ERRIE   == BIT06        ; R/W INTERRUPT ENABLE
198             001000          MRDY    == BIT09        ; RO MIRROR OF QDMA-07, READY
```

J2

MACRO DEFINITIONS

```
199          002000                  VNTACK  == BIT10        ; RO EVENT ACKNOWLEDGE
200          004000                  CMDPRS  == BIT11        ; RO COMMAND PRESENT
201          010000                  UPWRFL  == BIT12        ; RO UNIBUS POWER FAIL
202          020000                  QNEX    == BIT13        ; RO Q-BUS NON EXISTENT MEMORY ADDRESS
203          040000                  UNEX    == BIT14        ; RO UNIBUS NON EXISTENT MEMORY ADDRESS
204          100000                  ERR     == BIT15        ; RO ERROR
205                                  ;
206                                  ;       QCTL - Q-BUS CONTROL REGISTER
207                                  ;
208          000001                  WDTRST  == BIT00        ; WO WATCHDOG TIMER RESET
209          000002                  TMRENB  == BIT01        ; R/W TIMER ENABLE
210          000004                  WDT0    == BIT02        ; WO WATCHDOG TIMER TIME-OUT BIT (1 OF 2)
211          000010                  WDT1    == BIT03        ; WO WATCHDOG TIMER TIME-OUT BIT (2 OF 2)
212          000020                  QVREN   == BIT04        ; Q-VIEW READ ENABLE
213          000000                  TIME.5  == 0            ;1/2 SEC TIME CODE
214          000004                  TIME1   == WDT0         ;1 SEC TIME CODE
215          000010                  TIME2   == WDT1         ;2 SEC TIME CODE
216          000014                  TIME4   == WDT0!WDT1    ;4 SEC TIME CODE
217                                  ;
218                                  ;
219                                  ;       QDMA - Q-BUS DMA CONTROL AND STATUS REGISTER
220                                  ;
221          000001                  GO      == BIT00        ; WO INITIATES DMA TRANSFER
222          000002                  DMAWT   == BIT01        ; R/W DIRECTION FLOW OF DMA
223          000020                  UXAD16  == BIT04        ; R/W UNIBUS EXTENDED ADDRESS BIT 16
224          000040                  UXAD17  == BIT05        ; R/W UNIBUS EXTENDED ADDRESS BIT 17
225          000100                  RDYIE   == BIT06        ; R/W INTERRUPT ENABLE
226          000200                  RDY     == BIT07        ; RO READY (CLEARED BY GO)
227          000400                  QXAD16  == BIT08        ; R/W Q-BUS EXTENDED ADDRESS BIT 16
228          001000                  QXAD17  == BIT09        ; R/W Q-BUS EXTENDED ADDRESS BIT 17
229          002000                  QXAD18  == BIT10        ; R/W Q-BUS EXTENDED ADDRESS BIT 18
230          004000                  QXAD19  == BIT11        ; R/W Q-BUS EXTENDED ADDRESS BIT 19
231          010000                  QXAD20  == BIT12        ; R/W Q-BUS EXTENDED ADDRESS BIT 20
232          020000                  QXAD21  == BIT13        ; R/W Q-BUS EXTENDED ADDRESS BIT 21
233                                  ;
234                                  ;       QUBA - Q-BUS - UNIBUS BUS ADDRESS REGISTER
235                                  ;
236          000001                  QUBA00  == BIT00        ; WO UNIBUS TRANSFER ADDRESS BIT 00
237          000002                  QUBA01  == BIT01        ; WO UNIBUS TRANSFER ADDRESS BIT 01
238          000004                  QUBA02  == BIT02        ; WO UNIBUS TRANSFER ADDRESS BIT 02
239          000010                  QUBA03  == BIT03        ; WO UNIBUS TRANSFER ADDRESS BIT 03
240          000020                  QUBA04  == BIT04        ; WO UNIBUS TRANSFER ADDRESS BIT 04
241          000040                  QUBA05  == BIT05        ; WO UNIBUS TRANSFER ADDRESS BIT 05
242          000100                  QUBA06  == BIT06        ; WO UNIBUS TRANSFER ADDRESS BIT 06
243          000200                  QUBA07  == BIT07        ; WO UNIBUS TRANSFER ADDRESS BIT 07
244          000400                  QUBA08  == BIT08        ; WO UNIBUS TRANSFER ADDRESS BIT 08
245          001000                  QUBA09  == BIT09        ; WO UNIBUS TRANSFER ADDRESS BIT 09
246          002000                  QUBA10  == BIT10        ; WO UNIBUS TRANSFER ADDRESS BIT 10
247          004000                  QUBA11  == BIT11        ; WO UNIBUS TRANSFER ADDRESS BIT 11
248          010000                  QUBA12  == BIT12        ; WO UNIBUS TRANSFER ADDRESS BIT 12
249          020000                  QUBA13  == BIT13        ; WO UNIBUS TRANSFER ADDRESS BIT 13
250          040000                  QUBA14  == BIT14        ; WO UNIBUS TRANSFER ADDRESS BIT 14
251          100000                  QUBA15  == BIT15        ; WO UNIBUS TRANSFER ADDRESS BIT 15
252                                  ;
253                                  ;       QBA - BUS ADDRESS REGISTER
254                                  ;
255          000001                  QBA00   == BIT00        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 00
```

MACRO DEFINITIONS

```
256         000002          QBA01   == BIT01        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 01
257         000004          QBA02   == BIT02        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 02
258         000010          QBA03   == BIT03        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 03
259         000020          QBA04   == BIT04        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 04
260         000040          QBA05   == BIT05        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 05
261         000100          QBA06   == BIT06        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 06
262         000200          QBA07   == BIT07        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 07
263         000400          QBA08   == BIT08        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 08
264         001000          QBA09   == BIT09        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 09
265         002000          QBA10   == BIT10        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 10
266         004000          QBA11   == BIT11        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 11
267         010000          QBA12   == BIT12        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 12
268         020000          QBA13   == BIT13        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 13
269         040000          QBA14   == BIT14        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 14
270         100000          QBA15   == BIT15        ; R/W Q-BUS BUS ADDRESS REGISTER BIT 15
271                         ;
272                         ;       QWC - Q-BUS DMA WORD COUNT REGISTER
273                         ;
274         000001          QWC00   == BIT00        ; R/W DMA WORD COUNT REGISTER BIT00
275         000002          QWC01   == BIT01        ; R/W DMA WORD COUNT REGISTER BIT01
276         000004          QWC02   == BIT02        ; R/W DMA WORD COUNT REGISTER BIT02
277         000010          QWC03   == BIT03        ; R/W DMA WORD COUNT REGISTER BIT03
278         000020          QWC04   == BIT04        ; R/W DMA WORD COUNT REGISTER BIT04
279         000040          QWC05   == BIT05        ; R/W DMA WORD COUNT REGISTER BIT05
280         000100          QWC06   == BIT06        ; R/W DMA WORD COUNT REGISTER BIT06
281         000200          QWC07   == BIT07        ; R/W DMA WORD COUNT REGISTER BIT07
282         000400          QWC08   == BIT08        ; R/W DMA WORD COUNT REGISTER BIT08
283         001000          QWC09   == BIT09        ; R/W DMA WORD COUNT REGISTER BIT09
284         002000          QWC10   == BIT10        ; R/W DMA WORD COUNT REGISTER BIT10
285         004000          QWC11   == BIT11        ; R/W DMA WORD COUNT REGISTER BIT11
286         010000          QWC12   == BIT12        ; R/W DMA WORD COUNT REGISTER BIT12
287         020000          QWC13   == BIT13        ; R/W DMA WORD COUNT REGISTER BIT13
288         040000          QWC14   == BIT14        ; R/W DMA WORD COUNT REGISTER BIT14
289         100000          QWC15   == BIT15        ; R/W DMA WORD COUNT REGISTER BIT15
290                         ;
291                         ;       QMT1 - Q-BUS MAINTENANCE REGISTER 1
292                         ;
293         000002          MVADR   == BIT01        ; R/W MAINTENANCE VIEW ADDRESS
294         000004          FUNC0   == BIT02        ; R/W SELECT MAINTENANCE MODE BIT 02 FOR QMTO
295         000010          FUNC1   == BIT03        ; R/W SELECT MAINTENANCE MODE BIT 03 FOR QMTO
296         000020          FUNC2   == BIT04        ; R/W SELECT MAINTENANCE MODE BIT 04 FOR QMTO
297         000040          MQPFL   == BIT05        ; R/W MAINTENANCE Q-BUS POWER FAIL
298         000100          MQHALT  == BIT06        ; WO MAINTENANCE Q-BUS HALT
299         000200          MIODIS  == BIT07        ; R/W MAINTENANCE I/O DISABLE
300                         ;
301                         ;       QVAD - Q-BUS VIEW ADDRESS
302                         ;
303         000001          QVAD00  == BIT00        ; R/W Q-BUS VIEW ADDRESS REG. BIT 00
304         000002          QVAD01  == BIT01        ; R/W Q-BUS VIEW ADDRESS REG. BIT 01
305         000004          QVAD02  == BIT02        ; R/W Q-BUS VIEW ADDRESS REG. BIT 02
306         000010          QVAD03  == BIT03        ; R/W Q-BUS VIEW ADDRESS REG. BIT 03
307         000020          QVAD04  == BIT04        ; R/W Q-BUS VIEW ADDRESS REG. BIT 04
308         000040          QVAD05  == BIT05        ; R/W Q-BUS VIEW ADDRESS REG. BIT 05
309         000100          QVAD06  == BIT06        ; R/W Q-BUS VIEW ADDRESS REG. BIT 06
310         000200          QVAD07  == BIT07        ; R/W Q-BUS VIEW ADDRESS REG. BIT 07
311         000400          QVAD08  == BIT08        ; R/W Q-BUS VIEW ADDRESS REG. BIT 08
312         001000          QVAD09  == BIT09        ; R/W Q-BUS VIEW ADDRESS REG. BIT 09
```

MACRO DEFINITIONS

```
                        QVAD10  == BIT10        ; R/W Q-BUS VIEW ADDRESS REG. BIT 10
                        ;
                        ;       QVDA - Q-BUS VIEW DATA REGISTER
                        ;
000001                  QVDA00  == BIT00        ; R/W Q-BUS VIEW DATA REG. BIT00
000002                  QVDA01  == BIT01        ; R/W Q-BUS VIEW DATA REG. BIT01
000004                  QVDA02  == BIT02        ; R/W Q-BUS VIEW DATA REG. BIT02
000010                  QVDA03  == BIT03        ; R/W Q-BUS VIEW DATA REG. BIT03
000020                  QVDA04  == BIT04        ; R/W Q-BUS VIEW DATA REG. BIT04
000040                  QVDA05  == BIT05        ; R/W Q-BUS VIEW DATA REG. BIT05
000100                  QVDA06  == BIT06        ; R/W Q-BUS VIEW DATA REG. BIT06
000200                  QVDA07  == BIT07        ; R/W Q-BUS VIEW DATA REG. BIT07
000400                  QVDA08  == BIT08        ; R/W Q-BUS VIEW DATA REG. BIT08
001000                  QVDA09  == BIT09        ; R/W Q-BUS VIEW DATA REG. BIT09
002000                  QVDA10  == BIT10        ; R/W Q-BUS VIEW DATA REG. BIT10
004000                  QVDA11  == BIT11        ; R/W Q-BUS VIEW DATA REG. BIT11
010000                  QVDA12  == BIT12        ; R/W Q-BUS VIEW DATA REG. BIT12
020000                  QVDA13  == BIT13        ; R/W Q-BUS VIEW DATA REG. BIT13
040000                  QVDA14  == BIT14        . R/W Q-BUS VIEW DATA REG. BIT14
100000                  QVDA15  == BIT15        ; R/W Q-BUS VIEW DATA REG. BIT15
                        ;
                        ;       MAINTENANCE FUNCTION, BIT IN QMT1 TO DEFINE USE OF QMT0
                        ;
000004                  SELQEVENT == 000004     ; SELECT QEVENT
000010                  SELQCMD   == 000010     ; SELECT QCMD
000014                  SELUVADR  == 000014     ; SELECT UVADR
000020                  SELUBAR   == 000020     ; SELECT UBAR
000024                  SELDMADAT == 000024     ; SELECT DMADAT
000030                  SELUVDAT  == 000030     ; SELECT UVDAT
000034                  SELRDDMA  == 000034     ; SELECT DMA READ
                        ;
                        ; MEMORY MANAGEMENT MAPPING ADDRESSES
                        ;
177572                  MMR0      == 177572
172516                  MMR3      == 172516
172300                  KISDR0    == 172300
172340                  KISAR0    == 172340
170370                  HIUBM     == 170370
                        ;
                        ;       MISC. EQUATES
                        ;
000010                  MAXUNITS == 8.          ; MAXIMUM NUMBER OF UNITS ALLOWED
000000                  NO       == 0           ; LOGICAL VALUE FOR NO
000001                  YES      == 1           ; LOGICAL VALUE FOR YES
020040                  WSPACE   == 20040       ; WORD OF ASCII SPACES
```

```
                                .SBTTL  GLOBAL DATA SECTION

002273                          STARS
                                ;;*****************************************************************
                                ;*
                                ;*      GLOBAL DATA SECTION
                                ;*
002273                          STARS
                                ;;*****************************************************************

                                ;
                                ; CURRENT UNIT Q-BUS BRIDGE INTERFACE REGISTERS ADDRESSES ARE STORED HERE
                                ;
                                .EVEN
002274                          ADDRESSES::
002274  000000                  QCMD::          .WORD   0       ; Q-BUS COMMAND REGISTER
002276  000000                  QVNT::          .WORD   0       ; Q-BUS EVENT REGISTER
002300  000000                  QINT::          .WORD   0       ; Q-BUS INTERRUPT SUMMARY REGISTER
002302  000000                  QCTL::          .WORD   0       ; Q-BUS CONTROL REGISTER
002304  000000                  QDMA::          .WORD   0       ; Q-BUS DMA CONTROL & STATUS REGISTER
002306  000000                  QUBA::          .WORD   0       ; Q-BUS BUS ADDRESS UNIBUS REGISTER
002310  000000                  QBA::           .WORD   0       ; Q-BUS BUS ADDRESS Q-BUS REGISTER
002312  000000                  QWC::           .WORD   0       ; Q-BUS WORD COUNT REGISTER
002314  000000                  QMT0::          .WORD   0       ; Q-BUS MAINTENANCE REGISTER 0
002316  000000                  QMT1::          .WORD   0       ; Q-BUS MAINTENANCE REGISTER 1
002320  000000                  QVAD::          .WORD   0       ; Q-BUS VIEW ADDRESS REGISTER
002322  000000                  QVDA::          .WORD   0       ; Q-BUS DATA REGISTER

                                ;
                                ; CURRENT UNIT Q-BUS BRIDGE INTERFACE REGISTERS VALUES ARE STORED HERE
                                ;
002324                          VALUES::
002324  000000                  SQCMD::         .WORD   0       ; COMMAND REGISTER
002326  000000                  SQVNT::         .WORD   0       ; EVENT REGISTER
002330  000000                  SQINT::         .WORD   0       ; INTERRUPT SUMMARY REGISTER
002332  000000                  SQCTL::         .WORD   0       ; CONTROL REGISTER
002334  000000                  SQDMA::         .WORD   0       ; DMA CONTROL & STATUS REGISTER
002336  000000                  SQUBA::         .WORD   0       ; BUS ADDRESS UNIBUS REGISTER
002340  000000                  SQBA::          .WORD   0       ; BUS ADDRESS REGISTER
002342  000000                  SQWC::          .WORD   0       ; WORD COUNT REGISTER
002344  000000                  SQMT0::         .WORD   0       ; MAINTENANCE REGISTER 0
002346  000000                  SQMT1::         .WORD   0       ; MAINTENANCE REGISTER 1
002350  000000                  SQVAD::         .WORD   0       ; VIEW ADDRESS REGISTER
002352  000000                  SQVDA::         .WORD   0       ; DATA REGISTER

002354                          EXPECTED::                      ;USED FOR TABLE PROCESSING OF EXPECTED RESULTS
002354  000000                  EXQCMD::        .WORD   0       ; COMMAND REGISTER
002356  000000                  EXQVNT::        .WORD   0       ; EVENT REGISTER
002360  000000                  EXQINT::        .WORD   0       ; INTERRUPT SUMMARY REGISTER
002362  000000                  EXQCTL::        .WORD   0       ; CONTROL REGISTER
002364  000000                  EXQDMA::        .WORD   0       ; DMA CONTROL & STATUS REGISTER
002366  000000                  EXQUBA::        .WORD   0       ; BUS ADDRESS UNIBUS REGISTER
002370  000000                  EXQBA::         .WORD   0       ; BUS ADDRESS REGISTER
002372  000000                  EXQWC::         .WORD   0       ; WORD COUNT REGISTER
002374  000000                  EXQMT0::        .WORD   0       ; MAINTENANCE REGISTER 0
002376  000000                  EXQMT1::        .WORD   0       ; MAINTENANCE REGISTER 1
002400  000000                  EXQVAD::        .WORD   0       ; VIEW ADDRESS REGISTER
```

GLOBAL DATA SECTION

```
                                        EXQVDA::        .WORD   0       ; DATA REGISTER
                                        ;
                                        ;Q-BRIDGE REGISTER NAMES ARE STORED HERE. ACCESS IS INDEXED, EACH TAKING 2 WORDS
                                        ;
                                                .EVEN
002404  121     103     115     NAMES:: .ASCIZ  /QCMD/
002407  104     000
002411  121     126     116             .ASCIZ  /QVNT/
002414  124     000
002416  121     111     116             .ASCIZ  /QINT/
002421  124     000
002423  121     103     124             .ASCIZ  /QCTL/
002426  114     000
002430  121     104     115             .ASCIZ  /QDMA/
002433  101     000
002435  121     125     102             .ASCIZ  /QUBA/
002440  101     000
002442  121     102     101             .ASCIZ  /QBA /
002445  040     000
002447  121     127     103             .ASCIZ  /QWC /
002452  040     000
002454  121     115     124             .ASCIZ  /QMT0/
002457  060     000
002461  121     115     124             .ASCIZ  /QMT1/
002464  061     000
002466  121     126     101             .ASCIZ  /QVAD/
002471  104     000
002473  121     126     104             .ASCIZ  /QVDA/
002476  101     000
                                                .EVEN
                                        ;
                                        ;       MISC STORAGE AREAS
                                        ;
000000                                  BRIDGETYPE::    .WORD   0       ;BRIDGE TYPE (0=Q-BRIDGE 1=U-BRIDGE)
000000                                  BRDGFLG::       .WORD   0       ;BRIDGE CONFIGURED FLAG
000000                                  LOGUNIT::       .WORD   0       ;USED TO STORE CURRENT LOGICAL UNIT
000000                                  PLOC::          .WORD   0       ;POINTER TO CURRENT HARDWARE P-TABLE
000000                                  BASEADD::       .WORD   0       ;STORES BASE ADDRESS
000000                                  VECTOR::        .WORD   0       ;STORES VECTOR ADDRESS
000000                                  LOOPBK::        .WORD   0       ;STORES SOFTWARE LOOPBACK QUES. VALUE FOR CURRENT UNIT
000000                                  BRLEVEL::       .WORD   0       ;STORES BR LEVEL
000000                                  INITWORD::      .WORD   0       ;STORES INITIALIZE WORD SENT TO QMT1
000000                                  TRPFLG::        .WORD   0       ;TRAP FLAG INDICATOR
000000                                  ADD::           .WORD   0       ;STORES ADDRESS OF REGISTER FOR MESSAGES
000000                                  VAL::           .WORD   0       ;STORES VALUE OF REGISTER FOR MESSAGES
000000                                  EXP::           .WORD   0       ;STORES EXPECTED VALUE OF TEST FOR MESSAGES
000000                                  ERROR1::        .WORD   0       ;STORES ERROR COUNTS
000000                                  ERROR2::        .WORD   0       ;STORES ERROR COUNTS
000000                                  RAM::           .WORD   0       ;TELLS WHETHER RAM MESSAGE WAS PRINTED OUT
000000                                  VALUE::         .WORD   0       ;STORES CURRENT CPU PRIORITY LEVEL
000000                                  INTFLG::        .WORD   0       ;INTERRUPT FLAG
000000                                  HP1::           .WORD   0       ;TELLS WHETHER DRS HELP MESSAGE ALREADY PRINTED
000000                                  HP2::           .WORD   0       ;TELLS WHETHER DIA HELP MESSAGE ALREADY PRINTED
000000                                  MEMSIZ::        .WORD   0       ;STORE FREE MEMORY SIZE IN WORDS ABOVE DIAGNOSTIC
```

GLOBAL DATA SECTION

```
439 002552  000000                FREMEM::        .WORD   0       ;POINTS TO STARTING ADDRESS OF FREE MEMORY
440
```

GLOBAL TEXT SECTION

```
                                .SBTTL  GLOBAL TEXT SECTION

                                ;**
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
                                        DEVTYP  <FEP>
                                L$DVTYP::
                                        .ASCIZ  /FEP/

                                        .EVEN

                                ; TEST DESCRIPTION
                                ;
                                        DESCRIPT        <COMMAND BRIDGE INTERFACE DIAGNOSTIC>
                                L$DESC::
                                        .ASCIZ  /COMMAND BRIDGE INTERFACE DIAGNOSTIC/

                                        .EVEN
                                        .EVEN
```

```
002554
002554
002554  106  105  120
002557  000

002560
002560
002560  103  117  115
002563  115  101  116
002566  104  040  102
002571  122  111  104
002574  107  105  040
002577  111  116  124
002602  105  122  106
002605  101  103  105
002610  040  104  111
002613  101  107  116
002616  117  123  124
002621  111  103  000
```

GLOBAL TEXT SECTION

```
                                        .NLIST  BEX
                                        ;
                                        ;       BIT NAMES IN THE REGISTERS
                                        ;
002624      054     054     054  CMDBIT:: .ASCIZ /.........PRSIE......CMDACK/
002657      054     054     054  VNTBIT:: .ASCIZ /.........ACKIE......VNTPRS/
002712      105     122     122  INTBIT:: .ASCIZ /ERR,UNEX,QNEX,UPWRFL,CMDPRS,VNTACK,RDY,,,ERRIE/
002771      054     054     054  CTLBIT:: .ASCIZ /............QVREN,WDT1,WDT0,TMRENB,WDTRST/
003042      054     054     121  DMABIT:: .ASCII /,,QXAD21,QXAD20,QXAD19,QXAD18,QXAD17,QXAD16,RDY,RDYIE/
003127      054     125     130           .ASCIZ /,UXAD17,UXAD16,,,DMAWT,GO/
003161      054     054     054  MT1BIT:: .ASCIZ /........MIODIS,MQHLT,MQPFL,FUNC2,FUNC1,FUNC0,MVADR/

003244                               STARS
                                     ;;*****************************************************************
                                     ;
                                     ; FORMAT STATEMENTS USED IN PRINT CALLS
                                     ;
003244                               STARS
                                     ;;*****************************************************************

                                     ;.IF NDF ONEFILE
                                     .NLIST  BEX
                                     ;.ENDC

003244      045     116     061  CONFIG:: .ASCIZ /%N1%ACONFIGURED FOR %T%A BRIDGE MODULE%N1/
003316      121     055     102  QBUS::  .ASCIZ  /Q-BUS/
003324      125     116     111  UBUS::  .ASCIZ  /UNIBUS/

003333      045     116     061  FOR0::  .ASCIZ  /%N1/                    ;LINE FEED CARRIAGE RETURN ONLY
003337      045     116     061  FOR1::  .ASCIZ  /%N1%ANAME%S4%AADDRESS%S4%ACONTENTS%S4%AEXPECTED (UNDETERMINED BITS ARE ZERO
)/
003454      045     116     061  FOR2::  .ASCIZ  /%N1%T%S5%O6%S5%O6%S6%O6/
003504      045     124     045  FOR3::  .ASCIZ  /%T%A REGISTER AT %O6%A CANNOT BE READ/
003552      045     116     061  FOR4::  .ASCIZ  /%N1%S4%O4%S9%O6%S7%O6/
003600                           FOR4.5::
003600      045     116     061          .ASCIZ  /%N1%AUNKNOWN VIEW RAM SIZE, 1K ASSUMED FOR REST OF TEST/
003670      045     116     061  FOR5::  .ASCIZ  /%N1%AQVAD Q RAM ADDRESS INCREMENT SHOULD BE %O4%A WAS %O4/
003762                           FOR5.5::
003762      045     116     061          .ASCIZ  /%N1%AQVAD U RAM ADDRESS INCREMENT SHOULD BE %O4%A WAS %O4/
004054      045     116     061  FOR6::  .ASCIZ  /%N1%ATOTAL RAM CHECK ERRORS %D5/
004114      045     116     061  FOR7::  .ASCIZ  /%N1%ATOTAL INCREMENT ERRORS %D5/
004154      045     116     061  FOR8::  .ASCIZ  /%N1%ARAM ADDRESS     CONTENTS     EXPECTED/
004227      045     116     061  FOR9::  .ASCIZ  /%N1%AVIEW RAM SIZE %D1%AK/
004261      045     116     061  FOR10:: .ASCIZ  /%N1%ASHOULD BE %O6%A, WAS %O6%A LOOP #%D3%A OUT OF 100%N1/
004353      045     116     061  FOR11:: .ASCII  /%N1%ATEST 37: BELL WILL RING EACH TIME WDTO OCCURS (20 TIMES /
004450      124     117     124          .ASCIZ  /TOTAL)%N1/
004462      045     101     007  FOR12:: .ASCIZ  /%A/<7>
004466      045     116     045  FOR13:: .ASCIZ  /%N%ABR LEVEL %O1%A CPU PRIORITY LEVEL %O1%N1/
                                         .EVEN
```

GLOBAL TEXT SECTION

```
                                        STARS
                                        ;;*****************************************************************
                                        ;
                                        ; ASCII MESSAGES USED IN ERROR REPORTING
                                        ;
                                        STARS
                                        ;;*****************************************************************

004544      102     122     111  MSG1::   .ASCIZ  /BRIDGE REGISTER DUMP/
004571      102     101     104  MSG2::   .ASCIZ  /BAD ADDRESS ERROR - PASS ABORTED/
004632      122     105     107  MSG3::   .ASCIZ  /REGISTER READ AFTER RESET ERROR/
004672      114     117     101  MSG4::   .ASCIZ  /LOAD WITH ONE'S AND READ ERROR/
004731      114     117     101  MSG5::   .ASCIZ  /LOAD WITH ZERO'S AND READ ERROR/
004771      121     103     114  MSG6::   .ASCIZ  /QCLR RESET ERROR/
005012      121     103     115  MSG7::   .ASCIZ  /QCMD ROTATING 1 IN BITS 1-5 ERROR/
005054      121     103     115  MSG8::   .ASCIZ  /QCMD ROTATING 0 IN BITS 1-5 ERROR/
005116      121     103     115  MSG9::   .ASCIZ  /QCMD ROTATING 1 IN BITS 8-15 ERROR/
005161      121     103     115  MSG10::  .ASCIZ  /QCMD ROTATING 0 IN BITS 8-15 ERROR/
005224      121     126     116  MSG11::  .ASCIZ  /QVNT ROTATING 1 IN BITS 1-5 ERROR/
005266      121     126     116  MSG12::  .ASCIZ  /QVNT ROTATING 0 IN BITS 1-5 ERROR/
005330      121     126     116  MSG13::  .ASCIZ  /QVNT ROTATING 1 IN BITS 8-15 ERROR/
005373      121     126     116  MSG14::  .ASCIZ  /QVNT ROTATING 0 IN BITS 8-15 ERROR/
005436      121     111     116  MSG15::  .ASCIZ  /QINT BIT 11 (CMDPRS) SHOULD BE CLEAR AFTER SET QCMD CMDACK/
005531      121     111     116  MSG16::  .ASCIZ  /QINT BIT 10 (VNTACK) SHOULD BE SET AFTER SET QCMD CMDACK/
005622      121     111     116  MSG17::  .ASCIZ  /QINT BIT 10 (VNTACK) SHOULD BE CLEAR AFTER SET QVNT VNTPRS/
005715      121     111     116  MSG18::  .ASCIZ  /QINT BIT 11 (CMDPRS) SHOULD BE SET AFTER SET QVNT VNTPRS/
006006      121     102     101  MSG19::  .ASCIZ  /QBA ROTATING 1 ERROR/
006033      121     102     101  MSG20::  .ASCIZ  /QBA ROTATING 0 ERROR/
006060      121     104     115  MSG21::  .ASCIZ  /QDMA HI BITS (8-13) ROTATING 1 ERROR/
006125      121     104     115  MSG22::  .ASCIZ  /QDMA HI BITS (8-13) ROTATING 0 ERROR/
006172      121     104     115  MSG23::  .ASCIZ  /QDMA EXTENDED ADDRESS BITS 4 & 5 ERROR/
006241      121     125     102  MSG24::  .ASCIZ  /QUBA ROTATING 1 ERROR/
006267      121     125     102  MSG25::  .ASCIZ  /QUBA ROTATING 0 ERROR/
006315      121     127     103  MSG26::  .ASCIZ  /QWC ROTATING 1 ERROR/
006342      121     127     103  MSG27::  .ASCIZ  /QWC ROTATING 0 ERROR/
006367      121     126     101  MSG28::  .ASCIZ  /QVAD Q-REGISTER ROTATING 1 ERROR/
006430      121     126     101  MSG29::  .ASCIZ  /QVAD Q-REGISTER ROTATING 0 ERROR/
006471      121     126     101  MSG30::  .ASCIZ  /QVAD U-REGISTER ROTATING 1 ERROR/
006532      121     126     101  MSG31::  .ASCIZ  /QVAD U-REGISTER ROTATING 0 ERROR/
006573      121     126     101  MSG32::  .ASCIZ  /QVAD Q-REGISTER INCREMENT ERROR/
006633      121     126     101  MSG33::  .ASCIZ  /QVAD U-REGISTER INCREMENT ERROR/
006673      121     126     104  MSG34::  .ASCIZ  /QVDA VIEW REGISTER RAM CHECK ERROR/
006736      121     126     101  MSG35::  .ASCIZ  /QVAD RAM ADDRESS INCREMENT ERROR/
006777      121     102     101  MSG36::  .ASCIZ  /QBA INCREMENT ERROR/
007023                           MSG36.5::
007023      121     104     115          .ASCIZ  /QDMA INCREMENT BITS 8-13 EXTENDED ADDRESS ERROR/
007103      121     104     115  MSG37::  .ASCIZ  /QDMA RDY BIT 7 SHOULD BE CLEARED AFTER SET QDMA GO BIT/
007172      121     127     103  MSG38::  .ASCIZ  /QWC SHOULD HAVE INCREMENTED AFTER SET QDMA GO BIT/
007254      121     104     115  MSG39::  .ASCIZ  /QDMA RDY BIT 7 SHOULD BE SET AFTER SET QDMA GO BIT/
007337      121     127     103  MSG40::  .ASCIZ  /QWC INCREMENT AFTER QDMA DMAWT AND GO BIT SET ERROR/
007423      121     104     115  MSG41::  .ASCIZ  /QDMA RDY BIT 7 SHOULD BE SET AFTER QDMA DMAWA AND GO BIT SET/
007520      104     115     101  MSG42::  .ASCIZ  /DMA WRITE DATA TRANSFER ERROR/
007556      104     115     101  MSG42.5:: .ASCIZ /DMA WRITE HI MEMORY DATA TRANSFER ERROR/
007626      104     115     101  MSG43::  .ASCIZ  /DMA READ DATA TRANSFER ERROR/
007663      127     101     124  MSG44::  .ASCII  /WATCH DOG TIMER READ AFTER WRITE TIME CODE BITS 2 & 3 ERROR/
007756      015     012     124          .ASCIZ  <15><12>/TEST ABORTED/
```

GLOBAL TEXT SECTION

```
    603 007775      127     101     124  MSG45:: .ASCII  /WATCH DOG TIMER READ AFTER WRITE TMRENB BIT 1 ERROR/
    604 010060      015     012     124          .ASCIZ  <15><12>/TEST ABORTED/
    605 010077      127     101     124  MSG46:: .ASCIZ  /WATCH DOG TIMER DID NOT TIME OUT/
    606 010140      127     101     124  MSG47:: .ASCIZ  /WATCH DOG TIMO OUT RESET ERROR/
    607 010177      121     126     116  MSG48:: .ASCIZ  /QVNT ACKIE BIT 6 SET TO 1 AND READ AS 0/
    608 010247      111     116     124  MSG49:: .ASCIZ  /INTERRUPT DID NOT OCCUR AFTER SETTING QCMD CMDACK BIT 0/
    609 010337      121     103     115  MSG50:: .ASCIZ  /QCMD PRSIE BIT 6 SET TO 1 AND READ AS 0/
    610 010407      111     116     124  MSG51:: .ASCIZ  /INTERRUPT DID NOT OCCUR AFTER SETTING QVNT VNTPRS BIT 0/
    611 010477      121     104     115  MSG52:: .ASCIZ  /QDMA RDYIE BIT 6 SET TO 1 AND READ AS 0/
    612 010547      111     116     124  MSG53:: .ASCII  /INTERRUPT DID NOT OCCUR AFTER SETTING QDMA GO BIT 0 AND /
    613 010637      104     115     101          .ASCIZ  /DMAWT BIT 1/
    614 010653      121     111     116  MSG54:: .ASCIZ  /QINT ERRIE BIT 6 SET TO 1 AND READ AS 0/
    615 010723      111     116     124  MSG55:: .ASCIZ  /INTERRUPT DID NOT OCCUR AFTER SETTING QMT1 MQPFL BIT 5/
    616 011012      121     111     116  MSG56:: .ASCIZ  /QINT UNEX BIT 14 OR ERR BIT 15 NOT SET AFTER INTERRUPT/
    617 011101      111     116     124  MSG57:: .ASCII  /INTERRUPT DID NOT OCCUR AFTER SETTING QDMA DMAWT BIT 1 /
    618 011170      101     116     104          .ASCIZ  /AND GO BIT 0/
    619 011205      121     111     116  MSG58:: .ASCIZ  /QINT QNEX BIT 13 OR ERR BIT 15 NOT SET AFTER INTERRUPT/
    620 011274      127     104     124  MSG59:: .ASCIZ  /WDTO INTERRUPT DID NOT OCCUR AFTER 8 SECONDS/
    621 011351      121     111     116  MSG60:: .ASCIZ  /QINT UPWRFL, ERR BITS NOT SET AFTER WDTO INTERRUPT/
    622 011434      116     117     040  MSG61:: .ASCIZ  /NO INTERRUPT OCCURRED WITH BR LEVEL > THAN CPU LEVEL/
    623 011521      111     116     124  MSG62:: .ASCIZ  /INTERRUPT OCCURED WITH BR LEVEL <= CPU LEVEL/
    624 011576      120     122     111  HELP1:: .ASCIZ  /PRINT DIAGNOSTIC SUPERVISOR HELP FILE?/
    625 011645      120     122     111  HELP2:: .ASCIZ  /PRINT DIAGNOSTIC HELP FILE?/
    626                                          .EVEN
    627
```

GLOBAL ERROR REPORT SECTION

```
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

                                        ;++
                                        ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
                                        ; USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
                                        ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
                                        ;--


                                        ;       BAD ADDRESS ERROR
                                        ; REGISTER NUMBER (0-11.) EXPECTED TO BE IN R1

011702                                          BGNMSG  ERR2
011702                                  ERR2::
011702  010246                                  MOV     R2,-(SP)
011704  005002                                  CLR     R2                      ;CLEAR ACCUMULATOR
011706  060102                                  ADD     R1,R2                   ;MULTIPLY REGISTER POINTER TIMES 5
011710  060102                                  ADD     R1,R2
011712  060102                                  ADD     R1,R2
011714  060102                                  ADD     R1,R2
011716  060102                                  ADD     R1,R2
011720  062702  002404                          ADD     #NAMES,R2               ;POINT R2 TO REGISTER NAME
011724                                          PRINTB  #FOR3,R2,ADD            ;TELL OPERATOR BAD ADDRESS
011724  013746  002524                          MOV     ADD,-(SP)
011730  010246                                  MOV     R2,-(SP)
011732  012746  003504                          MOV     #FOR3,-(SP)
011736  012746  000003                          MOV     #3,-(SP)
011742  010600                                  MOV     SP,R0
011744  104414                                  TRAP    C$PNTB
011746  062706  000010                          ADD     #10,SP
011752                                          PRINTB  #FOR0                   ;INSERT LINE FEED CARRIAGE RETURN
011752  012746  003333                          MOV     #FOR0,-(SP)
011756  012746  000001                          MOV     #1,-(SP)
011762  010600                                  MOV     SP,R0
011764  104414                                  TRAP    C$PNTB
011766  062706  000004                          ADD     #4,SP
011772  012602                                  MOV     (SP)+,R2                ;RESTORE REGISTERS
011774                                          ENDMSG
011774                                  L10001:
011774  104423                                  TRAP    C$MSG


                                        ;       BAD REGISTER VALUE ERROR (SINGLE REGISTER TEST)
                                        ;REGISTER NUMBER (0-11.) EXPECTED TO BE IN R1
                                        ;REGISTER ADDRESS EXPECTED TO BE IN LOCATION ADD
                                        ;REGISTER VALUE EXPECTED TO BE IN LOCATION VAL
                                        ;REGISTER EXPECTED VALUE EXPECTED TO BE IN LOCATION EXP

011776                                          BGNMSG  ERR4
011776                                  ERR4::
011776  010146                                  MOV     R1,-(SP)                ;SAVE REGISTERS USED IN ROUTINES
012000  010246                                  MOV     R2,-(SP)
012002  005002                                  CLR     R2                      ;CLEAR ACCUMULATOR
012004  060102                                  ADD     R1,R2                   ;MULTIPLY REGISTER POINTER TIMES 5
012006  060102                                  ADD     R1,R2
012010  060102                                  ADD     R1,R2
012012  060102                                  ADD     R1,R2
```

GLOBAL ERROR REPORT SECTION

```
673 012014  060102                         ADD     R1,R2
674 012016  062702  002404                  ADD     #NAMES,R2               ;POINT R2 TO REGISTER NAME
675 012022  006301                          ASL     R1                      ;REGISTER POINTER TIMES 2
676 012024  016137  002274  002524          MOV     ADDRESSES(R1),ADD       ;MOVE CURRENT ADDRESS TO MESSAGE AREA
677 012032                                  PRINTX  #FOR1                   ;PRINT HEADER LINE
    012032  012746  003337                  MOV     #FOR1,-(SP)
    012036  012746  000001                  MOV     #1,-(SP)
    012042  010600                          MOV     SP,R0
    012044  104415                          TRAP    C$PNTX
    012046  062706  000004                  ADD     #4,SP
678 012052                                  PRINTX  #FOR2,R2,ADD,VAL,EXP    ;PRINT FULL ERROR DETAILS
    012052  013746  002530                  MOV     EXP,-(SP)
    012056  013746  002526                  MOV     VAL,-(SP)
    012062  013746  002524                  MOV     ADD,-(SP)
    012066  010246                          MOV     R2,-(SP)
    012070  012746  003454                  MOV     #FOR2,-(SP)
    012074  012746  000005                  MOV     #5,-(SP)
    012100  010600                          MOV     SP,R0
    012102  104415                          TRAP    C$PNTX
    012104  062706  000014                  ADD     #14,SP
679 012110                                  PRINTX  #FOR0                   ;INSERT LINE FEED CARRIAGE RETURN
    012110  012746  003333                  MOV     #FOR0,-(SP)
    012114  012746  000001                  MOV     #1,-(SP)
    012120  010600                          MOV     SP,R0
    012122  104415                          TRAP    C$PNTX
    012124  062706  000004                  ADD     #4,SP
680 012130  012602                          MOV     (SP)+,R2                ;RESTORE REGISTERS
681 012132  012601                          MOV     (SP)+,R1
682 012134                                  ENDMSG
    012134  104423                  L10002:
                                            TRAP    C$MSG
683
684                                 ;BAD VIEW REGISTER RAM ERROR
685                                 ;PRINTS HEADER ONLY
686
687 012136                                  BGNMSG  ERR5
    012136                          ERR5::
688 012136                                  PRINTX  #FOR8                   ;PRINT HEADER
    012136  012746  004154                  MOV     #FOR8,-(SP)
    012142  012746  000001                  MOV     #1,-(SP)
    012146  010600                          MOV     SP,R0
    012150  104415                          TRAP    C$PNTX
    012152  062706  000004                  ADD     #4,SP
689 012156                                  ENDMSG
    012156                          L10003:
    012156  104423                          TRAP    C$MSG
690
691                                 ;BR LEVEL INTERRUPT LEVEL TEST ERROR ROUTINE
692
693 012160                                  BGNMSG  ERR6
    012160                          ERR6::
694 012160  010146                          MOV     R1,-(SP)                ;SAVE REGISTERS USED IN ROUTINES
695 012162  010246                          MOV     R2,-(SP)
696 012164  010346                          MOV     R3,-(SP)
697 012166  013701  002516                  MOV     BRLEVEL,R1              ;DEVICE PRIORITY TO R1
698 012172  013702  002540                  MOV     VALUE,R2                ;CPU PRIORITY TO R2
699 012176  012703  000005                  MOV     #5,R3                   ;LOOP COUNTER TO R3
```

```
                                        ;SHIFT PRIORITY RIGHT 5 BITS FOR MESSAGE
                                10$:    ASR     R1                      ;SHIFT R1 RIGHT
                                        ASR     R2                      ;SHIFT R2 RIGHT
                                        DEC     R3                      ;DECREMENT COUNTER
                                        BNE     10$                     ;IF NOT DONE, BRANCH
                                ;PRINT ERROR MESSAGE
                                        PRINTB  #FOR13,R1,R2            ;TELL OPERATOR DETAILS
012212  010246                          MOV     R2,-(SP)
012214  010146                          MOV     R1,-(SP)
012216  012746  004466                  MOV     #FOR13,-(SP)
012222  012746  000003                  MOV     #3,-(SP)
012226  010600                          MOV     SP,R0
012230  104414                          TRAP    C$PNTB
012232  062706  000010                  ADD     #10,SP
012236  012603                          MOV     (SP)+,R3
012240  012602                          MOV     (SP)+,R2                ;RESTORE REGISTERS
012242  012601                          MOV     (SP)+,R1
012244                                  ENDMSG
012244                          L10004:
012244  104423                          TRAP    C$MSG
                                ;       BAD REGISTER VALUE ERROR (SINGLE REGISTER TEST)
                                ;       USED WITH ROTATING BITS TESTS
                                ;REGISTER NUMBER (0-11.) EXPECTED TO BE IN R1
                                ;REGISTER ADDRESS EXPECTED TO BE IN LOCATION ADD
                                ;REGISTER VALUE EXPECTED TO BE IN LOCATION VAL
                                ;REGISTER EXPECTED VALUE EXPECTED TO BE IN LOCATION EXP

012246                                  BGNMSG  ERR7
012246                          ERR7::
012246  010146                          MOV     R1,-(SP)                ;SAVE REGISTERS USED IN ROUTINES
012250  010246                          MOV     R2,-(SP)
012252  005002                          CLR     R2                      ;CLEAR ACCUMULATOR
012254  060102                          ADD     R1,R2                   ;MULTIPLY REGISTER POINTER TIMES 5
012256  060102                          ADD     R1,R2
012260  060102                          ADD     R1,R2
012262  060102                          ADD     R1,R2
012264  060102                          ADD     R1,R2
012266  062702  002404                  ADD     #NAMES,R2               ;POINT R2 TO REGISTER NAME
012272  006301                          ASL     R1                      ;REGISTER POINTER TIMES 2
012274  016137  002274  002524          MOV     ADDRESSES(R1),ADD       ;MOVE CURRENT ADDRESS TO MESSAGE AREA
012302                                  PRINTX  #FOR1                   ;PRINT HEADER LINE
012302  012746  003337                  MOV     #FOR1,-(SP)
012306  012746  000001                  MOV     #1,-(SP)
012312  010600                          MOV     SP,R0
012314  104415                          TRAP    C$PNTX
012316  062706  000004                  ADD     #4,SP
012322                                  PRINTX  #FOR2,R2,ADD,VAL,EXP    ;PRINT FULL ERROR DETAILS
012322  013746  002530                  MOV     EXP,-(SP)
012326  013746  002526                  MOV     VAL,-(SP)
012332  013746  002524                  MOV     ADD,-(SP)
012336  010246                          MOV     R2,-(SP)
012340  012746  003454                  MOV     #FOR2,-(SP)
012344  012746  000005                  MOV     #5,-(SP)
012350  010600                          MOV     SP,R0
012352  104415                          TRAP    C$PNTX
012354  062706  000014                  ADD     #14,SP
012360  012602                          MOV     (SP)+,R2                ;RESTORE REGISTERS
```

GLOBAL ERROR REPORT SECTION

```
    733 012362  01260:                          MOV     (SP)+,R1
    734 012364                                   ENDMSG
        012364                          L10005:
        012364  104423                           TRAP    C$MSG
```

GLOBAL SUBROUTINES SECTION

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;++
                                ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                                ; THAT ARE USED IN MORE THAN ONE TEST.
                                ;--

                                ;
                                ;       THIS SUBROUTINE PRINTS A ONE LINE DETAIL LINE WITHOUT THE
                                ;       THE PRIMARY REGISTER ERROR HEADER LINE.
                                ;
                                ;  INPUTS:
                                ;       REGISTER NUMBER (0-11.) EXPECTED TO BE IN R1
                                ;       REGISTER ADDRESS EXPECTED TO BE IN LOCATION ADD
                                ;       REGISTER VALUE EXPECTED TO BE IN LOCATION VAL
                                ;       REGISTER EXPECTED VALUE EXPECTED TO BE IN LOCATION EXP
                                ;
                                ;  CALLING SEQUENCE:
                                ;       JSR     PC,DETAIL
                                ;

012366                          DETAIL:
012366  010146                          MOV     R1,-(SP)                ;SAVE REGISTERS USED IN ROUTINES
012370  010246                          MOV     R2,-(SP)
012372  005002                          CLR     R2                      ;CLEAR ACCUMULATOR
012374  060102                          ADD     R1,R2                   ;MULTIPLY REGISTER POINTER TIMES 5
012376  060102                          ADD     R1,R2
012400  060102                          ADD     R1,R2
012402  060102                          ADD     R1,R2
012404  060102                          ADD     R1,R2
012406  062702  002404                  ADD     #NAMES,R2               ;POINT R2 TO REGISTER NAME
012412  006301                          ASL     R1                      ;REGISTER POINTER TIMES 2
012414  016137  002274  002524          MOV     ADDRESSES(R1),ADD       ;MOVE CURRENT ADDRESS TO MESSAGE AREA
012422                          PRINTX  #FOR2,R2,ADD,VAL,EXP    ;PRINT FULL ERROR DETAILS
012422  013746  002530                  MOV     EXP,-(SP)
012426  013746  002526                  MOV     VAL,-(SP)
012432  013746  002524                  MOV     ADD,-(SP)
012436  010246                          MOV     R2,-(SP)
012440  012746  003454                  MOV     #FOR2,-(SP)
012444  012746  000005                  MOV     #5,-(SP)
012450  010600                          MOV     SP,R0
012452  104415                          TRAP    C$PNTX
012454  062706  000014                  ADD     #14,SP
012460  012602                          MOV     (SP)+,R2                ;RESTORE REGISTERS
012462  012601                          MOV     (SP)+,R1
012464  000207                          RETURN
```

```
                        ;*****************************************************************************
                        ;**
                        ;                       BITS.MAC
                        ;       PROVIDES CODE FOR A SUBROUTINE TO INTERPRET THE BITS SET
                        ;       IN A WORD (OR REGISTER), AND OPTIONALLY FOR EVERY BIT SET PRINT OUT
                        ;       THE NAME OF THAT BIT IF DEFINED IN AN ASCIZ STATEMENT, AND ALWAYS
                        ;       PUT THE NAMES IN A BUFFER.
                        ;
                        ;       AUTHOR:  DALE PROCTOR
                        ;       DATE:    12/23/81
                        ;       VERSION: 1.00
                        ;
                        ;       ASCIZ STATEMENT FORMAT -
                        ;         LABEL: .ASCIZ /BIT15,BIT14,...,BIT10,..BIT7/
                        ;       MAXIMUM NUMBER OF DEFINITIONS IS 16
                        ;
                        ;       TO USE THE FULL ROUTINE, CALL BY USING THE MACRO SETBITS.
                        ;       SETBITS IS IN THE FORMAT-
                        ;         SETBITS       NAME,POINTER,VALUE
                        ;           WHERE NAME IS AN ASCII NAME THAT WILL BE PRINTED IN THE
                        ;           MESSAGE LINE, POINTER IS AN ADDRESS THAT POINTS TO THE
                        ;           ASCIZ FORMAT STATEMENT THAT CONTAINS THE BIT NAMES, AND
                        ;           VALUE IS THE WORD TO INTERPRET AND CONVERT TO BIT NAMES.
                        ;
                        ;       TO CREATE ONLY A PRINT BUFFER WITH THE SET BIT NAMES CALL
                        ;       THIS ROUTINE WITH THE MACRO S$BITS.
                        ;       S$BITS IS IN THE FORMAT-
                        ;         S$BITS        POINTER,VALUE
                        ;           ON RETURN FROM S$BITS ROUTINE R0 CONTAINS THE ADDRESS
                        ;           OF THE BUFFER.
                        ;--
                        ;*****************************************************************************
                        ;
                        ;CALL BY S$BITS MACRO, NO PRINTING
012466                  BIT$::
                        ;SAVE SOME REGISTERS
012466  010246                  MOV     R2,-(SP)
012470  010346                  MOV     R3,-(SP)
012472  010446                  MOV     R4,-(SP)
012474  012503                  MOV     (R5)+,R3        ;GET POINTER OF FORMAT IN R3
012476  012737  177777  013226  MOV     #-1,STATU$      ;TELL ROUTINE THAT THE S$BITS MACRO CALLED
012504  000411                  BR      BUILD$          ;BRANCH OVER SETBITS STUFF
                        ;CALL BY SETBITS MACRO, PRINT LINE IF SOMETHING TO PRINT
012506                  BITS$::
012506  005037  013226          CLR     STATU$          ;TELL ROUTINE THAT THE SETBITS MACRO CALLED
                        ;SAVE SOME REGISTERS
012512  010246                  MOV     R2,-(SP)
012514  010346                  MOV     R3,-(SP)
012516  010446                  MOV     R4,-(SP)
012520  010502                  MOV     R5,R2           ;GET ADDRESS IN R2
012522  061505                  ADD     (R5),R5         ;ADD LENGTH OF DATA TO RETURN PC
012524  005722                  TST     (R2)+           ;SKIP PAST LENGTH
012526  012203                  MOV     (R2)+,R3        ;GET POINTER OF FORMAT IN R3
                        ;REGISTER USAGE
                        ;       R0 = LOOP COUNTER
                        ;       R1 = CONTAINS REGISTER CONTENTS
                        ;       R2 = POINTS TOWARD REGISTER NAME
```

```
                                        ;       R3 = POINTS TO DEFINITION FORMAT
                                        ;       R4 = POINTS TO BUFFER
                                        ;       R5 = CONTAINS RETURN ADDRESS
                                        ;NOW BUILD BIT NAME BUFFER
012530                                  BUILD$:
012530  012704  013015                          MOV     #MES2$,R4       ;POINT R4 TO BUFFER
012534  012700  000204                          MOV     #MES2$L,R0      ;LENGTH OF BUFFER TO CLEAR
                                        ;ZERO OUT THE BUFFER
012540  105024                          1$:     CLRB    (R4)+           ;CLEAR BYTE
012542  005300                                  DEC     R0
012544  001375                                  BNE     1$
012546  012704  013015                          MOV     #MES2$,R4       ;POINT R4 TO BUFFER
012552  012700  000020                          MOV     #16.,R0         ;NUMBER OF LOOPS TO LOOP COUNTER R1
012556  010137  013222                          MOV     R1,STORE$       ;SAVE ORIGINAL BIT PATTERN
012562  005037  013224                          CLR     NO$CHAR         ;CLEAR CHARACTER MOVED FLAG
012566  105037  013225                  10$:    CLRB    NO$CHAR+1       ;CLEAR BIT DEFINED AND MOVED FLAG
012572  006101                                  ROL     R1              ;CHECK BIT
012574  103410                                  BCS     20$             ;IF BITS SET, GO MOVE BIT NAME
                                        ;BIT WAS NOT SET, FIND NEXT COMMA
012576  122713  000000                  5$:     CMPB    #0,(R3)         ;END OF DEFINITION?
012602  001433                                  BEQ     40$             ;IF END, GO PRINT MESSAGE
012604  122713  000054                          CMPB    #',,(R3)        ;IS IT A COMMA?
012610  001425                                  BEQ     35$             ;IF COMMA, CONTINUE TO PROCESS
012612  005203                                  INC     R3              ;POINT TO NEXT CHARACTER IN DEFINITION
012614  000770                                  BR      5$              ;CHECK NEXT CHARACTER IF NOT A COMMA
                                        ;BIT WAS SET, MOVE BIT NAME IF THERE
012616                                  20$:
012616  122713  000000                          CMPB    #0,(R3)         ;IS IT END OF FORMAT?
012622  001425                                  BEQ     45$             ;IF END, GO PRINT MESSAGE
012624  122713  000054                          CMPB    #',,(R3)        ;IS IT A COMMA?
012630  001406                                  BEQ     30$             ;IF IT WAS A COMMA, GO
                                        ;SHOULD BE A VALID CHARACTER, MOVE TO PRINT BUFFER
012632  112324                                  MOVB    (R3)+,(R4)+     ;MOVE CHARACTER
012634  105237  013224                          INCB    NO$CHAR         ;INCREMENT CHARACTER FLAG
012640  105237  013225                          INCB    NO$CHAR+1       ;INCREMENT BIT DEFINED AND MOVED FLAG
012644  000764                                  BR      20$             ;GO PROCESS NEXT CHAR
                                        ;END OF CHARACTER MOVE, ADVANCE POINTER TO PUT COMMA AFTER NAME
012646                                  30$:
012646  105737  013225                          TSTB    NO$CHAR+1       ;WAS THE LAST BIT DEFINED
012652  001404                                  BEQ     35$             ;IF LAST BIT NOT DEFINED, BRANCH
012654  112724  000054                          MOVB    #',,(R4)+       ;PUT COMMA AFTER BIT NAME
012660  112724  000040                          MOVB    #40,(R4)+       ;PUT SPACE AFTER COMMA
012664  005203                          35$:    INC     R3              ;POINT TO NEXT CHARACTER IN DEFINITION
012666  005300                                  DEC     R0              ;DECREMENT LOOP COUNTER
012670  001336                                  BNE     10$             ;IF NOT DONE, GO PROCESS NEXT BIT
                                        ;NOW SET UP TO PRINT THE MESSAGE
012672                                  40$:
012672  005304                                  DEC     R4              ;POINT BACK TO LAST COMMA AND SPACE
012674  005304                                  DEC     R4
012676  112714  000000                  45$:    MOVB    #0,(R4)         ;PUT ZERO AS TERMINATOR IN BUFFER
012702  005000                                  CLR     R0              ;DEFAULT RETURN CODE = NOTHING PRINTED
012704  005737  013226                          TST     STATU$          ;SEE WHICH MACRO CALLED ROUTINE
012710  001020                                  BNE     50$             ;BRANCH IF S$BITS MACRO CALLED
012712  005737  013224                          TST     NO$CHAR         ;SEE IF NOTHING PUT IN PRINT BUFFER
012716  001415                                  BEQ     50$             ;IF NOTHING THERE, DON'T PRINT IT
012720                                          PRINTB  #MES1$,R2,STORE$,#MES2$         ;PRINT LINE
012720  012746  013015                          MOV     #MES2$,-(SP)
```

GLOBAL SUBROUTINES SECTION

```
        012724  013746  013222                 MOV     STORE$,-(SP)
        012730  010246                          MOV     R2,-(SP)
        012732  012746  012766                  MOV     @MES1$,-(SP)
        012736  012746  000004                  MOV     @4,-(SP)
        012742  010600                          MOV     SP,R0
        012744  104414                          TRAP    C$PNTB
        012746  062706  000012                  ADD     @12,SP
    887                             ;RESTORE AND GO BACK
    888 012752                      50$:
    889 012752  012700  013015                  MOV     @MES2$,R0       ;RETURN WITH R0 POINTING TO BUFFER
    890 012756  012604                          MOV     (SP)+,R4
    891 012760  012603                          MOV     (SP)+,R3
    892 012762  012602                          MOV     (SP)+,R2
    893 012764  000205                          RTS     R5
    894
    895 012766     045     116     061  MES1$:: .ASCIZ  /%N1%T%A = %O6%A; %T%N1/
    896 013015                          MES2$:: .BLKB   132.            ;SAVE ALL KINDS OF SPACE FOR MESSAGE LINE
    897             000204              MES2$L  = .-MES2$
    898                                         .EVEN
    899 013222  000000                  STORE$: .WORD   0               ;STORES ORIGINAL BIT PATTERN
    900 013224  000000                  NO$CHAR: .WORD  0               ;FLAG IF CHARACTERS WERE MOVED TO BUFFER
    901 013226  000000                  STATU$: .WORD   0               ;STORE WHICH MACRO CALLED
```

```
903 013230                              STARS
                                        ;;******************************************************************
904                                     ;**
905                                     ;       SUBROUTINE DUMP.
906                                     ; FUNCTIONAL DESCRIPTION:
907                                     ;    SUBROUTINE TO PRINT THE CONTENTS OF ALL THE REGISTERS
908                                     ;    IT USES THE PRINTX PRINT EXTENDED FORMAT AND CAN BE DISABLED
909                                     ;    BY THE OPERATOR BY /FLAG:IXE, INHIBIT EXTENDED ERROR REPORTS
910                                     ; INPUTS:
911                                     ;       NONE
912                                     ; OUTPUT:
913                                     ;       DEVICE REGISTER DUMP PRINTED IF EXTENDED ERROR REPORTS ENABLED
914                                     ; CALLING SEQUENCE:
915                                     ;       JSR     PC,DUMP
916 013230                              STARS
                                        ;;******************************************************************
917
918 013230                              DUMP::
919                                     ;FIRST CHECK FOR INHIBIT EXTENDED ERROR REPORTS FLAG SET
920 013230                                      RFLAGS  FLAG
    013230  104421                             TRAP    C$RFLA
    013232  010037  014170                      MOV     R0,FLAG
921 013236  032737  004000  014170              BIT     #BIT11,FLAG             ;IS INHIBIT FLAGS SET?
922 013244  001144                              BNE     EDUMP                   ;EXIT ROUTINE IF SET
923 013246  010146                              MOV     R1,-(SP)                ;SAVE CPU REGISTERS
924 013250  010246                              MOV     R2,-(SP)
925 013252  010346                              MOV     R3,-(SP)
926 013254  010446                              MOV     R4,-(SP)
927 013256  010546                              MOV     R5,-(SP)
928 013260  017746  167016                      MOV     @QCTL,-(SP)
929                                     ;FIRST READ THE REGISTERS
930 013264  052777  000020  167010              BIS     #QVREN,@QCTL            ;ENABLE Q-VIEW READ
931 013272  012701  000014                      MOV     #12.,R1                 ;NUMBER OF REGISTERS TO READ
932 013276  013702  002274                      MOV     QCMD,R2                 ;BASE REGISTER ADDRESS TO R2
933 013302  012703  002324                      MOV     #VALUES,R3              ;RECEIVING STORAGE
934 013306  022701  000007              E10$:   CMP     #7,R1                   ;IS IT QUBA REGISTER?
935 013312  001002                              BNE     11$
936 013314  000137  013720                      JMP     P1                      ;BRANCH IF QUBA
937 013320  022701  000002              11$:    CMP     #2,R1                   ;IS IT QVAD REGISTER?
938 013324  001002                              BNE     12$                     ;BRANCH IF QVAD
939 013326  000137  013772                      JMP     P2
940 013332  012223                      12$:    MOV     (R2)+,(R3)+             ;STORE REGISTER VALUE
941 013334  005301                      E5$:    DEC     R1                      ;DECREMENT LOOP COUNTER
942 013336  001363                              BNE     E10$                    ;IF NOT DONE, LOOP
943                                     ;END READ, NOW PRINT THE RESULTS
944 013340                                      PRINTX  #RP1                    ;PRINT FIRST LINE OF MESSAGE
    013340  012746  014050                      MOV     #RP1,-(SP)
    013344  012746  000001                      MOV     #1,-(SP)
    013350  010600                              MOV     SP,R0
    013352  104415                              TRAP    C$PNTX
    013354  062706  000004                      ADD     #4,SP
945 013360  012704  002377                      MOV     #NAMES-5,R4             ;POINT R4 TO REGISTER NAMES TABLE
946 013364  005002                              CLR     R2                      ;CLEAR POINTER REGISTER
947 013366  012703  000014                      MOV     #12.,R3                 ;LOAD R3 AS LOOP COUNTER
948 013372  062704  000005              E6$:    ADD     #5,R4                   ;POINT R4 TO NEXT REGISTER NAME
949 013376  016237  002274  002524              MOV     ADDRESSES(R2),ADD       ;GET ADDRESS OF REGISTER READY TO PRINT
950 013404  016237  002324  002526              MOV     VALUES(R2),VAL          ;GET VALUE OF REGISTER READY TO PRINT
```

GLOBAL SUBROUTINES SECTION

```
951 013412                              PRINTX  #RP2,R4,ADD,VAL         ;PRINT REGISTER NAME, ADDRESS, VALUE
    013412  013746  002526              MOV     VAL,(SP)
    013416  013746  002524              MOV     ADD,(SP)
    013422  010446                      MOV     R4,(SP)
    013424  012746  014140              MOV     #RP2,(SP)
    013430  012746  000004              MOV     #4,(SP)
    013434  010600                      MOV     SP,R0
    013436  104415                      TRAP    C$PNTX
    013440  062706  000012              ADD     #12,SP
952                             ;CHECK TO SEE IF ONE OF THE BIT DEFINED REGISTERS, IF YES, BRANCH
953 013444  022703  000014              CMP     #12.,R3                 ;IS IT CMD REG
954 013450  001443                      BEQ     CMDB
955 013452  022703  000013              CMP     #11.,R3                 ;IS IT VNT REG
956 013456  001446                      BEQ     VNTB
957 013460  022703  000012              CMP     #10.,R3                 ;IS IT INT REG
958 013464  001451                      BEQ     INTB
959 013466  022703  000011              CMP     #9.,R3                  ;IS IT CTL REG
960 013472  001454                      BEQ     CTLB
961 013474  022703  000010              CMP     #8.,R3                  ;IS IT DMA REG
962 013500  001457                      BEQ     DMAB
963 013502  022703  000003              CMP     #3.,R3                  ;IS IT MT1 REG
964 013506  001462                      BEQ     MT1B
965 013510  062702  000002      X$:     ADD     #2,R2                   ;POINT R2 TO NEXT REGISTER DATA
966 013514  005303                      DEC     R3                      ;DECREMENT LOOP COUNTER
967 013516  001325                      BNE     E6$                     ;GO DO NEXT REGISTER IF NOT DONE
968 013520                              PRINTX  #RP0                    ;INSERT LINE FEED CARRIAGE RETURN
    013520  012746  014044              MOV     #RP0,-(SP)
    013524  012746  000001              MOV     #1,-(SP)
    013530  010600                      MOV     SP,R0
    013532  104415                      TRAP    C$PNTX
    013534  062706  000004              ADD     #4,SP
969 013540  012677  166536              MOV     (SP)+,@QCTL
970 013544  012605                      MOV     (SP)+,R5
971 013546  012604                      MOV     (SP)+,R4
972 013550  012603                      MOV     (SP)+,R3                ;RESTORE REGISTERS
973 013552  012602                      MOV     (SP)+,R2
974 013554  012601                      MOV     (SP)+,R1
975 013556  000207              EDUMP:  RTS     PC                      ;RETURN
976
977                             ;PROCESS PRINTING SET BIT DEFINITIONS
978 013560                      CMDB:   S$BITS  CMDBIT,VAL
979 013572  000435                      BR      PRINTIT
980
981 013574                      VNTB:   S$BITS  VNTBIT,VAL
982 013606  000427                      BR      PRINTIT
983
984 013610                      INTB:   S$BITS  INTBIT,VAL
985 013622  000421                      BR      PRINTIT
986
987 013624                      CTLB:   S$BITS  CTLBIT,VAL
988 013636  000413                      BR      PRINTIT
989
990 013640                      DMAB:   S$BITS  DMABIT,VAL
991 013652  000405                      BR      PRINTIT
992
993 013654                      MT1B:   S$BITS  MT1BIT,VAL
994
```

```
                                        ;PRINT BIT DEFINITION LINE
013666                                  PRINTIT:
013666  010005                                  MOV     R0,R5                   ;MOVE ADDRESS TO R5
013670  105715                                  TSTB    (R5)                    ;IS IT A BLANK
013672  001706                                  BEQ     X$                      ;BRANCH IF NOTHING TO PRINT
013674                                          PRINTB  @RP4,R5                 ;PRINT LINE
013674  010546                                  MOV     R5,-(SP)
013676  012746  014165                         MOV     @RP4, (SP)
013702  012746  000002                         MOV     #2, (SP)
013706  010600                                  MOV     SP,R0
013710  104414                                  TRAP    C$PNTB
013712  062706  000006                         ADD     #6,SP
013716  000674                                  BR      X$                      ;RETURN TO MAINLINE

                                        ;PROCESS QUBA REGISTER (NOT NORMAL REGISTER READ
013720  017746  166372                P1:       MOV     @QMT1,-(SP)             ;SAVE CURRENT QMT1
013724  052777  000200  166364                BIS     #MIODIS,@QMT1           ;ENABLE INTERNAL LOOPBACK
013732  052777  000020  166356                BIS     #BIT4,@QMT1             ;SELECT UBAR FUNCTION
013740  017723  166350                        MOV     @QMT0,(R3)+             ;READ QUBA
013744  005722                                TST     (R2)+                   ;CORRECT R2
013746  042777  000200  166342                BIC     #MIODIS,@QMT1           ;DISABLE INTERNAL LOOPBACK
013754  042777  000020  166334                BIC     #BIT4,@QMT1             ;DESELECT UBAR FUNCTION
013762  012677  166330                        MOV     (SP)+,@QMT1             ;RESTORE QMT1
013766  000137  013334                        JMP     E5$                     ;BRANCH BACK

                                        ;PROCESS QVAD REGISTER READ (NOT NORMAL REGISTER READ)
013772  017746  166320                P2:       MOV     @QMT1,-(SP)             ;SAVE CURRENT QMT1
013776  052777  000200  166312                BIS     #MIODIS,@QMT1           ;ENABLE INTERNAL LOOPBACK
014004  052777  000002  166304                BIS     #MVADR,@QMT1            ;ENABLE ADDRESS REGISTER READ
014012  017723  166302                        MOV     @QVAD,(R3)+             ;READ ADDRESS REGISTER
014016  005722                                TST     (R2)+                   ;CORRECT R2
014020  042777  000200  166270                BIC     #MIODIS,@QMT1           ;DISABLE INTERNAL LOOPBACK
014026  042777  000002  166262                BIC     #MVADR,@QMT1            ;DISABLE ADDRESS REGISTER READ
014034  012677  166256                        MOV     (SP)+,@QMT1             ;RESTORE QMT1
014040  000137  013334                        JMP     E5$                     ;BRANCH BACK

                                        ;FORMATS USED IN REGISTER DUMP PRINTING

014044      045     116     061       RP0::   .ASCIZ  /#N1/                   ;LINE FEED CARRIAGE RETURN ONLY
014050      045     116     061       RP1::   .ASCIZ  /#N1#S7#AREGISTER DUMP#N1#ANAME#S4#AADDRESS#S4#ACONTENTS/
                                              .EVEN
014140      045     116     061       RP2::   .ASCIZ  /#N1#T#S5#O6#S5#O6#S3/
014165      045     124     000       RP4::   .ASCIZ  /#T/
                                              .EVEN
014170  000000                        FLAG:   .WORD   0                       ;STORES RETURNED FLAG SETTINGS
```

GLOBAL SUBROUTINES SECTION

```
                                        STARS
                                        ;;*****************************************************************
                                        ;**
                                        ; FUNCTIONAL DESCRIPTION:
                                        ;       SUBROUTINE TO PRINT OUT A TEXT OR HELP FILE
                                        ; INPUTS:
                                        ;       NONE
                                        ; OUTPUTS:
                                        ;       CONTENTS OF EITHER DRS.HLP OR DIAGNOSTIC DEVICE HELP FILE
                                        ;       PRINTED ON CONSOLE.
                                        ;       CONTENTS OF R1 AND R2 ARE DESTROYED.
                                        ; CALLING SEQUENCE:
                                        ;       USES MACRO HLPMAC; FORMAT IS HLPMAC 'FILENAME.EXT'
                                        ;       MACRO IS DEFINED AT END OF SUBROUTINE.
                                        ;       MACRO USES FORMAT JSR R5,HLP.
                                        ;--
                                        STARS
                                        ;;*****************************************************************

014172                                  $HLP::
014172  010537  014366                          MOV     R5,$STORE               ;SAVE RETURN ADDRESS
014176                                        OPEN    R5                      ;OPEN FILE NAME
014176  010500                                MOV     R5,R0
014200  104434                                TRAP    C$OPEN
                                        ;FIRST PRINT CR/LF
014202  012702  000015                        MOV     #15,R2                  ;CARRIAGE RETURN TO R2
014206  004537  014270                        JSR     R5,60$                  ;GO PRINT IT
014212  012702  000012                        MOV     #12,R2                  ;LINE FEED TO R2
014216  004537  014270                        JSR     R5,60$                  ;GO PRINT IT
014222  005001                                CLR     R1                      ;CLEAR POSITION COUNTER
014224                                  10$:  GETBYTE R2                      ;GET BYTE COUNT OF RECORD
014224  104426                                TRAP    C$GETB
014226  110002                                MOVB    R0,R2
014230                                        BNCOMPLETE 50$                  ;BRANCH IF END OF FILE
014230  103011                                BCC     50$
014232  022702  000000                        CMP     #0,R2                   ;END OF RECORDS?
014236  001406                                BEQ     50$                     ;BRANCH IF END
014240  022702  000011                  30$:  CMP     #11,R2                  ;IS IT TAB CHARACTER?
014244  001437                                BEQ     80$                     ;IF YES HANDLE TAB
014246  004537  014270                  40$:  JSR     R5,60$                  ;GO PRINT IT
014252  000764                                BR      10$                     ;GO PROCESS NEXT CHARACTER
                                        ;END OF FILE OR LAST RECORD
                                        50$:
014254                                        CLOSE                           ;CLOSE FILE
014254  104435                                TRAP    C$CLOS
014256  013705  014366                        MOV     $STORE,R5               ;RESTORE RETURN ADDRESS
014262  062705  000012                        ADD     #10.,R5                 ;SKIP PAST FILE NAME
014266  000205                                RTS     R5
                                        ;PRINT AND COUNT ROUTINE
014270  010237  014364                  60$:  MOV     R2,$MESLNE              ;GET CHARACTER TO PRINT IN R2
014274                                        PRINTB  #$HLPFOR                ;GO PRINT LINE
014274  012746  014362                        MOV     #$HLPFOR,-(SP)
014300  012746  000001                        MOV     #1,-(SP)
014304  010600                                MOV     SP,R0
014306  104414                                TRAP    C$PNTB
014310  062706  000004                        ADD     #4,SP
014314  005201                                INC     R1                      ;INCREMENT POSITION COUNTER
```

GLOBAL SUBROUTINES SECTION

```
1080 014316  022701  000010                CMP     #8.,R1          ;IS IT MAXIMUM?
1081 014322  001406                        BEQ     65$             ;IF MAXIMUM, BRANCH
1082 014324  022702  000012                CMP     #12,R2          ;IS IT A LINE FEED?
1083 014330  001403                        BEQ     65$             ;IF LINE FEED, BRANCH
1084 014332  022702  000015                CMP     #15,R2          ;IS IT A CARRIAGE RETURN?
1085 014336  001001                        BNE     70$             ;IF NOT, BRANCH
1086 014340  005001                  65$:  CLR     R1              ;CLEAR POSITION COUNTER
1087 014342  000205                  70$:  RTS     R5              ;FROM PRINT AND COUNT ROUTINE
1088                                 ;TAB ROUTINE
1089 014344                          80$:
1090 014344  012702  000040                MOV     #40,R2          ;MOVE SPACE CHARACTERS
1091 014350  004537  014270                JSR     R5,60$          ;GO PRINT SPACE
1092 014354  005701                        TST     R1              ;DONE YET?
1093 014356  001372                        BNE     80$             ;IF NOT DONE, AGAIN
1094 014360  000721                        BR      10$             ;RETURN FROM TAB ROUTINE
1095                                 ;
1096                                 ;STORAGE, FORMAT  AND MESSAGE LINE
1097                                 ;
1098 014362                          $HLPFOR::
1099 014362     045     101                 .ASCII  /%A/
1100 014364                          $MESLNE:
1101 014364     000                         .BYTE   0               ;RESERVED FOR MESSAGE LINE
1102 014365     000                         .BYTE   0
1103                                        .EVEN
1104 014366  000000                  $STORE: .WORD   0               ;STORES MAIN SUBROUTINE RETURN ADDRESS
```

GLOBAL SUBROUTINES SECTION

```
1106 014370                            STARS
                                       ;;*****************************************************************
1107                                   ;**
1108                                   ;       MMINIT
1109                                   ; FUNCTIONAL DESCRIPTION-
1110                                   ;       SUBROUTINE TO INITIALIZE MEMORY MANAGEMENT REGISTERS
1111                                   ; INPUTS-
1112                                   ;       NONE
1113                                   ; OUTPUT-
1114                                   ;       NONE
1115                                   ; CALLING SEQUENCE-
1116                                   ;       JSR     PC,MMINIT
1117                                   ;
1118                                   ;--
1119 014370                            STARS
                                       ;;*****************************************************************
1120
1121 014370                            MMINIT::
1122 014370  005001                            CLR     R1
1123 014372  012702  172300                    MOV     #KISDR0,R2              ;GET DESCRIPTOR REGISTER ADDRESS
1124 014376  012703  000007                    MOV     #7,R3                   ;DO 7 REGISTERS
1125 014402  012712  077406            1$:     MOV     #77406,(R2)             ;SET 4K, R/W UP
1126 014406  010162  000040                    MOV     R1,40(R2)               ;SET ADDRESS REGISTER
1127 014412  005722                            TST     (R2)+                   ;SET TO NEXT REGISTER ADDRESS
1128 014414  062701  000200                    ADD     #200,R1
1129 014420  005303                            DEC     R3                      ;DONE?
1130 014422  001367                            BNE     1$                      ;IF NOT DONE
1131 014424  012712  077406                    MOV     #77406,(R2)             ;SET I/O PAGE
1132 014430  012762  177600  000040            MOV     #177600,40(R2)
1133 014436  000207                            RTS     PC
```

GLOBAL SUBROUTINES SECTION

```
1135 014440                            STARS
                                       ;;*****************************************************************
1136                                   ;**
1137                                   ;               INTERRUPT SERVICE ROUTINE
1138                                   ;               ADDRESS TRAP SERVICE ROUTINE
1139                                   ;--
1140 014440                            STARS
                                       ;;*****************************************************************
1141
1142 014440                                    BGNSRV  TRAP4
     014440                            TRAP4::
1143 014440  005237  002522                    INC     TRPFLG                  ;SET TRAP FLAG
1144 014444                                    ENDSRV
     014444                            L10006:
     014444  000002                            RTI
1145
1146 014446                                    ENDMOD
1147
1148
```

GLOBAL SUBROUTINES SECTION

```
11                                      .ENABL  AMA
12                                .SBTTL  REPORT CODING SECTION
40
41 014446                               BGNMOD
42
43                                ;++
44                                ; THE REPORT CODING SECTION CONTAINS THE
45                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
46                                ;--
47
48 014446                               BGNRPT
   014446                         L$RPT::
49 014446  004737  013230                CALL    DUMP                    ;GO PRINT THE DUMP ROUTINE
61
62 014452                               EXIT    RPT
   014452  000167                       .WORD   J$JMP
   014454  000000                       .WORD   L10007-2-.
74
75                                      .EVEN
76
77 014456                               ENDRPT
   014456                         L10007:
   014456  104425                       TRAP    C$RPT
```

PROTECTION TABLE

```
                                .SBTTL  PROTECTION TABLE
                                ;++
                                ; THIS TABLE IS USED BY THE RUNTIME SERVICES
                                ; TO PROTECT THE LOAD MEDIA.
                                ;--

014460                                  BGNPROT
014460                          L$PROT::

014460  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
014462  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
014464  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

014466                                  ENDPROT
```

```
                                        .SBTTL  INITIALIZE SECTION

                                        ;++
                                        ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                        ; AT THE BEGINNING OF EACH PASS.
                                        ;--

 014466                                         BGNINIT
 014466                                 L$INIT::

 014466                                         SETPRI  #PRI00                  ;INITIALIZE WITH CPU ZERO PRIORITY
 014466  012700  000000                         MOV     #PRI00,R0
 014472  104441                                 TRAP    C$SPRI

 014474                                 INIT0:  READEF  #EF.CONTINUE            ; WAS THERE A CONTINUE COMMAND
 014474  012700  000036                         MOV     #EF.CONTINUE,R0
 014500  104447                                 TRAP    C$REFG
 014502                                         BNCOMPLETE INIT1                ;IF NO, CONTINUE
 014502  103002                                 BCC     INIT1
 014504  000137  015032                         JMP     ENDI                    ;YES, SKIP INITIALIZATION

 014510                                 INIT1:  READEF  #EF.NEW                 ;NEW PASS OR SUB PASS?
 014510  012700  000035                         MOV     #EF.NEW,R0
 014514  104447                                 TRAP    C$REFG
 014516                                         BNCOMPLETE NEXT                 ;IF NOT NEW PASS, SKIP SETUP
 014516  103036                                 BCC     NEXT

 014520  005737  002502                 INIT2:  TST     BRDGFLG                 ;HAS BRIDGE TYPE BEEN DETERMINED?
 014524  001030                                 BNE     INIT3                   ;IF NE YES
 014526  005237  002502                         INC     BRDGFLG                 ;SET FLAG
 014532  005037  002500                         CLR     BRIDGETYPE              ;CLEAR BRIDGE TYPE FLAG (Q-BRIDGE)
 014536  012702  003316                         MOV     #QBUS,R2                ;LOAD BRIDGE TYPE PTR (Q-BRIDGE)
 014542                                         READBUS                         ;DETERMINE BRIDGE TYPE
 014542  104407                                 TRAP    C$RDBU
 014544                                         BCOMPLETE 10$                   ;IF COMPLETE THEN Q-BRIDGE
 014544  103407                                 BCS     10$
 014546  012737  177777  002500                 MOV     #-1,BRIDGETYPE          ;SET BRIDGE TYPE FLAG (U-BRIDGE)
 014554  012702  003324                         MOV     #UBUS,R2                ;LOAD BRIDGE TYPE PTR (U-BRIDGE)
 014560  005037  172516                         CLR     MMR3                    ;DISABLE UNIBUS MAPPING         ;V C.0
 014564                                 10$:    PRINTB  #CONFIG,R2              ;IDENTIFY BRIDGE TYPE
 014564  010246                                 MOV     R2,-(SP)
 014566  012746  003244                         MOV     #CONFIG,-(SP)
 014572  012746  000002                         MOV     #2,-(SP)
 014576  010600                                 MOV     SP,R0
 014600  104414                                 TRAP    C$PNTB
 014602  062706  000006                         ADD     #6,SP

                                        ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                                        ; HELP FILE SECTION REMOVED.  TO REINSTALL, REMOVED COMMENTS
                                        ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
 014606                                 INIT3:
                                        ;       TST     HP1                     ;WAS HELP FILE PRINTED
                                        ;       BNE     1$                      ;BRANCH IF ALREADY PRINTED
                                        ;       INC     HP1                     ;SET HELP FILE QUESTION FLAG
                                        ;       GMANIL  HELP1,VAL,1,YES         ;ASK IF HELP FILE WANTED
                                        ;       CMP     #NO,VAL                 ;DO WE WANT TO PRINT HELP FILE?
                                        ;       BEQ     1$                      ;IF NO, BRANCH
```

INITIALIZE SECTION

```
                                    ;       HLPMAC  DRS.HLP                 ;PRINT OUT SUPERVISOR HELP FILE
                                    ;1$:    TST     HP2                     ;WAS HELP FILE PRINTED
                                    ;       BNE     SETUP                   ;BRANCH IF ALREADY PRINTED
                                    ;       INC     HP2                     ;SET HELP FILE QUESTION FLAG
                                    ;2$:    GMANIL  HELP2,VAL,1,YES         ;ASK IF DIAGNOSTIC HELP FILE WANTED
                                    ;       CMP     #NO,VAL                 ;DO WE WANT TO PRINT HELP FILE
                                    ;       BEQ     SETUP                   ;IF NO, BRANCH
                                    ;       HLPMAC  Z0791.HLP               ;PRINT OUT DIAGNOSTIC HELP FILE

014606  012737  177777  002504    SETUP:  MOV     #-1,LOGUNIT             ;INITIALIZE LOGICAL UNIT COUNTER
```

INITIALIZE SECTION

```
                                                NEXT:   INC     LOGUNIT                 ;POINT TO NEXT LOGICAL UNIT
                                                        CMP     LOGUNIT,L$UNIT          ;HAVE WE PASSED MAXIMUM
                                                        BEQ     ABORT                   ;GO IF MAXIMUM PASSED
                                                        GPHARD  LOGUNIT,PLOC            ;GET P-TABLE ADDRESS
014630  013700  002504                                  MOV     LOGUNIT,R0
014634  104442                                          TRAP    C$GPHRD
014636  010037  002506                                  MOV     R0,PLOC
                                                        BNCOMPLETE NEXT                 ;IF NOT AVAILABLE, GET NEXT UNIT
014642  103364                                          BCC     NEXT
014644  013737  002504  002074                          MOV     LOGUNIT,L$LUN           ;STORE CURRENT LOGICAL UNIT IN HEADER
                                                ;GET ALL HARDWARE TABLE ENTRIES IN STORAGE
014652  013700  002506                                  MOV     PLOC,R0                 ;PLACE HEADER POINTER IN R0
014656  016037  000000  002510                          MOV     0(R0),BASEADD           ;BASE ADDRESS OF BOARD
014664  016037  000002  002512                          MOV     2(R0),VECTOR            ;VECTOR ADDRESS OF BOARD
014672  016037  000004  002514                          MOV     4(R0),LOOPBK            ;LOOPBACK OPTION INSTALLED?
014700  016037  000006  002516                          MOV     6(R0),BRLEVEL           ;BR INTERRUPT LEVEL ON THE BOARD
                                                ;FILL IN BRIDGE INTERFACE REGISTER TABLE
014706  013701  002510                                  MOV     BASEADD,R1              ;GET BASE ADDRESS IN R1
014712  012702  002274                                  MOV     #QCMD,R2                ;POINT TO REGISTER TABLE ADDRESS
014716  012703  000014                                  MOV     #12.,R3                 ;# OF REGISTERS IN TABLE
014722  010122                                  10$:    MOV     R1,(R2)+                ;PLACE REGISTER ADDRESS IN TABLE
014724  062701  000002                                  ADD     #2,R1                   ;ADJUST R1 TO NEXT REGISTER ADDRESS
014730  005303                                          DEC     R3                      ;ARE WE DONE?
014732  001373                                          BNE     10$                     ;IF NOT DONE, GO AGAIN
                                                ;TABLE BUILT, NOW ESTABLISH INITIALIZE WORD FOR TEST (LOOPBACK YES OR NO INSTALLED)
014734  005037  002520                                  CLR     INITWORD                ;CLEAR STORAGE
014740  005737  002514                                  TST     LOOPBK                  ;IS LOOPBACK CABLE INSTALLED
014744  001003                                          BNE     20$                     ;IF LOOPBACK CABLE INSTALLED, BRANCH
014746  052737  000200  002520                          BIS     #MIODIS,INITWORD        ;SET LOOPBACK ENABLED (NO CABLE INSTALLED)
014754                                          20$:
                                                ;GET FREE MEMORY ADDRESS AND SIZE
014754                                                  MEMORY  FREMEM                  ;GET MEMORY ADDRESS
014754  104431                                          TRAP    C$MEM
014756  010037  002552                                  MOV     R0,FREMEM
014762  017737  165564  002550                          MOV     @FREMEM,MEMSIZ          ;GET FREE MEMORY SIZE
                                                ;CHECK MEMORY SIZE
014770  022737  002000  002120                          CMP     #2000,L$HIME            ;> 32K WORD MEMORY?             ;V C.0
014776  003004                                          BGT     22$                     ;IF NOT                         ;V C.0
015000  005037  177572                                  CLR     MMR0                    ;DISABLE MEM. MGT.              ;V C.0
015004  005037  172516                                  CLR     MMR3                    ;DISABLE UMR                    ;V C.0
015010  022737  007577  002120              22$:        CMP     #7577,L$HIME            ;COMPARE 128K TO MEMORY SIZE
015016  100003                                          BPL     25$                     ;OK IF SIZE <= 128K
015020  012737  007577  002120                          MOV     #7577,L$HIME            ;RESET MEMORY SIZE TO 128K
015026  000401                                  25$:    BR      ENDI                    ;SKIP ABORT INSTRUCTION
015030                                          ABORT:  DOCLN                           ;DO CLEANUP AND ABORT THE PASS
015030  104444                                          TRAP    C$DCLN
015032                                          ENDI:   EXIT    INIT                    ;THAT'S ALL TO INITIALIZE PASS
015032  104432                                          TRAP    C$EXIT
015034  000002                                          .WORD   L10011-.


                                                ;       EXIT    INIT


                                                        .EVEN
```

INITIALIZE SECTION

```
    242 015036                              ENDINIT
        015036                      L10011:
        015036  104411                     TRAP    C$INIT
```

AUTODROP SECTION

```
                                .SBTTL  AUTODROP SECTION
                                ;**
                                ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
                                ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
                                ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
                                ; DROPPED FROM TESTING.
                                ;-
015040                                  BGNAUTO
015040                          L$AUTO::
015040  000240                            NOP             ; NO ON LINE SIGNAL PROVIDED TO USER FOR AUTODROP
015042                                  ENDAUTO
015042                          L10012:
015042  104461                          TRAP    C$AUTO
```

CLEANUP CODING SECTION

```
                                        .SBTTL  CLEANUP CODING SECTION

                                        ;++
                                        ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                        ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
                                        ;
                                        ; THE CLEANUP CODE WILL SET THE TRAP CATCHER, CLEAR MIODIS,
                                        ; RESET THE TRAP CATCHER, AND EXIT.
                                        ;--

015044                                          BGNCLN
015044                                  L$CLEAN::

015044                                          SETVEC  #4,#BAD,#PRIO7          ; SET FOR BAD ADDRESS TRAP
015044  012746  000340                          MOV     #PRIO7,-(SP)
015050  012746  015112                          MOV     #BAD,-(SP)
015054  012746  000004                          MOV     #4,-(SP)
015060  012746  000003                          MOV     #3,-(SP)
015064  104437                                  TRAP    C$SVEC
015066  062706  000010                          ADD     #10,SP
015072  042777  000200  165216                  BIC     #MIODIS,@MT1            ; CLEAR MAINTENANCE LOOP BACK BIT
015100                                          CLRVEC  #4                      ; RESET VECTOR
015100  012700  000004                          MOV     #4,R0
015104  104436                                  TRAP    C$CVEC
015106                                          EXIT    CLN
015106  104432                                  TRAP    C$EXIT
015110  000010                                  .WORD   L10013 .

015112  000002                  BAD:    RTI

015114                                          EXIT    CLN
015114  104432                                  TRAP    C$EXIT
015116  000002                                  .WORD   L10013 .

015120                                          ENDCLN
015120                          L10013:
015120  104412                                  TRAP    C$CLEAN
```

DROP UNIT SECTION

```
                                .SBTTL  DROP UNIT SECTION

                                ;**
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO NO LONGER BE TESTED.
                                ;--

015122                                  BGNDU
015122                          L$DU::

015122  000240                          NOP             ;NO SPECIAL DROP UNIT CODING NEEDED

015124                                  EXIT    DU
015124  000167                          .WORD   J$JMP
015126  000000                          .WORD   L10014 2-.

                                        .EVEN

015130                                  ENDDU
015130                          L10014:
015130  104453                          TRAP    C$DU
```

ADD UNIT SECTION

```
                                .SBTTL  ADD UNIT SECTION

                                ;**
                                ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
                                ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
                                ;  TO THE TEST CYCLE.
                                ; -

015132                                  BGNAU
015132                          L$AU::

015132  000240                            NOP             ;NO SPECIAL ADD UNIT CODING NEEDED

015134                                  EXIT    AU
015134  000167                          .WORD   J$JMP
015136  000000                          .WORD   L10015-2-.

                                        .EVEN

015140                                  ENDAU
015140                          L10015:
015140  104452                          TRAP    C$AU

015142                                  ENDMOD
```

ADD UNIT SECTION

```
                                        .ENABL  AMA
                                .DSABL GBL
```

ADD UNIT SECTION

```
                                        BGNMOD
                                .SBTTL  TEST1: RESET
                                STATST  <RESET>
015142      040     122     105 T1MSG:: .ASCIZ  / RESET/

                                ;*****************************************************************************
                                ;++
                                ;TEST DESCRIPTION-
                                ; CHECKS REGISTER ADDRESSING
                                ; CHECKS REGISTER BITS FOR RESET
                                ;TEST STEPS-
                                ; EXECUTE RESET INSTRUCTION
                                ; READ ALL REGISTERS
                                ; CHECK ALL VALUES
                                ;
                                ;--
                                ;*****************************************************************************

                                        BGNTST
                                T1::
                                        CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
015152  104421                          TRAP    C$RFLA
015154  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
015160  001012                          BNE     64$                     ; BRANCH IF NOT SET
015162  012746  015142                  MOV     @T1MSG,-(SP)
015166  012746  002270                  MOV     @PNTMSG,-(SP)
015172  012746  000002                  MOV     @2,-(SP)
015176  010600                          MOV     SP,R0
015200  104417                          TRAP    C$PNTF
015202  062706  000006                  ADD     @6,SP
015206  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
015212                                  SETVEC  @4,@TRAP4,@PRI07        ; SET FOR BAD ADDRESS TRAP
015212  012746  000340                  MOV     @PRI07,-(SP)
015216  012746  014440                  MOV     @TRAP4,-(SP)
015222  012746  000004                  MOV     @4,-(SP)
015226  012746  000003                  MOV     @3,-(SP)
015232  104437                          TRAP    C$SVEC
015234  062706  000010                  ADD     @10,SP
015240                          5$:     BREAK                           ; ALLOW CONTROL C
015240  104422                          TRAP    C$BRK
015242  005301                          DEC     R1                      ; DECREMENT DELAY COUNTER
015244  001375                          BNE     5$                      ; LOOP IF NOT DONE
015246  004537  016002          6$:     JSR     R5,BRESET               ; GO DO BUS RESET
                                ;
                                ; CHECK REGISTER ADDRESSING
                                ;
015252  005037  002522                  CLR     TRPFLG                  ; CLEAR TRAP FLAG
015256  012701  177777                  MOV     @-1,R1                  ; COUNTER IN R1
015262  013702  002274                  MOV     @CMD,R2                 ; REGISTER TABLE BASE TO R2
015266  005201                  10$:    INC     R1                      ; INCREMENT COUNTER
015270  022701  000014                  CMP     @12.,R1                 ; DONE YES?
015274  001423                          BEQ     CONT                    ; BRANCH IF DONE WITH CHECK
015276  012203                          MOV     (R2)+,R3                ; READ REGISTER
015300  005737  002522                  TST     TRPFLG                  ; ANY TRAPS?
015304  001770                          BEQ     10$                     ; BRANCH IF NO TRAPS
                                ;
                                ; MUST HAVE BEEN BAD ADDRESS
```

TEST1: RESET

```
                                        ;
015306  005037  002522                  CLR     TRPFLG                  ; CLEAR TRAP INDICATOR
015312  010237  002524                  MOV     R2,ADD                  ; PUT OFFENDING ADDRESS IN MEM
015316  162737  000002  002524          SUB     #2,ADD                  ; CORRECT FOR AUTO INCREMENT
015324                                  ERRDF   1,MSG2,ERR2             ; DEVICE FATAL ERROR MESSAGE
015324  104455                          TRAP    C$ERDF
015326  000001                          .WORD   1
015330  004571                          .WORD   MSG2
015332  011702                          .WORD   ERR2
015334                                  CLRVEC  #4                      ; RESET VECTOR
015334  012700  000004                  MOV     #4,R0
015340  104436                          TRAP    C$CVEC
015342                                  DOCLN                           ; EXIT PASS IF BAD ADDRESS
015342  104444                          TRAP    C$DCLN

                                        ;
                                        ; NOW TEST ALL THE REGISTERS
                                        ;
                                        CONT:
                                                BGNSUB
                                        T1.1:
                                                TRAP    C$BSUB
                                                INLOOP                  ; CHECK FOR ERROR LOOP
015346  104420                                  TRAP    C$INLP
                                                BNCOMPLETE      11$     ; BRANCH IF NOT ERROR LOOP
015350  103002                                  BCC     11$
015352  004537  016002                          JSR     R5,BRESET       ; IF ERROR LOOP, BRANCH
                                        11$:
                                        ;
                                        ; TEST QCMD FOR EXPECTED RESULTS
                                        ;
015356  017737  164712  002526                  MOV     @QCMD,VAL       ; READ QCMD REGISTER
015364  042737  177476  002526                  BIC     #177476,VAL     ; MASK OUT UNDETERMINED BITS
015372  012737  000000  002530                  MOV     #0,EXP          ; ALL ZEROS TO EXPECTED
015400  023737  002530  002526                  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
015406  001406                                  BEQ     30$             ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION 1
                                        ;
015410  012701  000000                          MOV     #0,R1           ; LOAD R1 WITH REGISTER NUMBER
015414                                          ERRHRD  1,MSG3,ERR4     ; UNEXPECTED RESULT, TELL OPERATOR
015414  104456                                  TRAP    C$ERHRD
015416  000001                                  .WORD   1
015420  004632                                  .WORD   MSG3
015422  011776                                  .WORD   ERR4
                                        30$:
                                                ENDSUB
                                        L10017:
015424  104403                                  TRAP    C$ESUB
                                                BGNSUB
                                        T1.2:
015426  104402                                  TRAP    C$BSUB
                                                INLOOP                  ; CHECK FOR ERROR LOOP
015430  104420                                  TRAP    C$INLP
                                                BNCOMPLETE      31$     ; BRANCH IF NOT ERROR LOOP
015432  103002                                  BCC     31$
015434  004537  016002                          JSR     R5,BRESET       ; IF ERROR LOOP, BRANCH
```

TEST1: RESET

```
    120 015440                                  31$:
    121                                         ;
    122                                         ; TEST QVNT REGISTER
    123                                         ;
    124 015440  017737  164632  002526                MOV     @QVNT,VAL               ; READ QVNT REGISTER
    125 015446  042737  177476  002526                BIC     #177476,VAL             ; MASK OUT UNDETERMINED BITS
    126 015454  012737  000000  002530                MOV     #0,EXP                  ; ALL ZEROS TO EXPECTED
    127 015462  023737  002530  002526                CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
    128 015470  001406                                BEQ     40$                     ; BRANCH IF OK
    129                                         ;
    130                                         ; ERROR CONDITION 2
    131                                         ;
    132 015472  012701  000001                        MOV     #1,R1                   ; LOAD R1 WITH REGISTER NUMBER
    133 015476                                        ERRHRD  2,MSG3,ERR4             ; UNEXPECTED RESULT, TELL OPERATOR
        015476  104456                                TRAP    C$ERHRD
        015500  000002                                .WORD   2
        015502  004632                                .WORD   MSG3
        015504  011776                                .WORD   ERR4
    134 015506                                  40$:
    135 015506                                        ENDSUB
        015506                                  L10020:
        015506  104403                                TRAP    C$ESUB
    136 015510                                        BGNSUB
        015510                                  T1.3:
        015510  104402                                TRAP    C$BSUB
    137 015512                                        INLOOP                          ; CHECK FOR ERROR LOOP
        015512  104420                                TRAP    C$INLP
    138 015514                                        BNCOMPLETE      41$             ; BRANCH IF NOT ERROR LOOP
        015514  103002                                BCC     41$
    139 015516  004537  016002                        JSR     R5,BRESET               ; IF ERROR LOOP, BRANCH
    140 015522                                  41$:
    141                                         ;
    142                                         ; TEST QINT REGISTER
    143                                         ;
    144 015522  017737  164552  002526                MOV     @QINT,VAL               ; READ QINT REGISTER
    145 015530  012737  001000  002530                MOV     #1000,EXP               ; EXPECTED VALUE TO EXP
    146 015536  023737  002530  002526                CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
    147 015544  001406                                BEQ     50$                     ; BRANCH IF OK
    148                                         ;
    149                                         ; ERROR CONDITION 3
    150                                         ;
    151 015546  012701  000002                        MOV     #2,R1                   ; LOAD R1 WITH REGISTER NUMBER
    152 015552                                        ERRHRD  3,MSG3,ERR4             ; UNEXPECTED RESULT, TELL OPERATOR
        015552  104456                                TRAP    C$ERHRD
        015554  000003                                .WORD   3
        015556  004632                                .WORD   MSG3
        015560  011776                                .WORD   ERR4
    153 015562                                  50$:
    154 015562                                        ENDSUB
        015562                                  L10021:
        015562  104403                                TRAP    C$ESUB
    155 015564                                        BGNSUB
        015564                                  T1.4:
        015564  104402                                TRAP    C$BSUB
    156 015566                                        INLOOP                          ; CHECK FOR ERROR LOOP
        015566  104420                                TRAP    C$INLP
    157 015570                                        BNCOMPLETE      51$             ; BRANCH IF NOT ERROR LOOP
```

TEST1: RESET

```
     015570  103002                          BCC     51$
158  015572  004537  016002                  JSR     R5,BRESET               ; IF ERROR LOOP, BRANCH
159  015576                          51$:
160                                  ;
161                                  ; TEST QCTL REGISTER
162                                  ;
163  015576  017737  164500  002526          MOV     @QCTL,VAL               ; READ QCTL REGISTER
164  015604  012737  000000  002530          MOV     #0,EXP                  ; ALL ZEROS TO EXP
165  015612  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
166  015620  001406                          BEQ     60$                     ; BRANCH IF OK
167                                  ;
168                                  ; ERROR CONDITION 4
169                                  ;
170  015622  012701  000003                  MOV     #3,R1                   ; LOAD R1 WITH REGISTER NUMBER
171  015626                                  ERRHRD  4,MSG3,ERR4             ; UNEXPECTED RESULT, TELL OPERATOR
     015626  104456                          TRAP    C$ERHRD
     015630  000004                          .WORD   4
     015632  004632                          .WORD   MSG3
     015634  011776                          .WORD   ERR4
172  015636                          60$:
173  015636                                  ENDSUB
     015636                          L10022:
     015636  104403                          TRAP    C$ESUB
174  015640                                  BGNSUB
     015640                          T1.5:
     015640  104402                          TRAP    C$BSUB
175  015642                                  INLOOP                          ; CHECK FOR ERROR LOOP
     015642  104420                          TRAP    C$INLP
176  015644                                  BNCOMPLETE      61$             ; BRANCH IF NOT ERROR LOOP
     015644  103002                          BCC     61$
177  015646  004537  016002                  JSR     R5,BRESET               ; IF ERROR LOOP, BRANCH
178  015652                          61$:
179                                  ;
180                                  ; TEST QDMA REGISTER
181                                  ;
182  015652  017737  164426  002526          MOV     @QDMA,VAL               ; READ QDMA REGISTER
183  015660  012737  000200  002530          MOV     #200,EXP                ; EXPECTED VALUE TO EXP
184  015666  02373,  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
185  015674  001406                          BEQ     70$                     ; BRANCH IF OK
186                                  ;
187                                  ; ERROR CONDITION 5
188                                  ;
189  015676  012701  000004                  MOV     #4,R1                   ; LOAD R1 WITH REGISTER NUMBER
190  015702                                  ERRHRD  5,MSG3,ERR4             ; UNEXPECTED RESULT, TELL OPERATOR
     015702  104456                          TRAP    C$ERHRD
     015704  000005                          .WORD   5
     015706  004632                          .WORD   MSG3
     015710  011776                          .WORD   ERR4
191  015712                          70$:
192  015712                                  ENDSUB
     015712                          L10023:
     015712  104403                          TRAP    C$ESUB
193                                  ;
194                                  ; QBA REGISTER NOT CHECKED
195                                  ;
196
197                                  ;QWC REGISTER NOT CHECKED
```

TEST1: RESET

```
                                        ;
                                        ; QMTO REGISTER NOT CHECKED (TRI-STATE)
                                        ;
                                                BGNSUB
                                        T1.6:
015714  104402                                  TRAP    C$BSUB
                                                INLOOP                          ; CHECK FOR ERROR LOOP
015716  104420                                  TRAP    C$INLP
                                                BNCOMPLETE      71$             ; BRANCH IF NOT ERROR LOOP
015720  103002                                  BCC     71$
016002                                  JSR     R5,BRESET               ; IF ERROR LOOP, BRANCH
                                        71$:
                                        ;
                                        ; TEST QMT1 REGISTER
                                        ;
017737  164364  002526                          MOV     @QMT1,VAL               ; READ QMT1 REGISTER
012737  000000  002530                          MOV     #0,EXP                  ; ALL ZEROS TO EXPECTED
023737  002530  002526                          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
001406                                  BEQ     80$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION 6
                                        ;
012701  000011                                  MOV     #9.,R1                  ; LOAD R1 WITH REGISTER NUMBER
                                                ERRHRD  6,MSG3,ERR4             ; UNEXPECTED RESULT, TELL OPERATOR
104456                                  TRAP    C$ERHRD
000006                                  .WORD   6
004632                                  .WORD   MSG3
011776                                  .WORD   ERR4
                                        80$:
                                                CLRVEC  #4                      ; CLEAR BAD ADDRESS TRAP
012700  000004                                  MOV     #4,R0
104436                                  TRAP    C$CVEC
                                                ENDSUB
                                        L10024:
104403                                  TRAP    C$ESUB

                                        ;
                                        ; QVAD AND QVDA REGISTERS ARE NOT CHECKED
                                        ;
                                                EXIT TST
104432                                  TRAP    C$EXIT
000020                                  .WORD   L10016-.
                                        ;
                                        ;SUBROUTINE TO DO A BUS RESET
                                        ; CALLING ROUTINE-
                                        ;  JSR  R5,BRESET
                                        ;
                                        BRESET:
012701  003000                                  MOV     #3000,R1                ; GET READY FOR DELAY LOOP
                                        10$:    BREAK                           ; ALLOW CONTROL C
104422                                  TRAP    C$BRK
005301                                  DEC     R1                      ; DECREMENT DELAY COUNTER
001375                                  BNE     10$                     ; LOOP IF NOT DONE
000005                                  RESET                           ; DO BUS RESET
000205                                  RTS     R5                      ; RETURN FROM SUBROUTINE
```

TEST1: RESET

```
    240
    241 016020                              ENDTST
        016020                      L10016:
        016020  104401                      TRAP    C$ETST
```

```
                                        .SBTTL  TEST2: CHECK QMT1 REGISTER
                                        STATST  <CHECK QMT1 REGISTER>
016022  040     103     110     T2MSG:: .ASCIZ  / CHECK QMT1 REGISTER/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS R/W BITS
                                        ;TEST STEPS-
                                        ; LOAD 1'S IN QMT1
                                        ; READ AND CHECK
                                        ; LOAD 0'S
                                        ; READ AND CHECK
                                        ;
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
                                        T2::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
016050  104421                                  TRAP    C$RFLA
016052  022700  001000                          CMP     #PNT,R0                 ; IS PNT FLAG SET?
016056  001012                                  BNE     64$                     ; BRANCH IF NOT SET
016060  012746  016022                          MOV     #T2MSG,-(SP)
016064  012746  002270                          MOV     #PNTMSG,-(SP)
016070  012746  000002                          MOV     #2,-(SP)
016074  010600                                  MOV     SP,R0
016076  104417                                  TRAP    C$PNTF
016100  062706  000006                          ADD     #6,SP
016104  005037  002532          64$:    CLR     ERROR1                          ; CLEAR ERROR COUNTER
                                                BGNSUB
                                        T2.1:
016110  104402                                  TRAP    C$BSUB
                                        ;
                                        ; FIRST CHECK WITH ONE'S
                                        ;
016112  012737  000236  002530                  MOV     #236,EXP                ; MOVE ALL 1'S TO EXPECTED
016120  013777  002530  164170                  MOV     EXP,@QMT1               ; MOVE EXPECTED TO QMT1
016126  017737  164164  002526                  MOV     @QMT1,VAL               ; MOVE RESULT TO VALUE STORAGE
016134  042737  177541  002526                  BIC     #^C236,VAL              ; CLEAR UNDETERMINED BITS
016142  023737  002526  002530                  CMP     VAL,EXP                 ; IS THAT WHAT IS REALLY THERE?
016150  001406                                  BEQ     1$                      ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION 1
                                        ;
016152  012701  000011                          MOV     #9.,R1                  ; REGISTER NUMBER FOR MESSAGE
016156                                          ERRHRD  1,MSG4,ERR4             ; GO REPORT ERROR TO OPERATOR
016156  104456                                  TRAP    C$ERHRD
016160  000001                                  .WORD   1
016162  004672                                  .WORD   MSG4
016164  011776                                  .WORD   ERR4
016166                                  1$:
                                                ENDSUB
016166                                  L10026:
016166  104403                                  TRAP    C$ESUB
016170                                          BGNSUB
016170                                  T2.2:
```

TEST2: CHECK QMT1 REGISTER

```
                016170  104402                          TRAP    C$BSUB
279                                             ;
280                                             ; NOW CHECK WITH ZERO'S
281                                             ;
282             016172  012737  000000  002530          MOV     #0,EXP                  ; MOVE ALL 0'S TO EXPECTED
283             016200  013777  002530  164110          MOV     EXP,@QMT1               ; MOVE EXPECTED TO QMT1
284             016206  017737  164104  002526          MOV     @QMT1,VAL               ; MOVE RESULT TO VALUE STORAGE
285             016214  042737  177541  002526          BIC     #^C236,VAL              ; CLEAR UNDETERMINED BITS
286             016222  023737  002526  002530          CMP     VAL,EXP                 ; IS THAT WHAT IS REALLY THERE?
287             016230  001411                          BEQ     2$                      ; BRANCH IF OK
288                                             ;
289                                             ; ERROR CONDITION 2
290                                             ;
291             016232  012701  000011                  MOV     #9.,R1                  ; REGISTER # FOR MESSAGE
292             016236  012737  000000  002530          MOV     #0,EXP                  ; EXPECTED RESULT FOR MESSAGE
293             016244                                  ERRHRD  2,MSG5,ERR4             ; GO REPORT ERROR TO OPERATOR
                016244  104456                          TRAP    C$ERHRD
                016246  000002                          .WORD   2
                016250  004731                          .WORD   MSG5
                016252  011776                          .WORD   ERR4
294             016254                          2$:
295             016254                                  ENDSUB
                016254                          L10027:
                016254  104403                          TRAP    C$ESUB
296             016256                                  ENDTST
                016256                          L10025:
                016256  104401                          TRAP    C$ETST
297
```

```
                        .SBTTL  TEST3: CHECK QCLR RESET
                        STATST  <CHECK QCLR RESET>
016260  040     103     110     T3MSG:: .ASCIZ  / CHECK QCLR RESET/
                        ;*****************************************************************************
                        ;**
                        ;TEST DESCRIPTION
                        ; CHECKS QCLR REGISTER BIT RESET
                        ; USES QHALT   QCLR LOOP-BACK
                        ;TEST STEPS-
                        ; LOAD ALL R/W BITS THAT ARE CLEARABLE (EXPEPT TMRENB)
                        ; SET QMT1 MQHLT BIT
                        ; READ ALL REGISTERS THAT WERE LOADED
                        ; CHECK FOR CLEAR (SEE EXPECTED VALUES IN TABLE EXP3)
                        ;
                        ;-
                        ;*****************************************************************************

016302                          BGNTST
016302                  T3::
016302                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
016302  104421                  TRAP    C$RFLA
016304  022700  001000          CMP     @PNT,R0                 ; IS PNT FLAG SET?
016310  001012                  BNE     64$                     ; BRANCH IF NOT SET
016312  012746  016260          MOV     @T3MSG,-(SP)
016316  012746  002270          MOV     @PNTMSG,-(SP)
016322  012746  000002          MOV     #2, (SP)
016326  010600                  MOV     SP,R0
016330  104417                  TRAP    C$PNTF
016332  062706  000006          ADD     #6,SP
016336  005037  002532  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
016342  005077  163732          CLR     @QINT                   ; CLEAR UPWRFL BIT FROM PREVIOUS        ; V C.0
                                                                ; TEST TO PREVENT UNEXPECTED    ; V C.0
                                                                ; INTERRUPT.                    ; V C.0
                        ;
                        ; LOAD ALL R/W BITS THAT ARE CLEARABLE
                        ;
016346  012777  177401  163720          MOV     #177401,@QCMD           ; SET ALL R/W BITS IN QCMD(NO IE);V C.0
016354  012777  177401  163714          MOV     #177401,@QVNT           ; SET ALL R/W BITS IN QVNT(NO IE);V C.0
016362  012777  000014  163712          MOV     #000014,@QCTL           ; SET ALL R/W BITS IN QCTL
016370  012777  037562  163706          MOV     #037562,@QDMA           ; SET ALL R/W BITS IN QDMA
016376  012777  000100  163674          MOV     #000100,@QINT           ; SET ALL R/W BITS IN QINT
016404  052777  000100  163662          BIS     #PRSIE,@QCMD            ; NOW ENABLE INTERRUPTS         ; V C.0
016412  052777  000100  163656          BIS     #ACKIE,@QVNT            ;                               ; V C.0
016420  012777  000236  163670          MOV     #000236,@QMT1           ;                               ; V C.0
016426  052777  000100  163662          BIS     #MQHALT,@QMT1           ; RESET REGISTERS
                        ;
                        ; TEST QCMD REGISTER
                        ;
016434  017737  163634  002526          MOV     @QCMD,VAL               ; GET VALUE IN VAL
016442  042737  177477  002526          BIC     #^C300,VAL              ; CLEAR UNTESTED BITS
016450  012737  000000  002530          MOV     #0,EXP                  ; EXPECTED IN EXP
016456  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
016464  001406                  BEQ     1$                      ; BRANCH IF OK
                        ;
                        ; ERROR CONDITION 1
                        ;
```

TEST3: CHECK QCLR RESET

```
344 016466  012701  000000                  MOV     #0,R1                   ; LOAD R1 WITH REGISTER NUMBER
345 016472                                  ERRMRD  1,MSG6,ERR4             ; REPORT ERROR TO OPERATOR
    016472  104456                          TRAP    C$ERMRD
    016474  000001                          .WORD   1
    016476  004771                          .WORD   MSG6
    016500  011776                          .WORD   ERR4
346 016502                              1$:
347                                     ;
348                                     ; TEST QVNT REGISTER
349                                     ;
350 016502  017737  163570  002526          MOV     @QVNT,VAL               ; GET VALUE IN VAL
351 016510  042737  177477  002526          BIC     #^C300,VAL              ; CLEAR UNTESTED BITS
352 016516  012737  000000  002530          MOV     #0,EXP                  ; EXPECTED IN EXP
353 016524  023737  002530  002526          CMP     EXP,VAL                 ; GET EXPECTED IN VAL
354 016532  001406                          BEQ     2$                      ; IF OK, BRANCH
355                                     ;
356                                     ; ERROR CONDITION 2
357                                     ;
358 016534  012701  000001                  MOV     #1,R1                   ; LOAD R1 WITH REGISTER NUMBER
359 016540                                  ERRMRD  2,MSG6,ERR4             ; REPORT ERROR TO OPERATOR
    016540  104456                          TRAP    C$ERMRD
    016542  000002                          .WORD   2
    016544  004771                          .WORD   MSG6
    016546  011776                          .WORD   ERR4
360 016550                              2$:
361                                     ;
362                                     ; TEST QINT REGISTER
363                                     ;
364 016550  017737  163524  002526          MOV     @QINT,VAL               ; GET VALUE IN VAL
365 016556  012737  001000  002530          MOV     #1000,EXP               ; EXPECTED VALUE TO EXP
366 016564  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
367 016572  001406                          BEQ     3$                      ; IF OK, BRANCH
368                                     ;
369                                     ; ERROR CONDITION 3
370                                     ;
371 016574  012701  000002                  MOV     #2,R1                   ; LOAD R1 WITH REGISTER NUMBER
372 016600                                  ERRMRD  3,MSG6,ERR4             ; REPORT ERROR TO OPERATOR
    016600  104456                          TRAP    C$ERMRD
    016602  000003                          .WORD   3
    016604  004771                          .WORD   MSG6
    016606  011776                          .WORD   ERR4
373 016610                              3$:
374                                     ;
375                                     ; TEST QCTL REGISTER
376                                     ;
377 016610  017737  163466  002526          MOV     @QCTL,VAL               ; GET VALUE IN VAL
378 016616  012737  000000  002530          MOV     #0,EXP                  ; ALL ZEROS TO EXPECTED
379 016624  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
380 016632  001406                          BEQ     4$                      ; IF OK, BRANCH
381                                     ;
382                                     ; ERROR CONDITION 4
383                                     ;
384 016634  012701  000003                  MOV     #3,R1                   ; LOAD R1 WITH REGISTER NUMBER
385 016640                                  ERRMRD  4,MSG6,ERR4             ; REPORT ERROR TO OPERATOR
    016640  104456                          TRAP    C$ERMRD
    016642  000004                          .WORD   4
    016644  004771                          .WORD   MSG6
```

TEST3: CHECK QCLR RESET

```
     016646  011776                          .WORD   ERR4
386  016650                          4$:
387                                  ;
388                                  ; TEST QDMA REGISTER
389                                  ;
390  016650  017737  163430  002526          MOV     @QDMA,VAL               ; GET VALUE IN VAL
391  016656  012737  000200  002530          MOV     #200,EXP                ; EXPECTED VALUE TO EXP
392  016664  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
393  016672  001406                          BEQ     5$                      ; IF OK, BRANCH
394                                  ;
395                                  ; ERROR CONDITION 5
396                                  ;
397  016674  012701  000004                  MOV     #4,R1                   ; LOAD R1 WITH REGISTER NUMBER
398  016700                          ERRHRD  5,MSG6,ERR4             ; REPORT ERROR TO OPERATOR
     016700  104456                  TRAP    C$ERHRD
     016702  000005                  .WORD   5
     016704  004771                  .WORD   MSG6
     016706  011776                  .WORD   ERR4
399  016710                          5$:
400                                  ;
401                                  ; QUBA REGISTER NOT TESTED
402                                  ;
403
404                                  ;QBA REGISTER NOT TESTED
405
406                                  ;
407                                  ; QWC REGISTER NOT TESTED
408                                  ;
409
410                                  ;QMTO REGISTER NOT TESTED
411
412                                  ;
413                                  ; TEST QMT1 REGISTER
414                                  ;
415  016710  017737  163402  002526          MOV     @QMT1,VAL               ; GET VALUE IN VAL
416  016716  012737  000000  002530          MOV     #0,EXP                  ; ALL ZEROS TO EXPECTED
417  016724  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
418  016732  001406                          BEQ     6$                      ; IF OK, BRANCH
419                                  ;
420                                  ; ERROR CONDITION 6
421                                  ;
422  016734  012701  000011                  MOV     #9.,R1                  ; LOAD R1 WITH REGISTER NUMBER
423  016740                          ERRHRD  6,MSG6,ERR4             ; REPORT ERROR TO OPERATOR
     016740  104456                  TRAP    C$ERHRD
     016742  000006                  .WORD   6
     016744  004771                  .WORD   MSG6
     016746  011776                  .WORD   ERR4
424  016750                          6$:
425
426                                  ;
427                                  ; QVAD AND QVDA REGISTERS ARE NOT CHECKED
428                                  ;
429
430  016750                          ENDTST
     016750                          L10030:
     016750  104401                  TRAP    C$ETST
```

TEST4: CHECK QCMD LO BITS (MOVING 1)

```
                                        .SBTTL  TEST4: CHECK QCMD LO BITS (MOVING 1)
                                       STATST  <CHECK QCMD LO BITS (MOVING 1)>
016752  040     103     110            T4MSG:: .ASCIZ  / CHECK QCMD LO BITS (MOVING 1)/
                                       ;*****************************************************************************
                                       ;**
                                       ;TEST DESCRIPTION-
                                       ; CHECKS THAT BITS 1-5 CAN BE LOADED
                                       ; USES LDUCMD - LDQEVENT LOOP-BACK
                                       ;TEST STEPS-
                                       ; SET QMT1 FUNC = 2 (SELQCMD)
                                       ; LOAD MOVING BIT WORD IN QMTO
                                       ; READ QCMD, CHECK BITS 1-5
                                       ; REPEAT FOR ALL 5 BITS
                                       ;
                                       ;--
                                       ;*****************************************************************************

017012                                         BGNTST
017012                                 T4::
017012                                         CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
017012  104421                                 TRAP    C$RFLA
017014  022700  001000                         CMP     @PNT,R0                 ; IS PNT FLAG SET?
017020  001012                                 BNE     64$                     ; BRANCH IF NOT SET
017022  012746  016752                         MOV     @T4MSG,-(SP)
017026  012746  002270                         MOV     @PNTMSG,-(SP)
017032  012746  000002                         MOV     #2,-(SP)
017036  010600                                 MOV     SP,R0
017040  104417                                 TRAP    C$PNTF
017042  062706  000006                         ADD     #6,SP
017046  005037  002532                 64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
017052                                         INIT
                                       ;INITIALIZE DEVICE
017052  012777  000100  163236                 MOV     @#QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
017060  013777  002520  163230                 MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
017066  005077  163206                         CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
017072  005037  002532                         CLR     ERROR1                  ;CLEAR ERROR FLAG
017076  005037  002532                         CLR     ERROR1
017102  012701  000000                         MOV     #0,R1                   ; REGISTER NUMBER TO R1
017106  012702  000005                         MOV     #5.,R2                  ; NUMBER OF BITS TO CHECK
017112  012737  000002  002530                 MOV     #BIT1,EXP               ; FIRST DATA PATTERN TO CHECK
017120  052777  000010  163170                 BIS     #SELQCMD,@QMT1          ; SELECT QCMD MODE IN QMT1
                                       ;
                                       ; TEST BITS 1-5 WITH MOVING ONE
                                       ;
017126                                         BGNSUB
017126                                 T4.1:
017126  104402                                 TRAP    C$BSUB
017130                                 10$:    BGNSEG
017130  104404                                 TRAP    C$BSEG
017132  013777  002530  163154                 MOV     EXP,@QMTO               ; MOVE EXPECTED TO REGISTER QMTO
017140  017737  163130  002526                 MOV     @QCMD,VAL               ; READ QCMD AND MOVE TO VALUE
017146  042737  177701  002526                 BIC     #177701,VAL             ; MASK OUT UNWANTED BITS
017154  023737  002530  002526                 CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
017162  001414                                 BEQ     20$                     ; IF OK, BRANCH
                                       ;
                                       ; ERROR CONDITION
```

TEST4: CHECK QCMD LO BITS (MOVING 1)

```
469
470 017164  005737  002532          ;
                                    TST     ERROR1                  ; SEE IF FIRST ERROR
471 017170  001005                  BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
472 017172                          ERRHRD  1,MSG7,ERR7             ; TELL OPERATOR ERROR
    017172  104456                  TRAP    C$ERHRD
    017174  000001                  .WORD   1
    017176  005012                  .WORD   MSG7
    017200  012246                  .WORD   ERR7
473 017202  000402                  BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
474 017204  004737  012366  15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
475 017210  005237  002532  18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
476                                 ;
477                                 ; END ERROR CONDITION
478                                 ;
479 017214                  20$:    ENDSEG
    017214                  10000$:
    017214  104405                  TRAP    C$ESEG
480 017216  006337  002530          ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
481 017222  005302                  DEC     R2                      ; DECREMENT LOOP COUNTER
482 017224  001341                  BNE     10$                     ; IF NOT DONE, BRANCH BACK
483 017226                          ENDSUB
    017226                  L10032:
    017226  104403                  TRAP    C$ESUB
484 017230  005737  002532          TST     ERROR1                  ; ANY ERRORS
485 017234  001410                  BEQ     500$                    ; BRANCH IF NO ERRORS
486 017236                          PRINTX  #FORO                   ; LINE FEED
    017236  012746  003333          MOV     #FORO,-(SP)
    017242  012746  000001          MOV     #1,-(SP)
    017246  010600                  MOV     SP,R0
    017250  104415                  TRAP    C$PNTX
    017252  062706  000004          ADD     #4,SP
487 017256          500$:
488 017256                          ENDTST
    017256                  L10031:
    017256  104401                  TRAP    C$ETST
```

TEST5: CHECK QCMD LOW BITS (MOVING 0)

```
                                        .SBTTL  TEST5: CHECK QCMD LOW BITS (MOVING 0)
                                        STATST  <CHECK QCMD LOW BITS (MOVING 0)>
017260  040     103     110            T5MSG:: .ASCIZ  / CHECK QCMD LOW BITS (MOVING 0)/
                                        ;*************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT BITS 1-5 CAN BE LOADED
                                        ; USES LDUCMD - LDQEVENT LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 2 (SELQCMD)
                                        ; LOAD MOVING BIT WORD IN QMT0
                                        ; READ QCMD, CHECK BITS 1-5
                                        ; REPEAT FOR ALL 5 BITS
                                        ;
                                        ;--
                                        ;*************************************************************************

017320                                  T5::    BGNTST
017320
017320                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
017320  104421                                  TRAP    C$RFLA
017322  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
017326  001012                                  BNE     64$             ; BRANCH IF NOT SET
017330  012746  017260                          MOV     @T5MSG,-(SP)
017334  012746  002270                          MOV     @PNTMSG,-(SP)
017340  012746  000002                          MOV     #2,-(SP)
017344  010600                                  MOV     SP,R0
017346  104417                                  TRAP    C$PNTF
017350  062706  000006                          ADD     #6,SP
017354  005037  002532            64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
017360                                          INIT
                                        ;INITIALIZE DEVICE
017360  012777  000100  162730                  MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
017366  013777  002520  162722                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
017374  005077  162700                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
017400  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
017404  012701  000000                          MOV     #0,R1           ; REGISTER NUMBER TO R1
017410  012702  000005                          MOV     #5.,R2          ; NUMBER OF BITS TO CHECK
017414  012703  000002                          MOV     #BIT1,R3        ; INITIAL DATA PATTERN TO R3
017420  010337  002530                          MOV     R3,EXP          ; MOVE BIT PATTERN TO EXPECTED
017424  005137  002530                          COM     EXP             ; COMPLEMENT FOR MOVING 0
017430  052777  000010  162660                  BIS     #SELQCMD,@QMT1  ; SELECT QCMD MODE IN QMT1
                                        ;
                                        ; TEST BITS 1-5 WITH MOVING ZERO
                                        ;
017436                                          BGNSUB
017436                                  T5.1:
017436  104402                                  TRAP    C$BSUB
017440                                  10$:    BGNSEG
017440  104404                                  TRAP    C$BSEG
017442  042737  177701  002530                  BIC     #177701,EXP     ; CLEAR UNTESTED BITS
017450  013777  002530  162636                  MOV     EXP,@QMT0       ; MOVE EXPECTED TO REGISTER QMT0
017456  017737  162612  002526                  MOV     @QCMD,VAL       ; READ QCMD AND MOVE TO VALUE
017464  042737  177701  002526                  BIC     #177701,VAL     ; MASK OUT UNDETERMINED BITS
017472  023737  002530  002526                  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
017500  001414                                  BEQ     20$             ; IF OK, BRANCH
```

TEST5: CHECK QCMD LOW BITS (MOVING 0)

```
                                        ;
                                        ; ERROR CONDITION
                                        ;
017502  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
017506  001005                          BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
017510                                  ERRHRD  1,MSG8,ERR7             ; TELL OPERATOR ERROR
017510  104456                          TRAP    C$ERHRD
017512  000001                          .WORD   1
017514  005054                          .WORD   MSG8
017516  012246                          .WORD   ERR7
017520  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
017522  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
017526  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
                                        ;
                                        ; END ERROR CONDITION
                                        ;
017532                          20$:    ENDSEG
017532                          100000$:
017532  104405                          TRAP    C$ESEG
017534  006303                          ASL     R3                      ; SHIFT BIT FOR NEXT TEST
017536  010337  002530                  MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
017542  005137  002530                  COM     EXP                     ; COMPLIMENT FOR MOVING 0
017546  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
017550  001333                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
017552                                  ENDSUB
017552                          L10034:
017552  104403                          TRAP    C$ESUB
017554  005737  002532                  TST     ERROR1                  ; ANY ERRORS
017560  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
017562                                  PRINTX  #FORO                   ; LINE FEED
017562  012746  003333                  MOV     #FORO,-(SP)
017566  012746  000001                  MOV     #1,-(SP)
017572  010600                          MOV     SP,R0
017574  104415                          TRAP    C$PNTX
017576  062706  000004                  ADD     #4,SP
017602                          500$:
017602                                  ENDTST
017602                          L10033:
017602  104401                          TRAP    C$ETST
```

TEST6: CHECK QCMD HI BITS (MOVING 1)

```
                                        .SBTTL  TEST6: CHECK QCMD HI BITS (MOVING 1)
017604                                  STATST  <CHECK QCMD HI BITS (MOVING 1)>
017604  040   103   110                 T6MSG:: .ASCIZ  / CHECK QCMD HI BITS (MOVING 1)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT BITS 8-15 CAN BE LOADED
                                        ;TEST STEPS-
                                        ; LOAD MOVING BIT WORD IN QCMD
                                        ; READ QCMD, CHECK BITS 8-15
                                        ; REPEAT FOR ALL 8 BITS
                                        ;
                                        ;-
                                        ;*****************************************************************************
017644                                          BGNTST
017644                                  T6::
017644                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
017644  104421                                  TRAP    C$RFLA
017646  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
017652  001012                                  BNE     64$             ; BRANCH IF NOT SET
017654  012746  017604                          MOV     @T6MSG,-(SP)
017660  012746  002270                          MOV     @PNTMSG,-(SP)
017664  012746  000002                          MOV     #2,-(SP)
017670  010600                                  MOV     SP,R0
017672  104417                                  TRAP    C$PNTF
017674  062706  000006                          ADD     #6,SP
017700  005037  002532                  64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
017704                                          INIT
                                        ;INITIALIZE DEVICE
017704  012777  000100  162404                  MOV     #MQHALT,@QMT1   ;INITIALIZE Q-BRIDGE
017712  013777  002520  162376                  MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
017720  005077  162354                          CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
017724  005037  002532                          CLR     ERROR1          ;CLEAR ERROR FLAG
017730  012701  000000                          MOV     #0,R1           ; REGISTER NUMBER TO R1
017734  012702  000010                          MOV     #8.,R2          ; NUMBER OF BITS TO CHECK
017740  012737  000400  002530                  MOV     #BIT8,EXP       ; FIRST DATA PATTERN TO CHECK
                                        ;
                                        ; TEST BITS 8-15 W' ' MOVING ONE
                                        ;
017746                                          BGNSUB
017746                                  T6.1:
017746  104402                                  TRAP    C$BSUB
017750                                  10$:    BGNSEG
017750  104404                                  TRAP    C$BSEG
017752  013777  002530  162314                  MOV     EXP,@QCMD       ; MOVE EXPECTED TO REGISTER QCMD
017760  017737  162310  002526                  MOV     @QCMD,VAL       ; READ QCMD AND MOVE TO VALUE
017766  042737  000377  002526                  BIC     #377,VAL        ; MASK OUT UNTESTED BITS
017774  023737  002530  002526                  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
020002  001414                                  BEQ     20$             ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
020004  005737  002532                          TST     ERROR1          ; SEE IF FIRST ERROR
020010  001005                                  BNE     15$             ; SKIP HEADER IF ALREADY PRINTED
020012                                          ERRHRD  1,MSG9,ERR7     ; TELL OPERATOR ERROR
```

TEST6: CHECK QCMD HI BITS (MOVING 1)

```
                020012  104456                  TRAP    C$ERHRD
                020014  000001                  .WORD   1
                020016  005116                  .WORD   MSG9
                020020  012246                  .WORD   ERR7
            589 020022  000402                  BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
            590 020024  004737  012366      15$:    CALL    DETAIL              ; PRIN` DETAIL LINE
            591 020030  005237  002532      18$:    INC     ERROR1              ; INCREMENT ERROR COUNT
            592                             ;
            593                             ; END ERROR CONDITION
            594                             ;
            595 020034                      20$:    ENDSEG
                020034                      10000$:
                020034  104405                  TRAP    C$ESEG
            596 020036  006337  002530              ASL     EXP                 ; SHIFT BIT FOR NEXT TEST
            597 020042  005302                      DEC     R2                  ; DECREMENT LOOP COUNTER
            598 020044  001341                      BNE     10$                 ; IF NOT DONE, BRANCH BACK
            599 020046                              ENDSUB
                020046                      L10036:
                020046  104403                  TRAP    C$ESUB
            600 020050  005737  002532              TST     ERROR1              ; ANY ERRORS
            601 020054  001410                      BEQ     500$                ; BRANCH IF NO ERRORS
            602 020056                              PRINTX  #FORO               ; LINE FEED
                020056  012746  003333              MOV     #FORO,-(SP)
                020062  012746  000001              MOV     #1,-(SP)
                020066  010600                      MOV     SP,R0
                020070  104415                      TRAP    C$PNTX
                020072  062706  000004              ADD     #4,SP
            603 020076                      500$:
            604 020076                              ENDTST
                020076                      L10035:
                020076  104401                  TRAP    C$ETST
```

```
                                        .SBTTL  TEST7: CHECK QCMD HI BITS (MOVING 0)
                                        STATST  <CHECK QCMD HI BITS (MOVING 0)>
020100  040  103  110                   T7MSG:: .ASCIZ  / CHECK QCMD HI BITS (MOVING 0)/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT BITS 8-15 CAN BE LOADED
                                        ;TEST STEPS-
                                        ; LOAD MOVING BIT WORD IN QCMD
                                        ; READ QCMD, CHECK BITS 8-15
                                        ; REPEAT FOR ALL 8 BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************
020140                                  T7::    BGNTST
020140                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
020140  104421                                 TRAP    C$RFLA
020142  022700  001000                         CMP     @PNT,R0                 ; IS PNT FLAG SET?
020146  001012                                 BNE     64$                     ; BRANCH IF NOT SET
020150  012746  020100                         MOV     @T7MSG,-(SP)
020154  012746  002270                         MOV     @PNTMSG,-(SP)
020160  012746  000002                         MOV     #2,-(SP)
020164  010600                                 MOV     SP,R0
020166  104417                                 TRAP    C$PNTF
020170  062706  000006                         ADD     #6,SP
020174  005037  002532                 64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
020200                                          INIT
                                        ;INITIALIZE DEVICE
020200  012777  000100  162110                 MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
020206  013777  002520  162102                 MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
020214  005077  162060                         CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
020220  005037  002532                         CLR     ERROR1                  ;CLEAR ERROR FLAG
020224  012701  000000                         MOV     #0,R1                   ; REGISTER NUMBER TO R1
020230  012702  000010                         MOV     #8.,R2                  ; NUMBER OF BITS TO CHECK
020234  012703  000400                         MOV     #BIT8,R3                ; INITIAL DATA PATTERN TO R3
020240  010337  002530                         MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
020244  005137  002530                         COM     EXP                     ; COMPLEMENT FOR MOVING 0
                                        ;
                                        ; TEST BITS 8-15 WITH MOVING ZERO
                                        ;
020250                                          BGNSUB
020250                                  T7.1:
020250  104402                                 TRAP    C$BSUB
020252                                  10$:    BGNSEG
020252  104404                                 TRAP    C$BSEG
020254  042737  000377  002530                 BIC     #377,EXP                ; MASK OUT UNTESTED BITS
020262  013777  002530  162004                 MOV     EXP,@QCMD               ; MOVE EXPECTED TO REGISTER QCMD
020270  017737  162000  002526                 MOV     @QCMD,VAL               ; READ QCMD AND MOVE TO VALUE
020276  042737  000377  002526                 BIC     #377,VAL                ; MASK OUT UNTESTED BITS
020304  023737  002530  002526                 CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
020312  001414                                 BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
```

TEST7: CHECK QCMD HI BITS (MOVING 0)

```
    643 020314  005737  002532          TST     ERROR1                  ; SEE IF FIRST ERROR
    644 020320  001005                  BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
    645 020322                          ERRHRD  1,MSG10,ERR7            ; TELL OPERATOR ERROR
        020322  104456                  TRAP    C$ERHRD
        020324  000001                  .WORD   1
        020326  005161                  .WORD   MSG10
        020330  012246                  .WORD   ERR7
    646 020332  000402                  BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
    647 020334  004737  012366  15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
    648 020340  005237  002532  18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
    649                                 ;
    650                                 ; END ERROR CONDITION
    651                                 ;
    652 020344                  20$:    ENDSEG
        020344                  10000$:
        020344  104405                  TRAP    C$ESEG
    653 020346  006303                  ASL     R3                      ; SHIFT BIT FOR NEXT TEST
    654 020350  010337  002530          MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
    655 020354  005137  002530          COM     EXP                     ; COMPLIMENT FOR MOVING 0
    656 020360  005302                  DEC     R2                      ; DECREMENT LOOP COUNTER
    657 020362  001333                  BNE     10$                     ; IF NOT DONE, BRANCH BACK
    658 020364                          ENDSUB
        020364                  L10040:
        020364  104403                  TRAP    C$ESUB
    659 020366  005737  002532          TST     ERROR1                  ; ANY ERRORS
    660 020372  001410                  BEQ     500$                    ; BRANCH IF NO ERRORS
    661 020374                          PRINTX  #FORO                   ; LINE FEED
        020374  012746  003333          MOV     #FORO,-(SP)
        020400  012746  000001          MOV     #1,-(SP)
        020404  010600                  MOV     SP,R0
        020406  104415                  TRAP    C$PNTX
        020410  062706  000004          ADD     #4,SP
    662 020414                  500$:
    663 020414                          ENDTST
        020414                  L10037:
        020414  104401                  TRAP    C$ETST
```

TEST8: CHECK QVNT LO BITS (MOVING 1)

```
                                        .SBTTL  TEST8: CHECK QVNT LO BITS (MOVING 1)
020416                                  STATST  <CHECK QVNT LO BITS (MOVING 1)>
020416    040    103    110             T8MSG:: .ASCIZ  / CHECK QVNT LO BITS (MOVING 1)/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT BITS 1-5 CAN BE LOADED
                                        ; USES LDUCMD - LDQEVENT LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 1 (SELQVNT)
                                        ; LOAD MOVING BIT WORD IN QMT0
                                        ; READ QVNT, CHECK BITS 1-5
                                        ; REPEAT FOR ALL 5 BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************
020456                                          BGNTST
020456                                  T8::
020456                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
020456  104421                                  TRAP    C$RFLA
020460  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
020464  001012                                  BNE     64$             ; BRANCH IF NOT SET
020466  012746  020416                          MOV     @T8MSG,-(SP)
020472  012746  002270                          MOV     @PNTMSG,-(SP)
020476  012746  000002                          MOV     #2,-(SP)
020502  010600                                  MOV     SP,R0
020504  104417                                  TRAP    C$PNTF
020506  062706  000006                          ADD     #6,SP
020512  005037  002532                  64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
020516                                          INIT
                                        ;INITIALIZE DEVICE
020516  012777  000100  161572                  MOV     #MQHALT,@QMT1   ;INITIALIZE Q-BRIDGE
020524  013777  002520  161564                  MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
020532  005077  161542                          CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
020536  005037  002532                          CLR     ERROR1          ;CLEAR ERROR FLAG
020542  012701  000001                          MOV     #1,R1           ; REGISTER NUMBER TO R1
020546  012702  000005                          MOV     #5.,R2          ; NUMBER OF BITS TO CHECK
020552  012737  000002  002530                  MOV     #BIT1,EXP       ; FIRST DATA PATTERN TO CHECK
020560  052777  000004  161530                  BIS     #SELQEVENT,@QMT1        ; SELECT QVNT MODE IN QMT1
                                        ;
                                        ; TEST BITS 1-5 WITH MOVING ONE
                                        ;
020566                                          BGNSUB
020566                                  T8.1:
020566  104402                                  TRAP    C$BSUB
020570                                  10$:    BGNSEG
020570  104404                                  TRAP    C$BSEG
020572  013777  002530  161514                  MOV     EXP,@QMT0       ; MOVE EXPECTED TO REGISTER QMT0
020600  017737  161472  002526                  MOV     @QVNT,VAL       ; READ QVNT AND MOVE TO VALUE
020606  042737  177701  002526                  BIC     #177701,VAL     ; MASK OUT UNTESTED BITS
020614  023737  002530  002526                  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
020622  001414                                  BEQ     20$             ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
```

TEST8: CHECK QVNT LO BITS (MOVING 1)

```
                                        TST     ERROR1                  ; SEE IF FIRST ERROR
                                        BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
                                        ERRHRD  1,MSG11,ERR7            ; TELL OPERATOR ERROR
020632  104456                          TRAP    C$ERHRD
020634  000001                          .WORD   1
020636  005224                          .WORD   MSG11
020640  012246                          .WORD   ERR7
020642  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
020644  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
020650  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
                                ;
                                ; END ERROR CONDITION
                                ;
020654                          20$:    ENDSEG
020654                          10000$:
020654  104405                          TRAP    C$ESEG
020656  006337  002530                  ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
020662  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
020664  001341                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
020666                                  ENDSUB
020666                          L10042:
020666  104403                          TRAP    C$ESUB
020670  005737  002532                  TST     ERROR1                  ; ANY ERRORS
020674  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
020676                                  PRINTX  #FORO                   ; LINE FEED
020676  012746  003333                  MOV     #FORO,-(SP)
020702  012746  000001                  MOV     #1,-(SP)
020706  010600                          MOV     SP,R0
020710  104415                          TRAP    C$PNTX
020712  062706  000004                  ADD     #4,SP
020716                          500$:
020716                                  ENDTST
020716                          L10041:
020716  104401                          TRAP    C$ETST
```

```
722                                     .SBTTL  TEST9: CHECK QVNT LOW BITS (MOVING 0)
723 020720                              STATST  <CHECK QVNT LOW BITS (MOVING 0)>
    020720      040     103     110     T9MSG:: .ASCIZ  / CHECK QVNT LOW BITS (MOVING 0)/
724
725                                     ;*****************************************************************************
726                                     ;**
727                                     ;TEST DESCRIPTION-
728                                     ; CHECKS THAT BITS 1 5 CAN BE LOADED
729                                     ; USES LDUCMD - LDQEVENT LOOP-BACK
730                                     ;TEST STEPS-
731                                     ; SET QMT1 FUNC = 1 (SELQVNT)
732                                     ; LOAD MOVING BIT WORD IN QMT0
733                                     ; READ QVNT, CHECK BITS 1-5
734                                     ; REPEAT FOR ALL 5 BITS
735                                     ;
736                                     ;--
737                                     ;*****************************************************************************
738
739 020760                                       BGNTST
    020760                              T9::
740 020760                                       CKPNT                          ; PRINT OUT TEST TITLE IF PNT SET
    020760  104421                              TRAP    C$RFLA
    020762  022700  001000                      CMP     @PNT,R0                ; IS PNT FLAG SET?
    020766  001012                              BNE     64$                    ; BRANCH IF NOT SET
    020770  012746  020720                      MOV     @T9MSG,-(SP)
    020774  012746  002270                      MOV     @PNTMSG,-(SP)
    021000  012746  000002                      MOV     #2,-(SP)
    021004  010600                              MOV     SP,R0
    021006  104417                              TRAP    C$PNTF
    021010  062706  000006                      ADD     #6,SP
    021014  005037  002532              64$:    CLR     ERROR1                 ; CLEAR ERROR COUNTER
741 021020                                       INIT
                                        ;INITIALIZE DEVICE
    021020  012777  000100  161270              MOV     #QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
    021026  013777  002520  161262              MOV     INITWORD,@QMT1         ;SET UP LOOPBACK OPTION
    021034  005077  161240                      CLR     @QINT                  ;CLEAR QINT AFTER LOOPBACK ENABLED
    021040  005037  002532                      CLR     ERROR1                 ;CLEAR ERROR FLAG
742
743 021044  012701  000001                      MOV     #1,R1                  ; REGISTER NUMBER TO R1
744
745 021050  012702  000005                      MOV     #5.,R2                 ; NUMBER OF BITS TO CHECK
746 021054  012703  000002                      MOV     #BIT1,R3               ; INITIAL DATA PATTERN TO R3
747 021060  010337  002530                      MOV     R3,EXP                 ; MOVE BIT PATTERN TO EXPECTED
748 021064  005137  002530                      COM     EXP                    ; COMPLEMENT FOR MOVING 0
749 021070  052777  000004  161220              BIS     #SELQEVENT,@QMT1       ; SELECT QVNT MODE IN QMT1
750                                     ;
751                                     ; TEST BITS 1-5 WITH MOVING ZERO
752                                     ;
753 021076                                       BGNSUB
    021076                              T9.1:
    021076  104402                              TRAP    C$BSUB
754 021100                              10$:     BGNSEG
    021100  104404                              TRAP    C$BSEG
755 021102  042737  177701  002530              BIC     #177701,EXP            ; MASK OUT UNTESTED BITS
756 021110  013777  002530  161176              MOV     EXP,@QMT0              ; MOVE EXPECTED TO REGISTER QMT0
757 021116  017737  161154  002526              MOV     @QVNT,VAL              ; READ QVNT AND MOVE TO VALUE
758 021124  042737  177701  002526              BIC     #177701,VAL            ; MASK OUT UNTESTED BITS
```

TEST9: CHECK QVNT LOW BITS (MOVING 0)

```
759 021132  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
760 021140  001414                           BEQ     20$                     ; IF OK, BRANCH
761                                  ;
762                                  ; ERROR CONDITION
763                                  ;
764 021142  005737  002532                   TST     ERROR1                  ; SEE IF FIRST ERROR
765 021146  001005                           BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
766 021150                                   ERRHRD  1,MSG12,ERR7            ; TELL OPERATOR ERROR
    021150  104456                           TRAP    C$ERHRD
    021152  000001                           .WORD   1
    021154  005266                           .WORD   MSG12
    021156  012246                           .WORD   ERR7
767 021160  000402                           BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
768 021162  004737  012366           15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
769 021166  005237  002532           18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
770                                  ;
771                                  ; END ERROR CONDITION
772                                  ;
773 021172                           20$:    ENDSEG
    021172                           100008:
    021172  104405                           TRAP    C$ESEG
774 021174  006303                           ASL     R3                      ; SHIFT BIT FOR NEXT TEST
775 021176  010337  002530                   MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
776 021202  005137  002530                   COM     EXP                     ; COMPLIMENT FOR MOVING 0
777 021206  005302                           DEC     R2                      ; DECREMENT LOOP COUNTER
778 021210  001333                           BNE     10$                     ; IF NOT DONE, BRANCH BACK
779 021212                                   ENDSUB
    021212                           L10044:
    021212  104403                           TRAP    C$ESUB
780 021214  005737  002532                   TST     ERROR1                  ; ANY ERRORS
781 021220  001410                           BEQ     500$                    ; BRANCH IF NO ERRORS
782 021222                                   PRINTX  #CRLF                   ; LINE FEED
    021222  012746  003333                   MOV     #CRLF,-(SP)
    021226  012746  000001                   MOV     #1,(SP)
    021232  010600                           MOV     SP,R0
    021234  104415                           TRAP    C$PNTX
    021236  062706  000004                   ADD     #4,SP
783 021242                           500$:
784 021242                                   ENDTST
    021242                           L10043:
    021242  104401                           TRAP    C$ETST
```

TEST10: CHECK QVNT HI BITS (MOVING 1)

```
                                        .SBTTL  TEST10: CHECK QVNT HI BITS (MOVING 1)
                                        STATST  <CHECK QVNT HI BITS (MOVING 1)>
021244  040     103     110             T10MSG::        .ASCIZ  / CHECK QVNT HI BITS (MOVING 1)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT BITS 8-15 CAN BE LOADED
                                        ;TEST STEPS-
                                        ; LOAD MOVING BIT WORD IN QVNT
                                        ; READ QVNT, CHECK BITS 8-15
                                        ; REPEAT FOR ALL 8 BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

021304                                  BGNTST
021304                                  T10:.
021304                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
021304  104421                          TRAP    C$RFLA
021306  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
021312  001012                          BNE     64$                     ; BRANCH IF NOT SET
021314  012746  021244                  MOV     @T10MSG,-(SP)
021320  012746  002270                  MOV     @PNTMSG,-(SP)
021324  012746  000002                  MOV     #2,-(SP)
021330  010600                          MOV     SP,R0
021332  104417                          TRAP    C$PNTF
021334  062706  000006                  ADD     #6,SP
021340  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
021344                                  INIT
                                        ;INITIALIZE DEVICE
021344  012777  000100  160744          MOV     #^MQHALT,@QMT1          ;INITIALIZE Q-BRIDGE
021352  013777  002520  160736          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
021360  005077  160714                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
021364  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
021370  012701  000001                  MOV     #1,R1                   ; REGISTER NUMBER TO R1
021374  012702  000010                  MOV     #8.,R2                  ; NUMBER OF BITS TO CHECK
021400  012737  000400  002530          MOV     #BIT8,EXP               ; FIRST DATA PATTERN TO CHECK
                                        ;
                                        ; TEST BITS 8-15 WITH MOVING ONE
                                        ;
021406                                  BGNSUB
021406                                  T10.1:
021406  104402                          TRAP    C$BSUB
021410                                  10$:    BGNSEG
021410  104404                          TRAP    C$BSEG
021412  013777  002530  160656          MOV     EXP,@QVNT               ; MOVE EXPECTED TO REGISTER QVNT
021420  017737  160652  002526          MOV     @QVNT,VAL               ; READ QVNT AND MOVE TO VALUE
021426  042737  000377  002526          BIC     #377,VAL                ; MASK OUT UNTESTED BITS
021434  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
021442  001414                          BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
021444  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
021450  001005                          BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
021452                                  ERRHRD  .,MSG13,ERR7            ; TELL OPERATOR ERROR
```

TEST10: CHECK QVNT MI BITS (MOVING 1)

```
        021452  104456                          TRAP    C$ERMRD
        021454  000001                          .WORD   1
        021456  005330                          .WORD   MSG13
        021460  012246                          .WORD   ERR7
   823  021462  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
   824  021464  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
   825  021470  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
   826                                          ;
   827                                          ; END ERROR CONDITION
   828                                          ;
   829  021474                          20$:    ENDSEG
        021474                          10000$:
        021474  104405                          TRAP    C$ESEG
   830  021476  006337  002530                  ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
   831  021502  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
   832  021504  001341                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
   833  021506                                  ENDSUB
        021506                          L10046:
        021506  104403                          TRAP    C$ESUB
   834  021510  005737  002532                  TST     ERROR1                  ; ANY ERRORS
   835  021514  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
   836  021516                                  PRINTX  #FORO                   ; LINE FEED
        021516  012746  003333                  MOV     #FORO,-(SP)
        021522  012746  000001                  MOV     #1,-(SP)
        021526  010600                          MOV     SP,R0
        021530  104415                          TRAP    C$PNTX
        021532  062706  000004                  ADD     #4,SP
   837  021536                          500$:
   838  021536                                  ENDTST
        021536                          L10045:
        021536  104401                          TRAP    C$ETST
```

TEST11: CHECK QVNT HI BITS (MOVING 0)

```
                                        .SBTTL  TEST11: CHECK QVNT HI BITS (MOVING 0)
021540                                  STATST  <CHECK QVNT HI BITS (MOVING 0)>
021540    040    103    110             T11MSG::        .ASCIZ  / CHECK QVNT HI BITS (MOVING 0)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT BITS 8-15 CAN BE LOADED
                                        ;TEST STEPS-
                                        ; LOAD MOVING BIT WORD IN QVNT
                                        ; READ QVNT, CHECK BITS 8-15
                                        ; REPEAT FOR ALL 8 BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************
                                                BGNTST
021600                                  T11::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
021600  104421                                  TRAP    C$RFLA
021602  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
021606  001012                                  BNE     64$                     ; BRANCH IF NOT SET
021610  012746  021540                          MOV     @T11MSG,-(SP)
021614  012746  002270                          MOV     @PNTMSG,-(SP)
021620  012746  000002                          MOV     @2,-(SP)
021624  010600                                  MOV     SP,R0
021626  104417                                  TRAP    C$PNTF
021630  062706  000006                          ADD     @6,SP
021634  005037  002532          64$:    CLR     ERROR1                          ; CLEAR ERROR COUNTER
021640                                          INIT
021640  012777  000100  160450                  MOV     @#QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
021646  013777  002520  160442                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
021654  005077  160420                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
021660  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
021664  012701  000001                          MOV     @1,R1                   ; REGISTER NUMBER TO R1
021670  012702  000010                          MOV     @8.,R2                  ; NUMBER OF BITS TO CHECK
021674  012703  000400                          MOV     @BIT8,R3                ; INITIAL DATA PATTERN TO R3
021700  010337  002530                          MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
021704  005137  002530                          COM     EXP                     ; COMPLEMENT FOR MOVING 0
                                        ;
                                        ; TEST BITS 8-15 WITH MOVING ZERO
                                        ;
021710                                          BGNSUB
021710                                  T11.1:
021710  104402                                  TRAP    C$BSUB
021712                                  10$:    BGNSEG
021712  104404                                  TRAP    C$BSEG
021714  042737  000377  002530                  BIC     @377,EXP                ; MASK OUT UNTESTED BITS
021722  013777  002530  160346                  MOV     EXP,@QVNT               ; MOVE EXPECTED TO REGISTER QVNT
021730  017737  160342  002526                  MOV     @QVNT,VAL               ; READ QVNT AND MOVE TO VALUE
021736  042737  000377  002526                  BIC     @377,VAL                ; MASK OUT UNTESTED BITS
021744  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
021752  001414                                  BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
```

TEST11: CHECK QVNT MI BITS (MOVING 0)

```
877 021754  005737  002532          TST     ERROR1          ; SEE IF FIRST ERROR
878 021760  001005                  BNE     15$             ; SKIP HEADER IF ALREADY PRINTED
879 021762                          ERRHRD  1,MSG14,ERR7    ; TELL OPERATOR ERROR
    021762  104456                  TRAP    C$ERHRD
    021764  000001                  .WORD   1
    021766  005373                  .WORD   MSG14
    021770  012246                  .WORD   ERR7
880 021772  000402                  BR      18$             ; SKIP DETAIL LINE IF HEADER PRINTED
881 021774  004737  012366  15$:    CALL    DETAIL          ; PRINT DETAIL LINE
882 022000  005237  002532  18$:    INC     ERROR1          ; INCREMENT ERROR COUNT
883                                 ;
884                                 ; END ERROR CONDITION
885                                 ;
886 022004                  20$:    ENDSEG
    022004                  10000$:
    022004  104405                  TRAP    C$ESEG
887 022006  006303                  ASL     R3              ; SHIFT BIT FOR NEXT TEST
888 022010  010337  002530          MOV     R3,EXP          ; PLACE PATTERN IN EXPECTED
889 022014  005137  002530          COM     EXP             ; COMPLIMENT FOR MOVING 0
890 022020  005302                  DEC     R2              ; DECREMENT LOOP COUNTER
891 022022  001333                  BNE     10$             ; IF NOT DONE, BRANCH BACK
892 022024                          ENDSUB
    022024                  L10050:
    022024  104403                  TRAP    C$ESUB
893 022026  005737  002532          TST     ERROR1          ; ANY ERRORS
894 022032  001410                  BEQ     500$            ; BRANCH IF NO ERRORS
895 022034                          PRINTX  #FORO           ; LINE FEED
    022034  012746  003333          MOV     #FORO,-(SP)
    022040  012746  000001          MOV     #1,-(SP)
    022044  010600                  MOV     SP,R0
    022046  104415                  TRAP    C$PNTX
    022050  062706  000004          ADD     #4,SP
896 022054                  500$:
897 022054                          ENDTST
    022054                  L10047:
    022054  104401                  TRAP    C$ETST
```

TEST12: CHECK ACK - PRS LINK

```
                                        .SBTTL  TEST12: CHECK ACK - PRS LINK
                                        STATST  <CHECK ACK - PRS LINK>
022056      040     103     110         T12MSG::        .ASCIZ  / CHECK ACK - PRS LINK/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION
                                        ; CHECKS THAT CMDACK SETS VNTACK AND CLEARS CMDPRS
                                        ; CHECKS THAT VNTPRS SETS CMDPRS AND CLEARS VNTACK
                                        ; USES QCMDACK - UEVENTACK LOOP-BACK
                                        ; USES QEVNTPRSNT - UCMDPRSNT LOOP-LOOPBACK
                                        ;TEST STEPS-
                                        ; SET QCMD CMDACK
                                        ; CHECK FOR CMDPRS CLEAR
                                        ; READ QVNT VNTACK, CHECK FOR SET
                                        ; SET VNTPRS
                                        ; READ VNTACK, CHECK FOR CLEAR
                                        ; READ QCMD CMDPRS, CHECK FOR SET
                                        ; LOOPS 100 TIMES OR COMPLETES CURRENT PASS WITH ERROR AND EXITS
                                        ;
                                        ;--
                                        ;*****************************************************************************
022104                                          BGNTST
022104                                  T12::
022104                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
022104  104421                                  TRAP    C$RFLA
022106  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
022112  001012                                  BNE     64$                     ; BRANCH IF NOT SET
022114  012746  022056                          MOV     @T12MSG,-(SP)
022120  012746  002270                          MOV     @PNTMSG,-(SP)
022124  012746  000002                          MOV     @2,-(SP)
022130  010600                                  MOV     SP,R0
022132  104417                                  TRAP    C$PNTF
022134  062706  000006                          ADD     @6,SP
022140  005037  002532          64$:    CLR     ERROR1                          ; CLEAR ERROR COUNTER
022144                                          INIT
                                        ;INITIALIZE DEVICE
022144  012777  000100  160144                  MOV     @MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
022152  013777  002520  160136                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
022160  005077  160114                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
022164  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
                                        ;
                                        ; LOOP THROUGH TEST 100 TIMES
                                        ;
022170  012702  000144                          MOV     @100.,R2                ; # OF LOOPS TO R2
022174                                  1$:
022174                                          BGNSEG
022174  104404                                  TRAP    C$BSEG
022176  012777  000001  160070                  MOV     @CMDACK,@QCMD           ; SET BIT 0 (CMDACK) IN QCMD
022204  017737  160070  002526                  MOV     @QINT,VAL               ; READ AND STORE QINT REGISTER
022212  032737  004000  002526                  BIT     @CMDPRS,VAL             ; IS CMDPRS CLEAR
022220  001413                                  BEQ     2$                      ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION 1
                                        ;
022222  005237  002532                          INC     ERROR1                  ; SET ERROR FLAG
```

TEST12: CHECK ACK - PRS LINK

```
    938 022226  012701  000002                  MOV     #2,R1                   ; REGISTER # FOR MESSAGE
    939 022232  012737  000000  002530          MOV     #0,EXP                  ; VALUE OF EXPECTED FOR MESSAGE
    940 022240                                  ERRHRD  1,MSG15                 ; GO REPORT ERROR TO OPERATOR
        022240  104456                          TRAP    C$ERHRD
        022242  000001                          .WORD   1
        022244  005436                          .WORD   MSG15
        022246  000000                          .WORD   0
    941 022250                              2$:
    942                                     ;
    943                                     ; CHECK TO SEE IF VNTACK WAS SET
    944                                     ;
    945 022250  017737  160024  002526          MOV     @QINT,VAL               ; READ AND STORE QINT REGISTER
    946 022256  032737  002000  002526          BIT     #VNTACK,VAL             ; TEST VNTACK BIT
    947 022264  001013                          BNE     3$                      ; BRANCH IF OK
    948                                     ;
    949                                     ; ERROR CONDITION 2
    950                                     ;
    951 022266  005237  002532                  INC     ERROR1                  ; SET ERROR FLAG
    952 022272  012701  000002                  MOV     #2,R1                   ; REGISTER # FOR MESSAGE
    953 022276  012737  002000  002530          MOV     #VNTACK,EXP             ; VALUE OF EXPECTED FOR MESSAGE
    954 022304                                  ERRHRD  2,MSG16                 ; GO REPORT ERROR TO OPERATOR
        022304  104456                          TRAP    C$ERHRD
        022306  000002                          .WORD   2
        022310  005531                          .WORD   MSG16
        022312  000000                          .WORD   0
    955 022314                              3$:
    956 022314                                  ENDSEG
        022314                              10000$:
        022314  104405                          TRAP    C$ESEG
    957
    958 022316                                  BGNSEG
        022316  104404                          TRAP    C$BSEG
    959 022320  012777  000001  157750          MOV     #VNTPRS,@QVNT           ; SET BIT 0 (VNTPRS) IN QVNT
    960 022326  017737  157746  002526          MOV     @QINT,VAL               ; READ AND STORE QINT REGISTER
    961 022334  032737  002000  002526          BIT     #VNTACK,VAL             ; IS VNTACK CLEAR
    962 022342  001413                          BEQ     4$                      ; BRANCH IF OK
    963                                     ;
    964                                     ; ERROR CONDITION 3
    965                                     ;
    966 022344  005237  002532                  INC     ERROR1                  ; SET ERROR FLAG
    967 022350  012701  000002                  MOV     #2,R1                   ; REGISTER # FOR MESSAGE
    968 022354  012737  000000  002530          MOV     #0,EXP                  ; VALUE OF EXPECTED FOR MESSAGE
    969 022362                                  ERRHRD  3,MSG17                 ; GO REPORT ERROR TO OPERATOR
        022362  104456                          TRAP    C$ERHRD
        022364  000003                          .WORD   3
        022366  005622                          .WORD   MSG17
        022370  000000                          .WORD   0
    970 022372                              4$:
    971                                     ;
    972                                     ; CHECK TO SEE IF QINT CMDPRS WAS SET
    973                                     ;
    974 022372  017737  157702  002526          MOV     @QINT,VAL               ; READ AND STORE QINT REGISTER
    975 022400  032737  004000  002526          BIT     #CMDPRS,VAL             ; TEST CMDPRS BIT
    976 022406  001013                          BNE     5$                      ; BRANCH IF OK
    977                                     ;
    978                                     ; ERROR CONDITION 4
    979                                     ;
```

TEST12: CHECK ACK - PRS LINK

```
    980 022410  005237  002532                  INC     ERROR1                  ; SET ERROR FLAG
    981 022414  012701  000002                  MOV     #2,R1                   ; REGISTER # FOR MESSAGE
    982 022420  012737  004000  002530          MOV     #CMDPRS,EXP             ; VALUE OF EXPECTED FOR MESSAGE
    983 022426                                  ERRHRD  4,MSG18                 ; GO REPORT ERROR TO OPERATOR
        022426  104456                          TRAP    C$ERHRD
        022430  000004                          .WORD   4
        022432  005715                          .WORD   MSG18
        022434  000000                          .WORD   0
    984 022436                          5$:
    985 022436                                  ENDSEG
        022436                          10001$:
        022436  104405                          TRAP    C$ESEG
    986 022440  005737  002532                  TST     ERROR1                  ; TEST ERROR FLAG
    987 022444  001002                          BNE     6$
    988 022446  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
    989 022450  001251                          BNE     1$                      ; LOOP IF NOT DONE
    990 022452                          6$:
    991 022452                                  ENDTST
        022452                          L10051:
        022452  104401                          TRAP    C$ETST
```

TEST13: CHECK QBA (MOVING 1)

```
                                        .SBTTL  TEST13: CHECK QBA (MOVING 1)
                                        STATST  <CHECK QBA (MOVING 1)>
022454  040  103  110                    T13MSG::        .ASCIZ  / CHECK QBA (MOVING 1)/

                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION
                                        ; CHECKS ALL REGISTER BITS
                                        ;TEST STEPS-
                                        ; LOAD WORD IN QBA
                                        ; READ QBA, CHECK
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
                                        T13::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
022502  104421                                  TRAP    C$RFLA
022504  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
022510  001012                                  BNE     64$                     ; BRANCH IF NOT SET
022512  012746  022454                          MOV     @T13MSG,-(SP)
022516  012746  002270                          MOV     @PNTMSG,-(SP)
022522  012746  000002                          MOV     @2,-(SP)
022526  010600                                  MOV     SP,R0
022530  104417                                  TRAP    C$PNTF
022532  062706  000006                          ADD     @6,SP
022536  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                                INIT
                                        ;INITIALIZE DEVICE
022542  012777  000100  157546                  MOV     @MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
022550  013777  002520  157540                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
022556  005077  157516                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
022562  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
022566  012701  000006                          MOV     @6,R1                   ; REGISTER NUMBER TO R1
022572  012702  000020                          MOV     @16.,R2                 ; NUMBER OF BITS TO CHECK
022576  012737  000001  002530                  MOV     @BIT0,EXP               ; FIRST DATA PATTERN TO CHECK
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ONE
                                        ;
                                                BGNSUB
                                        T13.1:
022604  104402                                  TRAP    C$BSUB
                                        10$:    BGNSEG
022606  104404                                  TRAP    C$BSEG
022610  013777  002530  157472                  MOV     EXP,@QBA                ; MOVE EXPECTED TO REGISTER QBA
022616  017737  157466  002526                  MOV     @QBA,VAL                ; READ QBA AND MOVE TO VALUE
022624  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
022632  001414                                  BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
022634  005737  002532                          TST     ERROR1                  ; SEE IF FIRST ERROR
022640  001005                                  BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
022642                                          ERRHRD  1,MSG19,ERR7            ; TELL OPERATOR ERROR
```

TEST13: CHECK QBA (MOVING 1)

```
        022642  104456                  TRAP    C$ERMRD
        022644  000001                  .WORD   1
        022646  006006                  .WORD   MSG19
        022650  012246                  .WORD   ERR7
  1030  022652  000402                  BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
  1031  022654  004737  012366  15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
  1032  022660  005237  002532  18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
  1033                                  ;
  1034                                  ; END ERROR CONDITION
  1035                                  ;
  1036  022664                  20$:    ENDSEG
        022664                  10000$:
        022664  104405                  TRAP    C$ESEG
  1037  022666  006337  002530          ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
  1038  022672  005302                  DEC     R2                      ; DECREMENT LOOP COUNTER
  1039  022674  001344                  BNE     10$                     ; IF NOT DONE, BRANCH BACK
  1040  022676                          ENDSUB
        022676                  L10053:
        022676  104403                  TRAP    C$ESUB
  1041  022700  005737  002532          TST     ERROR1                  ; ANY ERRORS
  1042  022704  001410                  BEQ     500$                    ; BRANCH IF NO ERRORS
  1043  022706                          PRINTX  #FORO                   ; LINE FEED
        022706  012746  003333          MOV     #FORO,-(SP)
        022712  012746  000001          MOV     #1,-(SP)
        022716  010600                  MOV     SP,R0
        022720  104415                  TRAP    C$PNTX
        022722  062706  000004          ADD     #4,SP
  1044  022726                  500$:
  1045  022726                          ENDTST
        022726                  L10052:
        022726  104401                  TRAP    C$ETST
```

TEST14: CHECK QBA (MOVING 0)

```
                                        .SBTTL  TEST14: CHECK QBA (MOVING 0)
                                        STATST  <CHECK QBA (MOVING 0)>
022730  040  103  110                   T14MSG::        .ASCIZ  / CHECK QBA (MOVING 0)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION
                                        ; CHECKS ALL REGISTER BITS
                                        ;TEST STEPS-
                                        ; LOAD WORD IN QBA
                                        ; READ QBA, CHECK
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
                                        T14::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
022756  104421                                  TRAP    C$RFLA
022760  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
022764  001012                                  BNE     64$                     ; BRANCH IF NOT SET
022766  012746  022730                          MOV     @T14MSG,-(SP)
022772  012746  002270                          MOV     @PNTMSG,-(SP)
022776  012746  000002                          MOV     #2,-(SP)
023002  010600                                  MOV     SP,R0
023004  104417                                  TRAP    C$PNTF
023006  062706  000006                          ADD     #6,SP
023012  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
023016                                          INIT
                                        ;INITIALIZE DEVICE
023016  012777  000100  157272                  MOV     @#QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
023024  013777  002520  157264                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
023032  005077  157242                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
023036  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
023042  012701  000006                          MOV     #6,R1                   ; REGISTER NUMBER TO R1
023046  012702  000020                          MOV     #16.,R2                 ; NUMBER OF BITS TO CHECK
023052  012703  000001                          MOV     #BIT0,R3                ; INITIAL DATA PATTERN TO R3
023056  010337  002530                          MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
023062  005137  002530                          COM     EXP                     ; COMPLEMENT FOR MOVING 0
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ZERO
                                        ;
                                                BGNSUB
                                        T14.1:
023066  104402                                  TRAP    C$BSUB
023070                                  10$:    BGNSEG
023070  104404                                  TRAP    C$BSEG
023072  013777  002530  157210                  MOV     EXP,@QBA                ; MOVE EXPECTED TO REGISTER QBA
023100  017737  157204  002526                  MOV     @QBA,VAL                ; READ QBA AND MOVE TO VALUE
023106  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
023114  001414                                  BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
023116  005737  002532                          TST     ERROR1                  ; SEE IF FIRST ERROR
```

TEST14: CHECK QBA (MOVING 0)

```
                                        BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
                                        ERRHRD  1,MSG20,ERR7            ; TELL OPERATOR ERROR
                                        TRAP    C$ERHRD
                                        .WORD   1
                                        .WORD   MSG20
                                        .WORD   ERR7
                                        BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
                                15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
                                18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
                                ;
                                ; END ERROR CONDITION
                                ;
                                20$:    ENDSEG
                                10000$:
                                        TRAP    C$ESEG
                                        ASL     R3                      ; SHIFT BIT FOR NEXT TEST
                                        MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
                                        COM     EXP                     ; COMPLIMENT FOR MOVING 0
                                        DEC     R2                      ; DECREMENT LOOP COUNTER
                                        BNE     10$                     ; IF NOT DONE, BRANCH BACK
                                        ENDSUB
                                L10055:
                                        TRAP    C$ESUB
                                        TST     ERROR1                  ; ANY ERRORS
                                        BEQ     500$                    ; BRANCH IF NO ERRORS
                                        PRINTX  #FORO                   ; LINE FEED
                                        MOV     #FORO,-(SP)
                                        MOV     #1,-(SP)
                                        MOV     SP,R0
                                        TRAP    C$PNTX
                                        ADD     #4,SP
                                500$:
                                        ENDTST
                                L10054:
                                        TRAP    C$ETST
```

TEST15: CHECK QDMA HI BITS (8 15) (MOVING 1)

```
1105                                    .SBTTL  TEST15: CHECK QDMA HI BITS (8-15) (MOVING 1)
1106 023220                             STATST  <CHECK QDMA HI BITS (8 15) (MOVING 1)>
     023220     040     103     110     T15MSG::        .ASCIZ  / CHECK QDMA HI BITS (8-15) (MOVING 1)/
1107
1108                                    ;*****************************************************************************
1109                                    ;++
1110                                    ;TEST DESCRIPTION
1111                                    ; CHECKS HI Q-BUS ADDRESS BITS
1112                                    ;TEST STEPS-
1113                                    ; LOAD QDMA BITS 8 15
1114                                    ; IF U-BRIDGE THEN LOAD BITS 8,9
1115                                    ; READ QDMA, CHECK BITS
1116                                    ; MOVE BIT
1117                                    ; REPEAT FOR ALL BITS
1118                                    ;
1119                                    ;--
1120                                    ;*****************************************************************************
1121
1122 023266                                     BGNTST
     023266                             T15::
1123 023266                                     CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
     023266  104421                             TRAP    C$RFLA
     023270  022700  001000                     CMP     @PNT,R0         ; IS PNT FLAG SET?
     023274  001012                             BNE     64$             ; BRANCH IF NOT SET
     023276  012746  023220                     MOV     @T15MSG,-(SP)
     023302  012746  002270                     MOV     @PNTMSG,-(SP)
     023306  012746  000002                     MOV     #2,-(SP)
     023312  010600                             MOV     SP,R0
     023314  104417                             TRAP    C$PNTF
     023316  062706  000006                     ADD     #6,SP
     023322  005037  002532             64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
1124 023326                                     INIT
                                        ;INITIALIZE DEVICE
     023326  012777  000100  156762             MOV     #MSHALT,@QMT1   ;INITIALIZE Q-BRIDGE
     023334  013777  002520  156754             MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
     023342  005077  156732                     CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
     023346  005037  002532                     CLR     ERROR1          ;CLEAR ERROR FLAG
1125 023352  012701  000004                     MOV     #4,R1           ; REGISTER NUMBER TO R1
1126 023356  012702  000006                     MOV     #6.,R2          ; NUMBER OF BITS TO CHECK
1127 023362  005737  002500                     TST     BRIDGETYPE      ; TEST BRIDGE TYPE
1128 023366  001402                             BEQ     1$              ; IF EQ THEN Q-BRIDGE
1129 023370  012702  000002                     MOV     #2.,R2          ; NUMBER OF BITS TO CHECK (U-BRIDGE)
1130 023374  012737  000400  002530     1$:     MOV     #BIT8,EXP       ; FIRST DATA PATTERN TO CHECK
1131                                    ;
1132                                    ; TEST BITS 8-13 WITH MOVING ONE
1133                                    ;
1134 023402                                     BGNSUB
     023402                             T15.1:
     023402  104402                             TRAP    C$BSUB
1135 023404                             10$:    BGNSEG
     023404  104404                             TRAP    C$BSEG
1136 023406  013777  002530  156670             MOV     EXP,@QDMA       ; MOVE EXPECTED TO REGISTER QDMA
1137 023414  017737  156664  002526             MOV     @QDMA,VAL       ; READ QDMA AND MOVE TO VALUE
1138 023422  042737  140377  002526             BIC     #140377,VAL     ; CLEAR UNTESTED BITS
1139 023430  005737  002500                     TST     BRIDGETYPE      ; TEST BRIDGE TYPE
1140 023434  001403                             BEQ     12$             ; IF EQ THEN Q-BRIDGE
1141 023436  042737  176377  002526             BIC     #176377,VAL     ; CLEAR UNTESTED BITS (U-BRIDGE)
```

TEST15: CHECK QDMA HI BITS (8 15) (MOVING 1)

```
1142 023444  023737  002530  002526  12$:    CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
1143 023452  001414                          BEQ     20$                     ; IF OK, BRANCH
1144                                 ;
1145                                 ; ERROR CONDITION
1146                                 ;
1147 023454  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
1148 023460  001005                          BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
1149 023462                                  ERRHRD  1,MSG21,ERR7            ; TELL OPERATOR ERROR
     023462  104456                          TRAP    C$ERHRD
     023464  000001                          .WORD   1
     023466  006060                          .WORD   MSG21
     023470  012246                          .WORD   ERR7
1150 023472  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
1151 023474  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
1152 023500  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
1153                                 ;
1154                                 ; END ERROR CONDITION
1155                                 ;
1156 023504                          20$:    ENDSEG
     023504                          10000$:
     023504  104405                          TRAP    C$ESEG
1157 023506  006337  002530                  ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
1158 023512  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
1159 023514  001333                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
1160 023516                                  ENDSUB
     023516                          L10057:
     023516  104403                          TRAP    C$ESUB
1161 023520  005737  002532                  TST     ERROR1                  ; ANY ERRORS
1162 023524  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
1163 023526                                  PRINTX  #FORO                   ; LINE FEED
     023526  012746  003333                  MOV     #FORO,-(SP)
     023532  012746  000001                  MOV     #1,-(SP)
     023536  010600                          MOV     SP,R0
     023540  104415                          TRAP    C$PNTX
     023542  062706  000004                  ADD     #4,SP
1164 023546                          500$:
1165 023546                                  ENDTST
     023546                          L10056:
     023546  104401                          TRAP    C$ETST
```

TEST16: CHECK QDMA HI BITS (8 13) (MOVING 0)

```
                                        .SBTTL  TEST16: CHECK QDMA HI BITS (8-13) (MOVING 0)
                                        STATST  <CHECK QDMA HI BITS (8-13) (MOVING 0)>
023550  040     103     110             T16MSG::        .ASCIZ  / CHECK QDMA HI BITS (8-13) (MOVING 0)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS HI Q-BUS ADDRESS BITS
                                        ;TEST STEPS-
                                        ; LOAD QDMA BITS 8-13
                                        ; IF U-BRIDGE THEN LOAD BITS 8,9
                                        ; READ QDMA, CHECK BITS
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
                                        T16::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
023616  104421                                  TRAP    C$RFLA
023620  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
023624  001012                                  BNE     64$                     ; BRANCH IF NOT SET
023626  012746  023550                          MOV     @T16MSG,-(SP)
023632  012746  002270                          MOV     @PNTMSG,-(SP)
023636  012746  000002                          MOV     #2,-(SP)
023642  010600                                  MOV     SP,R0
023644  104417                                  TRAP    C$PNTF
023646  062706  000006                          ADD     #6,SP
023652  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                                INIT
                                        ;INITIALIZE DEVICE
023656  012777  000100  156432                  MOV     @#BQMALT,BQMT1          ;INITIALIZE Q-BRIDGE
023664  013777  002520  156424                  MOV     INITWORD,BQMT1          ;SET UP LOOPBACK OPTION
023672  005077  156402                          CLR     BQINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
023676  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
                                                MOV     #4,R1                   ; REGISTER NUMBER TO R1
                                                MOV     #6.,R2                  ; NUMBER OF BITS TO CHECK
                                                TST     BRIDGETYPE              ; TEST BRIDGE TYPE
                                                BEQ     1$                      ; IF EQ THEN Q-BRIDGE
                                                MOV     #2.,R2                  ; NUMBER OF BITS (U-BRIDGE)
                                        1$:     MOV     #BIT8,R3                ; INITIAL DATA PATTERN TO R3
                                                MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
                                                COM     EXP                     ; COMPLEMENT FOR MOVING 0
                                        ;
                                        ; TEST BITS 8-13 WITH MOVING ZERO
                                        ;
                                                BGNSUB
                                        T16.1:
023740  104402                                  TRAP    C$BSUB
                                        10$:    BGNSEG
023742  104404                                  TRAP    C$BSEG
                                                TST     BRIDGETYPE              ; TEST BRIDGE TYPE
                                                BNE     11$                     ; IF NE THEN U-BRIDGE
                                                BIC     #140377,EXP             ; CLEAR UNTESTED BITS
                                                MOV     EXP,@QDMA               ; MOVE EXPECTED TO REGISTER QDMA
```

TEST16: CHECK QDMA HI BITS (8 13) (MOVING 0)

```
1204 023766  017737  156312  002526          MOV     @QDMA,VAL               ; READ QDMA AND MOVE TO VALUE
1205 023774  042737  140377  002526          BIC     #140377,VAL             ; CLEAR UNTESTED BITS
1206 024002  000414                          BR      12$                     ; COMPARE VALUES
1207 024004  042737  176377  002530  11$:    BIC     #176377,EXP             ; CLEAR UNTESTED BITS
1208 024012  013777  002530  156264          MOV     EXP,@QDMA               ; MOVE EXPECTED TO REGISTER QDMA
1209 024020  017737  156260  002526          MOV     @QDMA,VAL               ; READ QDMA AND MOVE TO VALUE
1210 024026  042737  176377  002526          BIC     #176377,VAL             ; CLEAR UNTESTED BITS
1211 024034  023737  002530  002526  12$:    CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
1212 024042  001414                          BEQ     20$                     ; IF OK, BRANCH
1213                                 ;
1214                                 ; ERROR CONDITION
1215                                 ;
1216 024044  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
1217 024050  001005                          BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
1218 024052                                  ERRHRD  1,MSG22,ERR7            ; TELL OPERATOR ERROR
     024052  104456                          TRAP    C$ERHRD
     024054  000001                          .WORD   1
     024056  006125                          .WORD   MSG22
     024060  012246                          .WORD   ERR7
1219 024062  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
1220 024064  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
1221 024070  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
1222                                 ;
1223                                 ; END ERROR CONDITION
1224                                 ;
1225 024074                          20$:    ENDSEG
     024074                          10000$:
     024074  104405                          TRAP    C$ESEG
1226 024076  006303                          ASL     R3                      ; SHIFT BIT FOR NEXT TEST
1227 024100  010337  002530                  MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
1228 024104  005137  002530                  COM     EXP                     ; COMPLIMENT FOR MOVING 0
1229 024110  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
1230 024112  001313                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
1231 024114                                  ENDSUB
     024114                          L10061:
     024114  104403                          TRAP    C$ESUB
1232 024116  005737  002532                  TST     ERROR1                  ; ANY ERRORS
1233 024122  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
1234 024124                                  PRINTX  #FORO                   ; LINE FEED
     024124  012746  003333                  MOV     #FORO,-(SP)
     024130  012746  000001                  MOV     #1,-(SP)
     024134  010600                          MOV     SP,R0
     024136  104415                          TRAP    C$PNTX
     024140  062706  000004                  ADD     #4,SP
1235 024144                          500$:
1236 024144                                  ENDTST
     024144                          L10060:
     024144  104401                          TRAP    C$ETST
```

TEST17: CHECK QDMA UXAD BITS

```
                                        .SBTTL  TEST17: CHECK QDMA UXAD BITS
                                        STATST  <CHECK QDMA UXAD BITS>
024146  040  103  110                   T17MSG::        .ASCIZ  / CHECK QDMA UXAD BITS/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS UNIBUS EXTENDED ADDRESS BITS
                                        ; LDUXAD DRIVER AND I21 ARE NOT CHECKED
                                        ;TEST STEPS-
                                        ; LOAD QDMA BITS 4,5
                                        ; READ QDMA, CHECK BITS
                                        ; INCREMENT PATTERN
                                        ; REPEAT FOR ALL PATTERNS
                                        ;
                                        ;--
                                        ;*****************************************************************************
024174                                  BGNTST
024174                                  T17::
024174                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
024174  104421                          TRAP    C$RFLA
024176  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
024202  001012                          BNE     64$                     ; BRANCH IF NOT SET
024204  012746  024146                  MOV     @T17MSG,-(SP)
024210  012746  002270                  MOV     @PNTMSG,-(SP)
024214  012746  000002                  MOV     @2,-(SP)
024220  010600                          MOV     SP,R0
024222  104417                          TRAP    C$PNTF
024224  062706  000006                  ADD     @6,SP
024230  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
024234                                  INIT
                                        ;INITIALIZE DEVICE
024234  012777  000100  156054          MOV     @MRESET,@QMT1           ;INITIALIZE Q-BRIDGE
024242  013777  002520  156046          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
024250  005077  156024                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
024254  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
024260  012701  000004                  MOV     @4,R1                   ; REGISTER NUMBER TO R1
024264                                  BGNSUB
024264                          T17.1:
024264  104402                          TRAP    C$BSUB
                                        ;
                                        ; TEST WITH BOTH BITS HIGH
                                        ;
024266  012737  000060  002530          MOV     @60,EXP                 ; BIT PATTERN IN EXPECTED
024274  013777  002530  156002          MOV     EXP,@QDMA               ; MOVE PATTERN TO QDMA REGISTER
024302  017737  155776  002526          MOV     @QDMA,VAL               ; GET VALUE OF REGISTER
024310  042737  177717  002526          BIC     @177717,VAL             ; MASK OUT UNDETERMINED BITS
024316  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
024324  001404                          BEQ     1$                      ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION 1
                                        ;
024326                                  ERRHRD  1,MSG23,ERR4            ; TELL OPERATOR ERROR
024326  104456                          TRAP    C$ERHRD
024330  000001                          .WORD   1
024332  006172                          .WORD   MSG23
```

```
TEST17: CHECK QDMA UXAD BITS

        024334  011776                          .W      ERR4
   1273 024336                          1$:
   1274 024336                                  ENDSUB
        024336                          L10063:
        024336  104403                          TRAP    C$ESUB
   1275 024340                                  BGNSUB
        024340                          T17.2:
        024340  104402                          TRAP    C$BSUB
   1276                                 ;
   1277                                 ; TEST WITH BIT 5 HIGH
   1278                                 ;
   1279 024342  012737  000040  002530          MOV     #40,EXP                 ; BIT PATTERN IN EXPECTED
   1280 024350  013777  002530  155726          MOV     EXP,@QDMA               ; MOVE PATTERN TO QDMA REGISTER
   1281 024356  017737  155722  002526          MOV     @QDMA,VAL               ; GET VALUE OF REGISTER
   1282 024364  042737  177717  002526          BIC     #177717,VAL             ; MASK OUT UNDETERMINED BITS
   1283 024372  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
   1284 024400  001404                          BEQ     2$                      ; IF OK, BRANCH
   1285                                 ;
   1286                                 ; ERROR CONDITION 2
   1287                                 ;
   1288 024402                                  ERRHRD  2,MSG23,ERR4            ; TELL OPERATOR ERROR
        024402  104456                          TRAP    C$ERHRD
        024404  000002                          .WORD   2
        024406  006172                          .WORD   MSG23
        024410  011776                          .WORD   ERR4
   1289 024412                          2$:
   1290 024412                                  ENDSUB
        024412                          L10064:
        024412  104403                          TRAP    C$ESUB
   1291 024414                                  BGNSUB
        024414                          T17.3:
        024414  104402                          TRAP    C$BSUB
   1292                                 ;
   1293                                 ; TEST WITH BIT 4 HIGH
   1294                                 ;
   1295 024416  012737  000020  002530          MOV     #20,EXP                 ; BIT PATTERN IN EXPECTED
   1296 024424  013777  002530  155652          MOV     EXP,@QDMA               ; MOVE PATTERN TO QDMA REGISTER
   1297 024432  017737  155646  002526          MOV     @QDMA,VAL               ; GET VALUE OF REGISTER
   1298 024440  042737  177717  002526          BIC     #177717,VAL             ; MASK OUT UNDETERMINED BITS
   1299 024446  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
   1300 024454  001404                          BEQ     3$                      ; IF OK, BRANCH
   1301                                 ;
   1302                                 ; ERROR CONDITION 3
   1303                                 ;
   1304 024456                                  ERRHRD  3,MSG23,ERR4            ; TELL OPERATOR ERROR
        024456  104456                          TRAP    C$ERHRD
        024460  000003                          .WORD   3
        024462  006172                          .WORD   MSG23
        024464  011776                          .WORD   ERR4
   1305 024466                          3$:
   1306 024466                                  ENDSUB
        024466                          L10065:
        024466  104403                          TRAP    C$ESUB
   1307 024470                                  BGNSUB
        024470                          T17.4:
        024470  104402                          TRAP    C$BSUB
   1308                                 ;
```

TEST17: CHECK QDMA UXAD BITS

```
1309                                        ; TEST WITH BOTH BITS LOW
1310                                        ;
1311 024472  012737  000000  002530          MOV     #00,EXP             ; BIT PATTERN IN EXPECTED
1312 024500  013777  002530  155576          MOV     EXP,@QDMA           ; MOVE PATTERN TO QDMA REGISTER
1313 024506  017737  155572  002526          MOV     @QDMA,VAL           ; GET VALUE OF REGISTER
1314 024514  042737  177717  002526          BIC     #177717,VAL         ; MASK OUT UNDETERMINED BITS
1315 024522  023737  002530  002526          CMP     EXP,VAL             ; COMPARE EXPECTED WITH VALUE
1316 024530  001404                          BEQ     4$                  ; IF OK, BRANCH
1317                                        ;
1318                                        ; ERROR CONDITION 4
1319                                        ;
1320 024532                                 ERRHRD  4,MSG23,ERR4        ; TELL OPERATOR ERROR
     024532  104456                         TRAP    C$ERHRD
     024534  000004                         .WORD   4
     024536  006172                         .WORD   MSG23
     024540  011776                         .WORD   ERR4
1321 024542                                 4$:
1322 024542                                 ENDSUB
     024542                                 L10066:
     024542  104403                         TRAP    C$ESUB
1323 024544                                 ENDTST
     024544                                 L10062:
     024544  104401                         TRAP    C$ETST
```

TEST18: CHECK QUBA (MOVING 1)

```
                                        .SBTTL  TEST18: CHECK QUBA (MOVING 1)
024546                                  STATST  <CHECK QUBA (MOVING 1)>
024546  040  103  110                   T18MSG::        .ASCIZ  / CHECK QUBA (MOVING 1)/

                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION
                                        ; CHECKS ALL REGISTER BITS
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 4 (SELUBAR)
                                        ; LOAD WORD IN QUBA
                                        ; READ QMTO, CHECK
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

024576                                          BGNTST
024576                                  T18::
024576                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
024576  104421                                  TRAP    C$RFLA
024600  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
024604  001012                                  BNE     64$             ; BRANCH IF NOT SET
024606  012746  024546                          MOV     @T18MSG,-(SP)
024612  012746  002270                          MOV     @PNTMSG,-(SP)
024616  012746  000002                          MOV     #2,-(SP)
024622  010600                                  MOV     SP,R0
024624  104417                                  TRAP    C$PNTF
024626  062706  000006                          ADD     #6,SP
024632  005037  002532                  64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
024636                                          INIT
                                        ;INITIALIZE DEVICE
024636  012777  000100  155452                  MOV     @#QHALT,@QMT1   ;INITIALIZE Q-BRIDGE
024644  013777  002520  155444                  MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
024652  005077  155422                          CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
024656  005037  002532                          CLR     ERROR1          ;CLEAR ERROR FLAG
024662  012701  000005                          MOV     #5,R1           ; REGISTER NUMBER TO R1
024666  012702  000020                          MOV     #16.,R2         ; NUMBER OF BITS TO CHECK
024672  012737  000001  002530                  MOV     #BIT0,EXP       ; FIRST DATA PATTERN TO CHECK
024700  052777  000020  155410                  BIS     #SELUBAR,@QMT1  ; SELECT QMT1 FUNC = 4
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ONE
                                        ;
024706                                          BGNSUB
024706                                  T18.1:
024706  104402                                  TRAP    C$BSUB
024710                                  10$:    BGNSEG
024710  104404                                  TRAP    C$BSEG
024712  013777  002530  155366                  MOV     EXP,@QUBA       ; MOVE EXPECTED TO REGISTER QUBA
024720  017737  155370  002526                  MOV     @QMTO,VAL       ; READ QMTO AND MOVE TO VALUE
024726  023737  002530  002526                  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
024734  001414                                  BEQ     20$             ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
```

TEST18: CHECK QUBA (MOVING 1)

```
1362 024736  005737  002532          TST     ERROR1          ; SEE IF FIRST ERROR
1363 024742  001005                  BNE     15$             ; SKIP HEADER IF ALREADY PRINTED
1364 024744                          ERRHRD  1,MSG24,ERR7    ; TELL OPERATOR ERROR
     024744  104456                  TRAP    C$ERHRD
     024746  000001                  .WORD   1
     024750  006241                  .WORD   MSG24
     024752  012246                  .WORD   ERR7
1365 024754  000402                  BR      18$             ; SKIP DETAIL LINE IF HEADER PRINTED
1366 024756  004737  012366  15$:    CALL    DETAIL          ; PRINT DETAIL LINE
1367 024762  005237  002532  18$:    INC     ERROR1          ; INCREMENT ERROR COUNT
1368                                 ;
1369                                 ; END ERROR CONDITION
1370                                 ;
1371 024766                  20$:    ENDSEG
     024766                  10000$:
     024766  104405                  TRAP    C$ESEG
1372 024770  006337  002530          ASL     EXP             ; SHIFT BIT FOR NEXT TEST
1373 024774  005302                  DEC     R2              ; DECREMENT LOOP COUNTER
1374 024776  001344                  BNE     10$             ; IF NOT DONE, BRANCH BACK
1375 025000                          ENDSUB
     025000                  L10070:
     025000  104403                  TRAP    C$ESUB
1376 025002  005737  002532          TST     ERROR1          ; ANY ERRORS
1377 025006  001410                  BEQ     500$            ; BRANCH IF NO ERRORS
1378 025010                          PRINTX  #FORO           ; LINE FEED
     025010  012746  003333          MOV     #FORO,-(SP)
     025014  012746  000001          MOV     #1,-(SP)
     025020  010600                  MOV     SP,R0
     025022  104415                  TRAP    C$PNTX
     025024  062706  000004          ADD     #4,SP
1379 025030                  500$:
1380 025030                          ENDTST
     025030                  L10067:
     025030  104401                  TRAP    C$ETST
```

TEST19: CHECK QUBA (MOVING 0)

```
                                        .SBTTL  TEST19: CHECK QUBA (MOVING 0)
                                        STATST  <CHECK QUBA (MOVING 0)>
025032  040  103  110                   T19MSG::        .ASCIZ  / CHECK QUBA (MOVING 0)/

                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION
                                        ; CHECKS ALL REGISTER BITS
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 4 (SELUBAR)
                                        ; LOAD WORD IN QUBA
                                        ; READ QMTO, CHECK
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

025062                                  BGNTST
025062                                  T19::
025062                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
025062  104421                          TRAP    C$RFLA
025064  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
025070  001012                          BNE     64$                     ; BRANCH IF NOT SET
025072  012746  025032                  MOV     @T19MSG,-(SP)
025076  012746  002270                  MOV     @PNTMSG,-(SP)
025102  012746  000002                  MOV     @2,-(SP)
025106  010600                          MOV     SP,R0
025110  104417                          TRAP    C$PNTF
025112  062706  000006                  ADD     @6,SP
025116  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
025122                                  INIT
                                        ;INITIALIZE DEVICE
025122  012777  000100  155166          MOV     @MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
025130  013777  002520  155160          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
025136  005077  155136                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
025142  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
025146  012701  000005                  MOV     @5,R1                   ; REGISTER NUMBER TO R1
025152  012702  000020                  MOV     @16.,R2                 ; NUMBER OF BITS TO CHECK
025156  012703  000001                  MOV     @BIT0,R3                ; INITIAL DATA PATTERN TO R3
025162  010337  002530                  MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
025166  005137  002530                  COM     EXP                     ; COMPLEMENT FOR MOVING 0
025172  052777  000020  155116          BIS     @SELUBAR,@QMT1          ; SELECT QMT1 FUNC = 4
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ZERO
                                        ;
025200                                  BGNSUB
025200                                  T19.1:
025200  104402                          TRAP    C$BSUB
025202                          10$:    BGNSEG
025202  104404                          TRAP    C$BSEG
025204  013777  002530  155074          MOV     EXP,@QUBA               ; MOVE EXPECTED TO REGISTER QUBA
025212  017737  155076  002526          MOV     @QMTO,VAL               ; READ QMTO AND MOVE TO VALUE
025220  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
025226  001414                          BEQ     20$                     ; IF OK, BRANCH
                                        ;
```

TEST19: CHECK QUBA (MOVING 0)

```
1419                                    ; ERROR CONDITION
1420                                    ;
1421 025230  005737  002532                    TST     ERROR1                  ; SEE IF FIRST ERROR
1422 025234  001005                            BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
1423 025236                                    ERRHRD  1,MSG25,ERR7            ; TELL OPERATOR ERROR
     025236  104456                            TRAP    C$ERHRD
     025240  000001                            .WORD   1
     025242  006267                            .WORD   MSG25
     025244  012246                            .WORD   ERR7
1424 025246  000402                            BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
1425 025250  004737  012366            15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
1426 025254  005237  002532            18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
1427                                    ;
1428                                    ; END ERROR CONDITION
1429                                    ;
1430 025260                            20$:    ENDSEG
     025260                            10000$:
     025260  104405                            TRAP    C$ESEG
1431 025262  006303                            ASL     R3                      ; SHIFT BIT FOR NEXT TEST
1432 025264  010337  002530                    MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
1433 025270  005137  002530                    COM     EXP                     ; COMPLIMENT FOR MOVING 0
1434 025274  005302                            DEC     R2                      ; DECREMENT LOOP COUNTER
1435 025276  001341                            BNE     10$                     ; IF NOT DONE, BRANCH BACK
1436 025300                                    ENDSUB
     025300                            L10072:
     025300  104403                            TRAP    C$ESUB
1437 025302  005737  002532                    TST     ERROR1                  ; ANY ERRORS
1438 025306  001410                            BEQ     500$                    ; BRANCH IF NO ERRORS
1439 025310                                    PRINTX  #FORO                   ; LINE FEED
     025310  012746  003333                    MOV     #FORO,-(SP)
     025314  012746  000001                    MOV     #1,-(SP)
     025320  010600                            MOV     SP,R0
     025322  104415                            TRAP    C$PNTX
     025324  062706  000004                    ADD     #4,SP
1440 025330                            500$:
1441 025330                                    ENDTST
     025330                            L10071:
     025330  104401                            TRAP    C$ETST
```

TEST20: CHECK QWC (MOVING 1)

```
                                        .SBTTL  TEST20: CHECK QWC (MOVING 1)
                                        STATST  <CHECK QWC (MOVING 1)>
025332  040   103   110                 T20MSG::        .ASCIZ  / CHECK QWC (MOVING 1)/

                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION
                                        ; CHECKS ALL REGISTER BITS
                                        ;TEST STEPS-
                                        ; LOAD WORD IN QWC
                                        ; READ QWC, CHECK
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

025360                                          BGNTST
025360                                  T20::
025360                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
025360  104421                                  TRAP    C$RFLA
025362  022700  001000                          CMP     #PNT,R0         ; IS PNT FLAG SET?
025366  001012                                  BNE     64$             ; BRANCH IF NOT SET
025370  012746  025332                          MOV     #T20MSG,-(SP)
025374  012746  002270                          MOV     #PNTMSG,-(SP)
025400  012746  000002                          MOV     #2,-(SP)
025404  010600                                  MOV     SP,R0
025406  104417                                  TRAP    C$PNTF
025410  062706  000006                          ADD     #6,SP
025414  005037  002532                  64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
025420                                          INIT
                                        ;INITIALIZE DEVICE
025420  012777  000100  154670                  MOV     #MQHALT,@QMT1   ;INITIALIZE Q-BRIDGE
025426  013777  002520  154662                  MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
025434  005077  154640                          CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
025440  005037  002532                          CLR     ERROR1          ;CLEAR ERROR FLAG
025444  012701  000007                          MOV     #7,R1           ; REGISTER NUMBER TO R1
025450  012702  000020                          MOV     #16.,R2         ; NUMBER OF BITS TO CHECK
025454  012737  000001  002530                  MOV     #BIT0,EXP       ; FIRST DATA PATTERN TO CHECK
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ONE
                                        ;
025462                                          BGNSUB
025462                                  T20.1:
025462  104402                                  TRAP    C$BSUB
025464                                  10$:    BGNSEG
025464  104404                                  TRAP    C$BSEG
025466  013777  002530  154616                  MOV     EXP,@QWC        ; MOVE EXPECTED TO REGISTER QWC
025474  017737  154612  002526                  MOV     @QWC,VAL        ; READ QWC AND MOVE TO VALUE
025502  023737  002530  002526                  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
025510  001414                                  BEQ     20$             ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
025512  005737  002532                          TST     ERROR1          ; SEE IF FIRST ERROR
025516  001005                                  BNE     15$             ; SKIP HEADER IF ALREADY PRINTED
025520                                          ERRHRD  1,MSG26,ERR7    ; TELL OPERATOR ERROR
```

TEST20: CHECK QWC (MOVING 1)

```
        025520  104456                  TRAP    C$ERHRD
        025522  000001                  .WORD   1
        025524  006315                  .WORD   MSG26
        025526  012246                  .WORD   ERR7
   1480 025530  000402                  BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
   1481 025532  004737  012366  15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
   1482 025536  005237  002532  18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
   1483                                 ;
   1484                                 ; END ERROR CONDITION
   1485                                 ;
   1486 025542                  20$:    ENDSEG
        025542                  10000$:
        025542  104405                  TRAP    C$ESEG
   1487 025544  006337  002530          ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
   1488 025550  005302                  DEC     R2                      ; DECREMENT LOOP COUNTER
   1489 025552  001344                  BNE     10$                     ; IF NOT DONE, BRANCH BACK
   1490 025554                          ENDSUB
        025554                  L10074:
        025554  104403                  TRAP    C$ESUB
   1491 025556  005737  002532          TST     ERROR1                  ; ANY ERRORS
   1492 025562  001410                  BEQ     500$                    ; BRANCH IF NO ERRORS
   1493 025564                          PRINTX  #FORO                   ; LINE FEED
        025564  012746  003333          MOV     #FORO,-(SP)
        025570  012746  000001          MOV     #1,-(SP)
        025574  010600                  MOV     SP,R0
        025576  104415                  TRAP    C$PNTX
        025600  062706  000004          ADD     #4,SP
   1494 025604                  500$:
   1495 025604                          ENDTST
        025604                  L10073:
        025604  104401                  TRAP    C$ETST
```

TEST21: CHECK QWC (MOVING 0)

```
                                        .SBTTL  TEST21: CHECK QWC (MOVING 0)
                                        STATST  <CHECK QWC (MOVING 0)>
025606  040     103     110     T21MSG::        .ASCIZ  / CHECK QWC (MOVING 0)/
                                        ;*****************************************************************
                                        ;..
                                        ;TEST DESCRIPTION
                                        ; CHECKS ALL REGISTER BITS
                                        ;TEST STEPS
                                        ; LOAD WORD IN QWC
                                        ; READ QWC, CHECK
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;
                                        ;*****************************************************************

025634                                  BGNTST
025634                                  T21::
025634                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
025634  104421                          TRAP    C$RFLA
025636  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
025642  001012                          BNE     64$                     ; BRANCH IF NOT SET
025644  012746  025606                  MOV     @T21MSG,-(SP)
025650  012746  002270                  MOV     @PNTMSG,-(SP)
025654  012746  000002                  MOV     #2,-(SP)
025660  010600                          MOV     SP,R0
025662  104417                          TRAP    C$PNTF
025664  062706  000006                  ADD     #6,SP
025670  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
025674                                  INIT
                                        ;INITIALIZE DEVICE
025674  012777  000100  154414          MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
025702  013777  002520  154406          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
025710  005077  154364                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
025714  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
025720  012701  000007                  MOV     #7,R1                   ; REGISTER NUMBER TO R1
025724  012702  000020                  MOV     #16.,R2                 ; NUMBER OF BITS TO CHECK
025730  012703  000001                  MOV     #BIT0,R3                ; INITIAL DATA PATTERN TO R3
025734  010337  002530                  MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
025740  005137  002530                  COM     EXP                     ; COMPLEMENT FOR MOVING 0
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ZERO
                                        ;
025744                                  BGNSUB
025744                                  T21.1:
025744  104402                          TRAP    C$BSUB
025746                          10$:    BGNSEG
025746  104404                          TRAP    C$BSEG
025750  013777  002530  154334          MOV     EXP,@QWC                ; MOVE EXPECTED TO REGISTER QWC
025756  017737  154330  002526          MOV     @QWC,VAL                ; READ QWC AND MOVE TO VALUE
025764  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
025772  001414                          BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
025774  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
```

```
026000  001005                  BNE     15$             ; SKIP HEADER IF ALREADY PRINTED
026002                          ERRHRD  1,MSG27,ERR7    ; TELL OPERATOR ERROR
026002  104456                  TRAP    C$ERHRD
026004  000001                  .WORD   1
026006  006342                  .WORD   MSG27
026010  012246                  .WORD   ERR7
026012  000402                  BR      18$             ; SKIP DETAIL LINE IF HEADER PRINTED
026014  004737  012366  15$:    CALL    DETAIL          ; PRINT DETAIL LINE
026020  005237  002532  18$:    INC     ERROR1          ; INCREMENT ERROR COUNT
                        ;
                        ; END ERROR CONDITION
                        ;
026024                  20$:    ENDSEG
026024                  10000$:
026024  104405                  TRAP    C$ESEG
026026  006303                  ASL     R3              ; SHIFT BIT FOR NEXT TEST
026030  010337  002530          MOV     R3,EXP          ; PLACE PATTERN IN EXPECTED
026034  005137  002530          COM     EXP             ; COMPLIMENT FOR MOVING 0
026040  005302                  DEC     R2              ; DECREMENT LOOP COUNTER
026042  001341                  BNE     10$             ; IF NOT DONE, BRANCH BACK
026044                          ENDSUB
026044                  L10076:
026044  104403                  TRAP    C$ESUB
026046  005737  002532          TST     ERROR1          ; ANY ERRORS
026052  001410                  BEQ     500$            ; BRANCH IF NO ERRORS
026054                          PRINTX  #FORO           ; LINE FEED
026054  012746  003333          MOV     #FORO,-(SP)
026060  012746  000001          MOV     #1,-(SP)
026064  010600                  MOV     SP,R0
026066  104415                  TRAP    C$PNTX
026070  062706  000004          ADD     #4,SP
026074                  500$:
026074                          ENDTST
026074                  L10075:
026074  104401                  TRAP    C$ETST
```

TEST22: CHECK QVAD (Q REGISTER) (MOVING 1)

```
                                        .SBTTL  TEST22: CHECK QVAD (Q-REGISTER) (MOVING 1)
                                        STATST  <CHECK QVAD (Q-REGISTER) (MOVING 1)>
026076  040  103  110                   T22MSG::        .ASCIZ  / CHECK QVAD (Q-REGISTER) (MOVING 1)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION
                                        ; CHECKS THAT QVAD (A-COUNTER) CAN BE LOADED AND READ
                                        ;TEST STEPS-
                                        ; LOAD WORD IN QVAD
                                        ; SET QMT1 MVADR BIT
                                        ; READ QVAD, CHECK
                                        ; CLEAR QMT1 MVADR BIT
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
                                        T22::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
026142  104421                                 TRAP    C$RFLA
026144  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
026150  001012                                 BNE     64$                     ; BRANCH IF NOT SET
026152  012746  026076                         MOV     @T22MSG,-(SP)
026156  012746  002270                         MOV     @PNTMSG,-(SP)
026162  012746  000002                         MOV     #2,-(SP)
026166  010600                                 MOV     SP,R0
026170  104417                                 TRAP    C$PNTF
026172  062706  000006                         ADD     #6,SP
026176  005037  002532                 64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                        ;
                                        ; INITIALIZE DEVICE
                                        ;
026202  012777  000100  154106                  MOV     #MQHALT,@QMT1           ; INITIALIZE Q-BRIDGE
026210  005037  002532                         CLR     ERROR1                  ; CLEAR ERROR COUNTER
026214  012701  000012                         MOV     #10.,R1                 ; REGISTER NUMBER TO R1
026220  012702  000020                         MOV     #16.,R2                 ; NUMBER OF BITS TO CHECK
026224  012737  000001  002530                 MOV     #BIT0,EXP               ; FIRST DATA PATTERN TO CHECK
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ONE
                                        ;
                                                BGNSUB
                                        T22.1:
026232  104402                                 TRAP    C$BSUB
                                        10$:    BGNSEG
026234  104404                                 TRAP    C$BSEG
026236  052777  000020  154036                  BIS     #QVREN,@QCTL            ; READ ENABLE
026244  015777  002530  154046                  MOV     EXP,@QVAD               ; MOVE EXPECTED TO REGISTER QVAD
026252  052777  000002  154036                  BIS     #MVADR,@QMT1            ; SET MVADR BIT IN QMT1
026260  017737  154034  002526                  MOV     @QVAD,VAL               ; READ QVAD AND MOVE TO VALUE
026266  042777  000002  154022                  BIC     #MVADR,@QMT1            ; CLEAR MVADR BIT IN QMT1
026274  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
026302  001414                                 BEQ     20$                     ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
```

TEST22: CHECK QVAD (Q REGISTER) (MOVING 1)

```
 1597                                    ;
 1598 026304  005737  002532             TST     ERROR1                  ; SEE IF FIRST ERROR
 1599 026310  001005                     BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
 1600 026312                             ERRHRD  1,MSG28,ERR7            ; TELL OPERATOR ERROR
      026312  104456                     TRAP    C$ERHRD
      026314  000001                     .WORD   1
      026316  006367                     .WORD   MSG28
      026320  012246                     .WORD   ERR7
 1601 026322  000402                     BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
 1602 026324  004737  012366     15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
 1603 026330  005237  002532     18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
 1604                                    ;
 1605                                    ; END ERROR CONDITION
 1606                                    ;
 1607 026334                     20$:    ENDSEG
      026334                     10000$:
      026334  104405                     TRAP    C$ESEG
 1608 026336  006337  002530             ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
 1609 026342  005302                     DEC     R2                      ; DECREMENT LOOP COUNTER
 1610 026344  001333                     BNE     10$                     ; IF NOT DONE, BRANCH BACK
 1611 026346                             ENDSUB
      026346                     L10100:
      026346  104403                     TRAP    C$ESUB
 1612 026350  005737  002532             TST     ERROR1                  ; ANY ERRORS
 1613 026354  001410                     BEQ     500$                    ; BRANCH IF NO ERRORS
 1614 026356                             PRINTX  #FORO                   ; LINE FEED
      026356  012746  003333             MOV     #FORO,-(SP)
      026362  012746  000001             MOV     #1,-(SP)
      026366  010600                     MOV     SP,R0
      026370  104415                     TRAP    C$PNTX
      026372  062706  000004             ADD     #4,SP
 1615 026376                     500$:
 1616 026376                             ENDTST
      026376                     L10077:
      026376  104401                     TRAP    C$ETST
```

TEST23: CHECK QVAD (Q REGISTER) (MOVING 0)

```
                                        .SBTTL  TEST23: CHECK QVAD (Q-REGISTER) (MOVING 0)
                                        STATST  <CHECK QVAD (Q-REGISTER) (MOVING 0)>
026400  040  103  110                   T23MSG::        .ASCIZ  / CHECK QVAD (Q-REGISTER) (MOVING 0)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION
                                        ; CHECKS THAT QVAD (A-COUNTER) CAN BE LOADED AND READ
                                        ;TEST STEPS-
                                        ; LOAD WORD IN QVAD
                                        ; SET QMT1 MVADR BIT
                                        ; READ QVAD, CHECK
                                        ; CLEAR QMT1 MVADR BIT
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************
026444                                  BGNTST
026444                                  T23::
026444                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
026444  104421                          TRAP    C$RFLA
026446  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
026452  001012                          BNE     64$                     ; BRANCH IF NOT SET
026454  012746  026400                  MOV     @T23MSG,-(SP)
026460  012746  002270                  MOV     @PNTMSG,-(SP)
026464  012746  000002                  MOV     #2,-(SP)
026470  010600                          MOV     SP,R0
026472  104417                          TRAP    C$PNTF
026474  062706  000006                  ADD     #6,SP
026500  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                        ;
                                        ; INITIALIZE DEVICE
                                        ;
026504  012777  000100  153604          MOV     #MQHALT,@QMT1           ; INITIALIZE Q-BRIDGE
026512  005037  002532                  CLR     ERROR1                  ; CLEAR ERROR FLAG
026516  012701  000012                  MOV     #10.,R1                 ; REGISTER NUMBER TO R1
026522  012702  000020                  MOV     #16.,R2                 ; NUMBER OF BITS TO CHECK
026526  012703  000001                  MOV     #BIT0,R3                ; INITIAL DATA PATTERN TO R3
026532  010337  002530                  MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
026536  005137  002530                  COM     EXP                     ; COMPLEMENT FOR MOVING 0
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ZERO
                                        ;
026542                                  BGNSUB
026542                                  T23.1:
026542  104402                          TRAP    C$BSUB
026544                          10$:    BGNSEG
026544  104404                          TRAP    C$BSEG
026546  052777  000020  153526          BIS     #QVREN,@QCTL            ; READ ENABLE
026554  013777  002530  153536          MOV     EXP,@QVAD               ; MOVE EXPECTED TO REGISTER QVAD
026562  052777  000002  153526          BIS     #MVADR,@QMT1            ; SET MVADR BIT IN QMT1
026570  017737  153524  002526          MOV     @QVAD,VAL               ; READ QVAD AND MOVE TO VALUE
026576  042777  000002  153512          BIC     #MVADR,@QMT1            ; CLEAR MVADR BIT IN QMT1
026604  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
026612  001414                          BEQ     20$                     ; IF OK, BRANCH
```

TEST23: CHECK QVAD (Q-REGISTER) (MOVING 0)

```
1660                                    ;
1661                                    ; ERROR CONDITION
1662                                    ;
1663 026614  005737  002532            TST     ERROR1                  ; SEE IF FIRST ERROR
1664 026620  001005                    BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
1665 026622                            ERRHRD  1,MSG29,ERR7            ; TELL OPERATOR ERROR
     026622  104456                    TRAP    C$ERHRD
     026624  000001                    .WORD   1
     026626  006430                    .WORD   MSG29
     026630  012246                    .WORD   ERR7
1666 026632  000402                    BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
1667 026634  004737  012366    15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
1668 026640  005237  002532    18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
1669                                    ;
1670                                    ; END ERROR CONDITION
1671                                    ;
1672 026644                    20$:    ENDSEG
     026644                    10000$:
     026644  104405                    TRAP    C$ESEG
1673 026646  006303                    ASL     R3                      ; SHIFT BIT FOR NEXT TEST
1674 026650  010337  002530            MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
1675 026654  005137  002530            COM     EXP                     ; COMPLIMENT FOR MOVING 0
1676 026660  005302                    DEC     R2                      ; DECREMENT LOOP COUNTER
1677 026662  001330                    BNE     10$                     ; IF NOT DONE, BRANCH BACK
1678 026664                            ENDSUB
     026664                    L10102:
     026664  104403                    TRAP    C$ESUB
1679 026666  005737  002532            TST     ERROR1                  ; ANY ERRORS
1680 026672  001410                    BEQ     500$                    ; BRANCH IF NO ERRORS
1681 026674                            PRINTX  #FORO                   ; LINE FEED
     026674  012746  003333            MOV     #FORO,-(SP)
     026700  012746  000001            MOV     #1,-(SP)
     026704  010600                    MOV     SP,R0
     026706  104415                    TRAP    C$PNTX
     026710  062706  000004            ADD     #4,SP
1682 026714                    500$:
1683 026714                            ENDTST
     026714                    L10101:
     026714  104401                    TRAP    C$ETST
1684 026716                            INIT
                                ;INITIALIZE DEVICE
     026716  012777  000100  153372    MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
     026724  013777  002520  153364    MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
     026732  005077  153342            CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
     026736  005037  002532            CLR     ERROR1                  ;CLEAR ERROR FLAG
```

TEST24: CHECK QVAD (U REGISTER) (MOVING 1)

```
                                        .SBTTL  TEST24: CHECK QVAD (U-REGISTER) (MOVING 1)
                                        STATST  <CHECK QVAD (U-REGISTER) (MOVING 1)>
026742  040  103  110                   T24MSG::        .ASCIZ  / CHECK QVAD (U-REGISTER) (MOVING 1)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION
                                        ; CHECKS THAT QVAD (C-COUNTER) CAN BE LOADED AND READ
                                        ; USES LDQVADR - LDUVADR LOOP-BACK
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 3 (SELUVADR)
                                        ; LOAD WORD IN QMT0
                                        ; SET QMT1 MVADR BIT
                                        ; READ QMT0, CHECK
                                        ; CLEAR QMT1 MVADR BIT
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************

027006                                  BGNTST
027006                                  T24::
027006                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
027006  104421                                  TRAP    C$RFLA
027010  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
027014  001012                                  BNE     64$                     ; BRANCH IF NOT SET
027016  012746  026742                          MOV     @T24MSG,-(SP)
027022  012746  002270                          MOV     @PNTMSG,-(SP)
027026  012746  000002                          MOV     #2,-(SP)
027032  010600                                  MOV     SP,R0
027034  104417                                  TRAP    C$PNTF
027036  062706  000006                          ADD     #6,SP
027042  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
027046                                          INIT
                                        ;INITIALIZE DEVICE
027046  012777  000100  153242                  MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
027054  013777  002520  153234                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
027062  005077  153212                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
027066  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
027072  012701  000012                          MOV     #10.,R1                 ; REGISTER NUMBER TO R1
027076  052777  000014  153212                  BIS     #SELUVADR,@QMT1         ; SELECT QMT1 FUNCTION = 3
027104  012702  000020                          MOV     #16.,R2                 ; NUMBER OF BITS TO CHECK
027110  012737  000001  002530                  MOV     #BIT0,EXP               ; FIRST DATA PATTERN TO CHECK
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ONE
                                        ;
027116                                          BGNSUB
027116                                  T24.1:
027116  104402                                  TRAP    C$BSUB
027120                                  10$:    BGNSEG
027120  104404                                  TRAP    C$BSEG
027122  013777  002530  153164                  MOV     EXP,@QMT0               ; MOVE EXPECTED TO REGISTER QVAD
027130  052777  000002  153160                  BIS     #MVADR,@QMT1            ; SET MVADR BIT IN QMT1
027136  017737  153152  002526                  MOV     @QMT0,VAL               ; READ QVAD AND MOVE TO VALUE
027144  042777  000002  153144                  BIC     #MVADR,@QMT1            ; CLEAR MVADR BIT IN QMT1
```

TEST24: CHECK QVAD (U REGISTER) (MOVING 1)

```
1723 027152  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
1724 027160  001414                          BEQ     20$                     ; IF OK, BRANCH
1725                                         ;
1726                                         ; ERROR CONDITION
1727                                         ;
1728 027162  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
1729 027166  001005                          BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
1730 027170                                  ERRHRD  1,MSG30,ERR7            ; TELL OPERATOR ERROR
     027170  104456                          TRAP    C$ERHRD
     027172  000001                          .WORD   1
     027174  006471                          .WORD   MSG30
     027176  012246                          .WORD   ERR7
1731 027200  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
1732 027202  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
1733 027206  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
1734                                         ;
1735                                         ; END ERROR CONDITION
1736                                         ;
1737 027212                          20$:    ENDSEG
     027212                          10000$:
     027212  104405                          TRAP    C$ESEG
1738 027214  006337  002530                  ASL     EXP                     ; SHIFT BIT FOR NEXT TEST
1739 027220  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
1740 027222  001336                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
1741 027224                                  ENDSUB
     027224                          L10104:
     027224  104403                          TRAP    C$ESUB
1742 027226  005737  002532                  TST     ERROR1                  ; ANY ERRORS
1743 027232  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
1744 027234                                  PRINTX  #FORO                   ; LINE FEED
     027234  012746  003333                  MOV     #FORO,-(SP)
     027240  012746  000001                  MOV     #1,-(SP)
     027244  010600                          MOV     SP,R0
     027246  104415                          TRAP    C$PNTX
     027250  062706  000004                  ADD     #4,SP
1745 027254                          500$:
1746 027254                                  ENDTST
     027254                          L10103:
     027254  104401                          TRAP    C$ETST
```

TEST25: CHECK QVAD (U REGISTER) (MOVING 0)

```
                                        .SBTTL  TEST25: CHECK QVAD (U-REGISTER) (MOVING 0)
                                        STATST  <CHECK QVAD (U-REGISTER) (MOVING 0)>
027256  040  103  110                   T25MSG::        .ASCIZ  / CHECK QVAD (U-REGISTER) (MOVING 0)/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION
                                        ; CHECKS THAT QVAD (C-COUNTER) CAN BE LOADED AND READ
                                        ; USES LDQVADR - LDUVADR LOOP-BACK
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 3 (SELUVADR)
                                        ; LOAD WORD IN QMT0
                                        ; SET QMT1 MVADR BIT
                                        ; READ QMT0, CHECK
                                        ; CLEAR QMT1 MVADR BIT
                                        ; MOVE BIT
                                        ; REPEAT FOR ALL BITS
                                        ;
                                        ;--
                                        ;*****************************************************************************
                                                BGNTST
                                        T25::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
027322  104421                                 TRAP    C$RFLA
027324  022700  001000                         CMP     @PNT,R0                 ; IS PNT FLAG SET?
027330  001012                                 BNE     64$                     ; BRANCH IF NOT SET
027332  012746  027256                         MOV     @T25MSG,-(SP)
027336  012746  002270                         MOV     @PNTMSG,-(SP)
027342  012746  000002                         MOV     #2,-(SP)
027346  010600                                 MOV     SP,R0
027350  104417                                 TRAP    C$PNTF
027352  062706  000006                         ADD     #6,SP
027356  005037  002532                 64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                                INIT
                                        ;INITIALIZE DEVICE
027362  012777  000100  152726                 MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
027370  013777  002520  152720                 MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
027376  005077  152676                         CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
027402  005037  002532                         CLR     ERROR1                  ;CLEAR ERROR FLAG
027406  012701  000012                         MOV     #10.,R1                 ; REGISTER NUMBER TO R1
027412  012702  000020                         MOV     #16.,R2                 ; NUMBER OF BITS TO CHECK
027416  012703  000001                         MOV     #BIT0,R3                ; INITIAL DATA PATTERN TO R3
027422  010337  002530                         MOV     R3,EXP                  ; MOVE BIT PATTERN TO EXPECTED
027426  005137  002530                         COM     EXP                     ; COMPLEMENT FOR MOVING 0
027432  052777  000014  152656                 BIS     #SELUVADR,@QMT1         ; SELECT FUNCTION 3 IN QMT1
                                        ;
                                        ; TEST BITS 0-15 WITH MOVING ZERO
                                        ;
                                                BGNSUB
                                        T25.1:
027440  104402                                 TRAP    C$BSUB
                                        10$:    BGNSEG
027442  104404                                 TRAP    C$BSEG
027444  013777  002530  152642                 MOV     EXP,@QMT0               ; MOVE EXPECTED TO REGISTER QVAD
027452  052777  000002  152636                 BIS     #MVADR,@QMT1            ; SET MVADR BIT IN QMT1
```

TEST25: CHECK QVAD (U-REGISTER) (MOVING 0)

```
 1785 027460  017737  152630  002526          MOV     @QMT0,VAL               ; MOVE CONTENTS OF QVAD TO VALUE
 1786 027466  042777  000002  152622          BIC     #MVADR,@QMT1            ; CLEAR MVADR BIT IN QMT1
 1787 027474  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
 1788 027502  001414                          BEQ     20$                     ; IF OK, BRANCH
 1789                                 ;
 1790                                 ; ERROR CONDITION
 1791                                 ;
 1792 027504  005737  002532                  TST     ERROR1                  ; SEE IF FIRST ERROR
 1793 027510  001005                          BNE     15$                     ; SKIP HEADER IF ALREADY PRINTED
 1794 027512                                  ERRHRD  1,MSG31,ERR7            ; TELL OPERATOR ERROR
      027512  104456                          TRAP    C$ERHRD
      027514  000001                          .WORD   1
      027516  006532                          .WORD   MSG31
      027520  012246                          .WORD   ERR7
 1795 027522  000402                          BR      18$                     ; SKIP DETAIL LINE IF HEADER PRINTED
 1796 027524  004737  012366          15$:    CALL    DETAIL                  ; PRINT DETAIL LINE
 1797 027530  005237  002532          18$:    INC     ERROR1                  ; INCREMENT ERROR COUNT
 1798                                 ;
 1799                                 ; END ERROR CONDITION
 1800                                 ;
 1801 027534                          20$:    ENDSEG
      027534                          10000$:
      027534  104405                          TRAP    C$ESEG
 1802 027536  006303                          ASL     R3                      ; SHIFT BIT FOR NEXT TEST
 1803 027540  010337  002530                  MOV     R3,EXP                  ; PLACE PATTERN IN EXPECTED
 1804 027544  005137  002530                  COM     EXP                     ; COMPLIMENT FOR MOVING 0
 1805 027550  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
 1806 027552  001333                          BNE     10$                     ; IF NOT DONE, BRANCH BACK
 1807 027554                                  ENDSUB
      027554                          L10106:
      027554  104403                          TRAP    C$ESUB
 1808 027556  005737  002532                  TST     ERROR1                  ; ANY ERRORS
 1809 027562  001410                          BEQ     500$                    ; BRANCH IF NO ERRORS
 1810 027564                                  PRINTX  #FORO                   ; LINE FEED
      027564  012746  003333                  MOV     #FORO,-(SP)
      027570  012746  000001                  MOV     #1,-(SP)
      027574  010600                          MOV     SP,R0
      027576  104415                          TRAP    C$PNTX
      027600  062706  000004                  ADD     #4,SP
 1811 027604                          500$:
 1812 027604                                  ENDTST
      027604                          L10105:
      027604  104401                          TRAP    C$ETST
```

TEST26: CHECK QVAD INCREMENT (Q-REGISTER)

```
                                        .SBTTL  TEST26: CHECK QVAD INCREMENT (Q-REGISTER)
                                        STATST  <CHECK QVAD INCREMENT (Q-REGISTER)>
027606  040  103  110                   T26MSG::        .ASCIZ  / CHECK QVAD INCREMENT (Q-REGISTER)/
                                        ;**************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION
                                        ; CHECKS THAT QVAD (A-COUNTER) INCREMENT ON READ QVDA
                                        ;TEST STEPS-
                                        ; LOAD QVAD WITH 1'S
                                        ; READ QVDA
                                        ; READ QVAD, CHECK FOR 0
                                        ;
                                        ;--
                                        ;**************************************************************************
                                                BGNTST
                                        T26::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
027652  104421                                  TRAP    C$RFLA
027654  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
027660  001012                                  BNE     64$                     ; BRANCH IF NOT SET
027662  012746  027606                          MOV     @T26MSG,-(SP)
027666  012746  002270                          MOV     @PNTMSG,-(SP)
027672  012746  000002                          MOV     #2,-(SP)
027676  010600                                  MOV     SP,R0
027700  104417                                  TRAP    C$PNTF
027702  062706  000006                          ADD     #6,SP
027706  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                        ;
                                        ; INITIALIZE DEVICE
                                        ;
027712  012777  000100  152376                  MOV     #MQHALT,@QMT1           ; INITIALIZE Q-BRIDGE
027720  012701  000012                          MOV     #10.,R1                 ; REGISTER NUMBER TO R1
                                        ;
                                        ; TEST FOR INCREMENT
                                        ;
                                                BGNSUB
                                        T26.1:
                                                TRAP    C$BSUB
027726  052777  000020  152346                  BIS     #QVREN,@QCTL            ; READ ENABLE
027734  012777  177777  152356                  MOV     #-1,@QVAD               ; MOVE ALL ONE'S TO QVAD
027742  017702  152354                          MOV     @QVDA,R2                ; DUMMY READ TO QVDA
027746  012737  000000  002530                  MOV     #0,EXP                  ; CLEAR EXPECTED FIELD
027754  052777  000002  152334                  BIS     #MVADR,@QMT1            ; ENABLE TO READ QVAD
027762  017737  152332  002526                  MOV     @QVAD,VAL               ; MOVE CURRENT VALUE OF QVAD TO VAL
027770  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
027776  001406                                  BEQ     1$                      ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
030000                                          ERRHRD  1,MSG32,ERR4            ; TELL OPERATOR ERROR
030000  104456                                  TRAP    C$ERHRD
030002  000001                                  .WORD   1
030004  006573                                  .WORD   MSG32
030006  011776                                  .WORD   ERR4
030010  004737  013230                          JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
```

TEST26: CHECK QVAD INCREMENT (Q-REGISTER)

```
1853 030014                                1$:
1854 030014                                        ENDSUB
     030014                                L10110:
     030014  104403                                TRAP    C$ESUB
1855 030016                                        ENDTST
     030016                                L10107:
     030016  104401                                TRAP    C$ETST
```

```
                                        .SBTTL  TEST27: CHECK QVAD INCREMENT (U-REGISTER)
030020                                  STATST  <CHECK QVAD INCREMENT (U-REGISTER)>
030020      040     103     110         T27MSG::        .ASCIZ  / CHECK QVAD INCREMENT (U-REGISTER)/

                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT QVAD (C-COUNTER) INCREMENT ON LDUVDAT
                                        ; USES LDQVADR - LDUVADR LOOP-BACK
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QMT1 FUNC = 3 (SELUVADR)
                                        ; LOAD QMTO WITH 1'S
                                        ; SET QMT1 FUNC = 6 (SEL UV DAT)
                                        ; LOAD QMTO (ANY VALUE)
                                        ; SET QMT1 MVADR BIT
                                        ; SET QMT1 FUNC = 3 (SELUVADR)
                                        ; READ QMTO, CHECK FOR 0
                                        ;
                                        ;--
                                        ;*****************************************************************************

030064                                          BGNTST
030064                                  T27::
030064                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
030064  104421                                  TRAP    C$RFLA
030066  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
030072  001012                                  BNE     64$                     ; BRANCH IF NOT SET
030074  012746  030020                          MOV     @T27MSG,-(SP)
030100  012746  002270                          MOV     @PNTMSG,-(SP)
030104  012746  000002                          MOV     #2,-(SP)
030110  010600                                  MOV     SP,R0
030112  104417                                  TRAP    C$PNTF
030114  062706  000006                          ADD     #6,SP
030120  005037  002532            64$:          CLR     ERROR1                  ; CLEAR ERROR COUNTER
030124                                          INIT
                                        ;INITIALIZE DEVICE
030124  012777  000100  152164                  MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
030132  013777  002520  152156                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
030140  005077  152134                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
030144  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
030150  012701  000012                          MOV     #10.,R1                 ; REGISTER NUMBER TO R1
                                        ;
                                        ; CHECK QVAD INCREMEMT
                                        ;
030154                                          BGNSUB
030154                                  T27.1:
030154  104402                                  TRAP    C$BSUB
030156  052777  000014  152132                  BIS     #SELUVADR,@QMT1         ; SELECT QMT1 FUNC = 3 (SELUVADR)
030164  012777  177777  152122                  MOV     #-1,@QMTO               ; LOAD QMTO WITH ALL ZERO'S
030172  042777  000014  152116                  BIC     #SELUVADR,@QMT1         ; DESELECT QMT1 FUNC = 3 (SELUVADR)
030200  052777  000030  152110                  BIS     #SELUVDAT,@QMT1         ; SELECT QMT1 FUNC = 6 (SELUVDAT)
030206  012777  177777  152100                  MOV     #-1,@QMTO               ; LOAD QMTO (ANY VALUE)
030214  042777  000030  152074                  BIC     #SELUVDAT,@QMT1         ; DESELECT QMT1 FUNC = 6 (SELUVDAT)
030222  052777  000016  152066                  BIS     #SELUVADR!MVADR,@QMT1   ; SET QMT1 = FUNC 3 (SELUVADR) & MVARD BIT
030230  012737  000000  002530                  MOV     #0,EXP                  ; LOAD EXP WITH EXPECTED VALUE
030236  017737  152052  002526                  MOV     @QMTO,VAL               ; READ QMTO, LOAD INTO VALUE
```

TEST27: CHECK QVAD INCREMENT (U REGISTER)

```
1895 030244  042777  000016  152044          BIC     @SELUVADR!MVADR,@QMT1   ; DESELECT QMT1 + FUNC 3 (SELUVADR) & MVADR BIT
1896 030252  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
1897 030260  001406                          BEQ     10                      ; IF OK, BRANCH
1898                                ;
1899                                ; ERROR CONDITION
1900                                ;
1901 030262                                   ERRMRD  1,MSG33,ERR4            ; TELL OPERATOR ERROR
     030262  104456                           TRAP    C$ERMRD
     030264  000001                           .WORD   1
     030266  006633                           .WORD   MSG33
     030270  011776                           .WORD   ERR4
1902 030272  004737  013230                   JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
1903 030276                                10:
1904 030276                                   ENDSUB
     030276                                L10112:
     030276  104403                           TRAP    C$ESUB
1905 030300                                   ENDTST
     030300                                L10111:
     030300  104401                           TRAP    C$ETST
```

TEST28: CHECK VIEW REGISTER RAM

```
                                        .SBTTL  TEST28: CHECK VIEW REGISTER RAM
                                        STATST  <CHECK VIEW REGISTER RAM>
030302  040     103     110     T28MSG::        .ASCIZ  / CHECK VIEW REGISTER RAM/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS EACH VIEW DATA RAM LOCATION WITH 1'S AND 0 S
                                        ; CHECKS ALL LOCATIONS FOR NO CHANGE
                                        ; USES QVAD (Q-REGISTER) FOR READ ADDRESS
                                        ; USED QVAD (U-REGISTER) FOR WRITE ADDRESS
                                        ; USES LUQVDAT - LDUVDAT LOOP-BACK
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-

                                        ; SET MIODIS (LOOPBACK ENABLE BIT)
                                        ; SELECT FUNCTION = 3 UVADR
                                        ; LOAD QVAD (UVADR) WITH ADDRESS (START AT 0)
                                        ; DESELECT FUNCTION 3
                                        ; SELECT FUNCTION = 6 UVDAT
                                        ; LOAD QMTO WITH 1'S (QVDA WRITE)
                                        ; CLEAR MIODIS (LOOPBACK ENABLE BIT) AND FUNCTION 6
                                        ; LOAD QVAD (QVADR) WITH ADDRESS
                                        ; READ QVDA, CHECK FOR ALL 1'S

                                        ; SET MIODIS (LOOPBACK ENABLE BIT)
                                        ; SELECT FUNCTION = 3 UVADR
                                        ; LOAD QVAD (UVADR) WITH ADDRESS
                                        ; DESELECT FUNCTION 3
                                        ; SELECT FUNCTION = 6 UVDAT
                                        ; LOAD QMTO WITH 0'S (QVDA WRITE)
                                        ; SELECT FUNCTION = 3 UVADR
                                        ; SET MVADR BIT (QVAD DIRECTION BIT) IN QMT1
                                        ; READ QMTO (UVADR) ADDRESS, CHECK FOR INCREMENT
                                        ; CLEAR MVADR BIT IN QMT1
                                        ; CLEAR MIODIS (LOOPBACK ENABLE BIT) AND FUNCTION 6
                                        ; LOAD QVAD (QVADR) WITH ADDRESS
                                        ; READ QVDA, CHECK FOR ALL 0'S
                                        ; SET MVADR BIT (QVAD DIRECTION BIT) IN QMT1
                                        ; READ QVAD, CHECK FOR ADDRESS INCREMENTS
                                        ; CLEAR QMT1 MVADR BIT
                                        ; REPEAT FOR 2K ADDRESSES (RAM ADDRESS SHOULD BE 0 AT END)
                                        ;
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
030334                                  T28::
030334                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
030334  104421                                  TRAP    C$RFLA
030336  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
030342  001012                                  BNE     64$                     ; BRANCH IF NOT SET
030344  012746  030302                          MOV     @T28MSG,-(SP)
030350  012746  002270                          MOV     @PNTMSG,-(SP)
030354  012746  000002                          MOV     #2,-(SP)
030360  010600                                  MOV     SP,R0
030362  104417                                  TRAP    C$PNTF
```

TEST28: CHECK VIEW REGISTER RAM

```
                030364  062706  000006          ADD     #6,SP
                030370  005037  002532  648:    CLR     ERROR1          ; CLEAR ERROR COUNTER
1954                                            ;
1955                                            ; INITIALIZE DEVICE
1956                                            ;
1957            030374  012777  000100  151714  MOV     #MQHALT,@QMT1   ; INITIALIZE Q-BRIDGE
1958            030402  012701  004000          MOV     #4000,R1        ; ALLOW FOR 2K RAM IN LOOP
1959            030406  005037  002532          CLR     ERROR1          ; CLEAR ERROR OCCURED COUNTER
1960            030412  005037  002534          CLR     ERROR2          ; CLEAR ERROR OCCURED COUNTER
1961            030416  012737  000000  002524  MOV     #0,ADD          ; START AT RAM ADDRESS ZERO
1962                                            ;
1963                                            ; CHECK VIEW REGISTER RAM
1964                                            ;
1965            030424                          BGNSUB
                030424                  T28.1:
                030424  104402                  TRAP    C$BSUB
1966            030426                  L28:                            ; START LOOPS HERE
1967            030426                          BGNSEG
                030426  104404                  TRAP    C$BSEG
1968                                            ;
1969                                            ; FIRST CHECK WITH 1'S
1970                                            ; SET U SIDE RAM ADDRESS
1971                                            ;
1972            030430  013777  002520  151660  MOV     INITWORD,@QMT1  ; SET UP LOOPBACK OPTION
1973            030436  005077  151636          CLR     @QINT           ; CLEAR QINT AFTER LOOPBACK ENABLED
1974            030442  042777  000020  151632  BIC     #QVREN,@QCTL    ; CLEAR READ ENABLE
1975            030450  052777  000014  151640  BIS     #SELUVADR,@QMT1 ; SELECT UVADR
1976            030456  013777  002524  151630  MOV     ADD,@QMTO       ; LOAD U SIDE OF QVAD WITH RAM ADDRESS
1977                                            ;
1978                                            ; SET UP FOR WRITE TO RAM
1979                                            ;
1980            030464  042777  000014  151624  BIC     #SELUVADR,@QMT1 ; DESELECT UVADR
1981            030472  052777  000030  151616  BIS     #SELUVDAT,@QMT1 ; SELECT UVDAT TO WRITE TO RAM
1982                                            ;
1983                                            ; WRITE DATA
1984                                            ;
1985            030500  012737  177777  002530  MOV     #-1,EXP         ; EXPECTED PATTEN TO EXP
1986            030506  013777  002530  151600  MOV     EXP,@QMTO       ; MOVE ALL 1'S TO QVDA THROUGH QMTO
1987                                            ;
1988                                            ; SET Q SIDE RAM ADDRESS
1989                                            ;
1990            030514  052777  000020  151560  BIS     #QVREN,@QCTL    ; READ ENABLE
1991            030522  012777  000000  151566  MOV     #0,@QMT1        ; CLEAR MAINTENANCE LOOP-BACK BIT
1992            030530  013777  002524  151562  MOV     ADD,@QVAD       ; LOAD QVAD WITH RAM ADDRESS
1993                                            ;
1994                                            ; READ DATA
1995                                            ;
1996            030536  017737  151560  002526  MOV     @QVDA,VAL       ; READ QVDA
1997            030544  023737  002530  002526  CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
1998            030552  001433                  BEQ     20$             ; IF OK, BRANCH
1999                                            ;
2000                                            ; ERROR CONDITION 1
2001                                            ;
2002            030554  005737  002532          TST     ERROR1          ; CHECK FOR PREVIOUS ERRORS
2003            030560  001004                  BNE     10$             ; BRANCH IF PREVIOUS ERRORS
2004            030562                          ERRHRD  1,MSG34,ERR5    ; PRINT ERROR HEADER
                030562  104456                  TRAP    C$ERHRD
```

TEST28: CHECK VIEW REGISTER RAM

```
        030564  000001                          .WORD   1
        030566  006673                          .WORD   MSG34
        030570  012136                          .WORD   ERR5
2005    030572  022737  000023  002532  10$:    CMP     #19.,ERROR1             ; SFE IF TOO MANY TO PRINT
2006    030600  100416                          BMI     15$                     ; BRANCH IF MORE THAN 10 PRINTED
2007    030602                                  PRINTB  #FOR4,ADD,VAL,EXP       ; PRINT ERROR LINE
        030602  013746  002530                  MOV     EXP,-(SP)
        030606  013746  002526                  MOV     VAL,-(SP)
        030612  013746  002524                  MOV     ADD,-(SP)
        030616  012746  003552                  MOV     #FOR4,-(SP)
        030622  012746  000004                  MOV     #4,-(SP)
        030626  010600                          MOV     SP,R0
        030630  104414                          TRAP    C$PNTB
        030632  062706  000012                  ADD     #12,SP
2008    030636  005237  002532          15$:    INC     ERROR1                  ; INCREMENT ERROR OCCURRED COUNTER
2009    030642                          20$:
2010                                    ;
2011                                    ; NOW CHECK WITH ALL 0'S
2012                                    ; RESTORE U SIDE RAM ADDRESS AFTER INCREMENT
2013                                    ;
2014    030642  013777  002520  151446            MOV     INITWORD,@QMT1          ; SET UP LOOPBACK OPTION
2015    030650  005077  151424                    CLR     @QINT                   ; CLEAR QINT AFTER LOOPBACK ENABLED
2016    030654  042777  000020  151420            BIC     #QVREN,@QCTL            ; CLEAR READ ENABLE
2017    030662  052777  000014  151426            BIS     #SELUVADR,@QMT1         ; SELECT UVADR
2018    030670  013777  002524  151416            MOV     ADD,@QMTO               ; LOAD U SIDE OF QVAD WITH RAM ADDRESS
2019                                    ;
2020                                    ; SET UP FOR WRITE TO RAM
2021                                    ;
2022    030676  042777  000014  151412            BIC     #SELUVADR,@QMT1         ; DESELECT UVADR
2023    030704  052777  000030  151404            BIS     #SELUVDAT,@QMT1         ; SELECT UVDAT TO WRITE TO RAM
2024                                    ;
2025                                    ; WRITE DATA
2026                                    ;
2027    030712  012737  000000  002530            MOV     #0,EXP                  ; LOAD ALL 0'S TO EXP
2028    030720  013777  002530  151366            MOV     EXP,@QMTO               ; MOVE ALL 0'S TO QVDA THROUGH QMTO
2029                                    ;
2030                                    ; NOW CHECK TO SEE IF QVAD U SIDE REALLY DID INCREMENT
2031                                    ;
2032    030726  042777  000030  151362            BIC     #SELUVDAT,@QMT1         ; DESELECT UVDAT
2033    030734  052777  000014  151354            BIS     #SELUVADR,@QMT1         ; SELECT UVADR
2034    030742  052777  000002  151346            BIS     #MVADR,@QMT1            ; SET MVADR BIT IN QMT1
2035    030750  017737  151340  002526            MOV     @QMTO,VAL               ; READ U ADDRESS IN QVAD FROM QMTO
2036    030756  042777  000002  151332            BIC     #MVADR,@QMT1            ; CLEAR MVADR BIT IN QMT1
2037    030764  042777  000014  151324            BIC     #SELUVADR,@QMT1         ; DESELECT UVADR
2038    030772  013737  002524  002530            MOV     ADD,EXP                 ; MOVE CURRENT ADDRESS TO EXPECTED
2039    031000  005237  002530                    INC     EXP                     ; INCREMENT EXPECTED
2040    031004  023737  002530  002526            CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2041    031012  001434                            BEQ     170$                    ; IF OK, BRANCH
2042                                    ;
2043                                    ; ERROR CONDITION 2
2044                                    ;
2045    031014  005737  002532                    TST     ERROR1                  ; CHECK FOR PREVIOUS ERROR FROM ABOVE
2046    031020  001007                            BNE     150$                    ; IF ERROR FROM ABOVE, BRANCH
2047    031022  005737  002534                    TST     ERROR2                  ; CHECK FOR PREVIOUS ERROR
2048    031026  001004                            BNE     150$                    ; BRANCH IF PREVIOUS ERRORS
2049    031030                                  ERRHRD  2,MSG35
        031030  104456                          TRAP    C$ERHRD
```

TEST28: CHECK VIEW REGISTER RAM

```
        031032  000002                          .WORD   2
        031034  006736                          .WORD   MSG35
        031036  000000                          .WORD   0
2050    031040  022737  000011  002534  150$:   CMP     #9.,ERROR2              ; SEE IF TOO MANY TO PRINT
2051    031046  100414                          BMI     160$                    ; BRANCH IF MORE THAN 10 PRINTED
2052    031050                                  PRINTB  #FOR5.5,EXP,VAL         ; PRINT ERROR LINE
        031050  013746  002526                  MOV     VAL,-(SP)
        031054  013746  002530                  MOV     EXP,-(SP)
        031060  012746  003762                  MOV     #FOR5.5,-(SP)
        031064  012746  000003                  MOV     #3,-(SP)
        031070  010600                          MOV     SP,R0
        031072  104414                          TRAP    C$PNTB
        031074  062706  000010                  ADD     #10,SP
2053    031100  005237  002534          160$:   INC     ERROR2                  ; INCREMENT ERROR COUNTER
2054
2055    031104                          170$:
2056                                            ;
2057                                            ; RESTORE Q SIDE RAM ADDRESS AFTER INCREMENT
2058                                            ;
2059    031104  052777  000020  151170          BIS     #QVREN,@QCTL            ; READ ENABLE
2060    031112  012777  000000  151176          MOV     #0,@QMT1                ; CLEAR MAINTENANCE LOOP-BACK BIT
2061    031120  013777  002524  151172          MOV     ADD,@QVAD               ; LOAD QVAD WITH RAM ADDRESS
2062    031126  012737  000000  002530          MOV     #0,EXP                  ; RESTORE EXPECTED VALUE
2063                                            ;
2064                                            ; READ DATA
2065                                            ;
2066    031134  017737  151162  002526          MOV     @QVDA,VAL               ; READ QVDA
2067    031142  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2068    031150  001433                          BEQ     40$                     ; IF OK, BRANCH
2069                                            ;
2070                                            ; ERROR CONDITION 3
2071                                            ;
2072    031152  005737  002532                  TST     ERROR1                  ; CHECK FOR PREVIOUS ERRORS
2073    031156  001004                          BNE     30$                     ; BRANCH IF PREVIOUS ERRORS
2074    031160                                  ERRHRD  3,MSG34,ERR5            ; PRINT ERROR HEADER
        031160  104456                          TRAP    C$ERHRD
        031162  000003                          .WORD   3
        031164  006673                          .WORD   MSG34
        031166  012136                          .WORD   ERR5
2075    031170  022737  000023  002532  30$:    CMP     #19.,ERROR1             ; SEE IF TOO MANY TO PRINT
2076    031176  100416                          BMI     35$                     ; BRANCH IF MORE THAN 10 PRINTED
2077    031200                                  PRINTB  #FOR4,ADD,VAL,EXP       ; PRINT ERROR LINE
        031200  013746  002530                  MOV     EXP,-(SP)
        031204  013746  002526                  MOV     VAL,-(SP)
        031210  013746  002524                  MOV     ADD,-(SP)
        031214  012746  003552                  MOV     #FOR4,-(SP)
        031220  012746  000004                  MOV     #4,-(SP)
        031224  010600                          MOV     SP,R0
        031226  104414                          TRAP    C$PNTB
        031230  062706  000012                  ADD     #12,SP
2078    031234  005237  002532          35$:    INC     ERROR1                  ; INCREMENT ERROR OCCURRED COUNTER
2079
2080    031240                          40$:
2081                                            ;
2082                                            ; NOW CHECK TO SEE IF QVAD Q SIDE REALLY DID INCREMENT
2083                                            ;
2084    031240  052777  000002  151050          BIS     #MVADR,@QMT1            ; SET MVADR BIT IN QMT1
```

TEST28: CHECK VIEW REGISTER RAM

```
2085 031246  017737  151046  002526          MOV     @QVAD,VAL               ; READ ADDRESS
2086 031254  042777  000002  151034          BIC     #MVADR,@QMT1            ; CLEAR MVADR BIT IN QMT1
2087 031262  013737  002524  002530          MOV     ADD,EXP                 ; MOVE CURRENT ADDRESS TO EXPECTED
2088 031270  005237  002530                  INC     EXP                     ; INCREMENT EXPECTED
2089 031274  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2090 031302  001434                          BEQ     61$                     ; IF OK, BRANCH
2091                                         ;
2092                                         ; ERROR CONDITION 4
2093                                         ;
2094 031304  005737  002532                  TST     ERROR1                  ; CHECK FOR PREVIOUS ERROR FROM ABOVE
2095 031310  001007                          BNE     50$                     ; IF ERROR FROM ABOVE, BRANCH
2096 031312  005737  002534                  TST     ERROR2                  ; CHECK FOR PREVIOUS ERROR
2097 031316  001004                          BNE     50$                     ; BRANCH IF PREVIOUS ERRORS
2098 031320                                  ERRHRD  4,MSG35
     031320  104456                          TRAP    C$ERHRD
     031322  000004                          .WORD   4
     031324  006736                          .WORD   MSG35
     031326  000000                          .WORD   0
2099 031330  022737  000011  002534  50$:    CMP     #9.,ERROR2              ; SEE IF TOO MANY TO PRINT
2100 031336  100414                          BMI     60$                     ; BRANCH IF MORE THAN 10 PRINTED
2101 031340                                  PRINTB  #FOR5,EXP,VAL           ; PRINT ERROR LINE
     031340  013746  002526                  MOV     VAL,-(SP)
     031344  013746  002530                  MOV     EXP,-(SP)
     031350  012746  003670                  MOV     #FOR5,-(SP)
     031354  012746  000003                  MOV     #3,-(SP)
     031360  010600                          MOV     SP,R0
     031362  104414                          TRAP    C$PNTB
     031364  062706  000010                  ADD     #10,SP
2102 031370  005237  002534          60$:    INC     ERROR2                  ; INCREMENT ERROR COUNTER
2103
2104 031374                          61$:
2105 031374                                  ENDSEG
     031374                          10000$:
     031374  104405                          TRAP    C$ESEG
2106                                         ;
2107                                         ; SET UP FOR NEXT LOOP
2108                                         ;
2109 031376  005237  002524                  INC     ADD                     ; INCREMENT TO NEXT RAM ADDRESS
2110 031402  005301                          DEC     R1                      ; DECREMENT LOOP COUNTER
2111 031404  001402                          BEQ     75$                     ; IF DONE, BRANCH
2112 031406  000137  030426                  JMP     L28                     ; NOT FINISHED WITH LOOP, JUMP
2113                                         ;
2114                                         ; LOOP MUST BE COMPLETED, PRINT FINAL ERROR TOTALS IF ANY
2115                                         ;
2116 031412                          75$:
2117 031412  013701  002532                  MOV     ERROR1,R1               ; MOVE ERROR1 COUNT TO R1
2118 031416  063701  002534                  ADD     ERROR2,R1               ; ADD ERROR2 COUNT TO R1
2119 031422  005701                          TST     R1                      ; TEST FOR NO ERRORS
2120 031424  001434                          BEQ     80$                     ; BRANCH IF NO ERRORS
2121                                         ;
2122                                         ; PRINT FINAL ERROR TOTALS
2123                                         ;
2124 031426                                  PRINTB  #FOR6,ERROR1            ; ERROR ON ERROR COUNT
     031426  013746  002532                  MOV     ERROR1,-(SP)
     031432  012746  004054                  MOV     #FOR6,-(SP)
     031436  012746  000002                  MOV     #2,-(SP)
     031442  010600                          MOV     SP,R0
```

TEST28: CHECK VIEW REGISTER RAM

```
        031444  104414                          TRAP    C$PNTB
        031446  062706  000006                  ADD     #6,SP
  2125  031452                                  PRINTB  #FOR7,ERROR2            ; ERROR TWO ERROR COUNT
        031452  013746  002534                  MOV     ERROR2,-(SP)
        031456  012746  004114                  MOV     #FOR7,-(SP)
        031462  012746  000002                  MOV     #2,-(SP)
        031466  010600                          MOV     SP,R0
        031470  104414                          TRAP    C$PNTB
        031472  062706  000006                  ADD     #6,SP
  2126  031476                                  PRINTB  #FOR0                   ; LINE FEED CARRIAGE RETURN
        031476  012746  003333                  MOV     #FOR0,-(SP)
        031502  012746  000001                  MOV     #1,-(SP)
        031506  010600                          MOV     SP,R0
        031510  104414                          TRAP    C$PNTB
        031512  062706  000004                  ADD     #4,SP
  2127  031516                          80$:    ENDSUB
        031516                        L10114:
        031516  104403                          TRAP    C$ESUB
  2128  031520                                  ENDTST
        031520                        L10113:
        031520  104401                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)
                                        STATST  <CHECK RAM ADDRESSING (ADDRESS VALUE)>
031522  040  103  110                   T29MSG::        .ASCIZ  / CHECK RAM ADDRESSING (ADDRESS VALUE)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS VIEW DATA RAM ADDRESSING
                                        ; LOADS ALL RAM LOCATIONS WITH ADDRESS
                                        ; CHECKS ALL RAM LOCATIONS
                                        ; USES QVAD (Q-REGISTER) FOR ADDRESS
                                        ; USES LDQVDAT - LDUVDAT LOOP-BACK
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; SELECT FUNCTION UVADR
                                        ; LOAD QVAD WITH ZERO'S THROUGH QMTO
                                        ; DESELECT FUNCTION UVADR
                                        ; SELECT FUNCTION UVDAT
                                        ; LOAD QVDA WITH ADDRESS VALUE THROUGH QMTO
                                        ; REPEAT LOAD QVDA THROUGH QMTO FOR 2K ADDRESSES
                                        ; CLEAR MIODIS MAINTENANCE LOOP-BACK BIT
                                        ; LOAD QVAD WITH ZERO
                                        ; READ QVDA, CHECK FOR ZERO OR 2000 (CONFIRMS 1K OR 2K)
                                        ; TELL OPERATOR 1K RAM, 2K RAM OR UNKNOWN SIZE RAM
                                        ; READ QVDA, CHECK DATA
                                        ; REPEAT READ QVDA FOR ALL RAM
                                        ;
                                        ;--
                                        ;*****************************************************************************
031570                                          BGNTST
                                        T29::
                                                CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
031570  104421                                  TRAP    C$RFLA
031572  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
031576  001012                                  BNE     64$             ; BRANCH IF NOT SET
031600  012746  031522                          MOV     @T29MSG,-(SP)
031604  012746  002270                          MOV     @PNTMSG,-(SP)
031610  012746  000002                          MOV     #2,-(SP)
031614  010600                                  MOV     SP,R0
031616  104417                                  TRAP    C$PNTF
031620  062706  000006                          ADD     #6,SP
031624  005037  002532                  64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
031630                                          INIT
                                        ;INITIALIZE DEVICE
031630  012777  000100  150460                  MOV     @#MQHALT,@QMT1  ;INITIALIZE Q-BRIDGE
031636  013777  002520  150452                  MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
031644  005077  150430                          CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
031650  005037  002532                          CLR     ERROR1          ;CLEAR ERROR FLAG
031654  012701  000000                          MOV     #0,R1           ; CLEAR R1 AS COUNTER
                                        ;
                                        ; SET ADDRESS
                                        ;
031660  042777  000020  150414                  BIC     #QVREN,@QCTL    ; CLEAR READ ENABLE
031666  052777  000014  150422                  BIS     #SELUVADR,@QMT1 ; SELECT UVADR
031674  012777  000000  150412                  MOV     #0,@QMTO        ; LOAD U SIDE OF QVAD WITH RAM ADDRESS
                                        ;
```

TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```
                                        ; SET UP FOR WRITE TO RAM
                                        ;
032702  042777  000014  150406                BIC     #SELUVADR,@QMT1         ; DESELECT UVADR
032710  052777  000030  150400                BIS     #SELUVDAT,@QMT1         ; SELECT UVDAT TO WRITE TO RAM
                                        ;
                                        ; LOAD RAM WITH ADDRESSES
                                        ;
032716                                  10$:
032716  010177  150372                        MOV     R1,@QMTO                ; LOAD QVDA WITH ADDRESS DATA THROUGH QMTO
032722  005201                                INC     R1                      ; INCREMENT COUNTER
032724  020127  004000                        CMP     R1,#4000                ; HAS IT LOOPED 2K YET?
032730  001372                                BNE     10$                     ; IF NOT DONE, GO AGAIN
                                        ;
                                        ; ALL RAM LOADED, NOW CHECK FOR 1K OR 2K PRESENT
                                        ;
032732  042777  000200  150356                BIC     #MIODIS,@QMT1           ; CLEAR MAINTENANCE LOOP-BACK BIT
032740  012737  000002  002530                MOV     #2,EXP                  ; AMOUNT OF RAM TO EXP FOR MESSAGE
032746  012701  000000                        MOV     #0,R1                   ; SET COUNTER FOR 2K RAM
032752  012737  000000  002524                MOV     #0,ADD                  ; POINT ADD TO FIRST RAM ADDRESS
                                        ;
                                        ; SET ADDRESS
                                        ;
032760  052777  000020  150314                BIS     #QVREN,@QCTL            ; READ ENABLE
032766  012777  000000  150324                MOV     #0,@QVAD                ; LOAD QVAD WITH RAM ADDRESS
032774  017737  150322  002526                MOV     @QVDA,VAL               ; READ FIRST RAM ADDRESS
033002  022737  000000  002526                CMP     #0,VAL                  ; IS IT 2K RAM
033010  001450                                BEQ     20$                     ; BRANCH IF 2K RAM
033012  012737  000001  002530                MOV     #1,EXP                  ; ASSUME 1K RAM FOR MESSAGES
033020  012701  002000                        MOV     #2000,R1                ; LOAD R1 COUNTER WITH 1K
033024  022737  002000  002526                CMP     #2000,VAL               ; IS IT 1K RAM
033032  001437                                BEQ     20$                     ; BRANCH IF 1K RAM
                                        ;
                                        ; ERROR CONDITION, UNKNOWN RAM SIZE
                                        ; ASSUME 1K FOR REST OF TEST
                                        ;
033034  005237  002532                        INC     ERROR1                  ; INCREMENT ERROR COUNTER
033040                                      ERRHRD  1,MSG34,ERR5            ; PRINT ERROR HEADER
033040  104456                              TRAP    C$ERHRD
033042  000001                              .WORD   1
033044  006673                              .WORD   MSG34
033046  012136                               WORD   ERR5
                                        ;
                                        ; TELL OPERATOR UNKNOWN RAM SIZE
                                        ;
033050                                      PRINTB  #FOR4.5                 ; PRINT UNKNOWN RAM SIZE MESSAGE
033050  012746  003600                        MOV     #FOR4.5,-(SP)
033054  012746  000001                        MOV     #1,-(SP)
033060  010600                                MOV     SP,R0
033062  104414                              TRAP    C$PNTB
033064  062706  000004                        ADD     #4,SP
033070  010137  002530                        MOV     R1,EXP                  ; EXPECTED VALUE FROM R1
033074                                      PRINTB  #FOR4,ADD,VAL,EXP       ; PRINT ERROR LINE
033074  013746  002530                        MOV     EXP,-(SP)
033100  013746  002526                        MOV     VAL,-(SP)
033104  013746  002524                        MOV     ADD,-(SP)
033110  012746  003552                        MOV     #FOR4,-(SP)
033114  012746  000004                        MOV     #4,-(SP)
```

TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```
        032120  010600                          MOV     SP,R0
        032122  104414                          TRAP    C$PNTB
        032124  062706  000012                  ADD     #12,SP
   2213 032130  000417                          BR      40$
   2214                                 ;
   2215                                 ; END ERROR CONDITION
   2216                                 ; PRINT RAM SIZE IF DETERMINED AND NOT PREVIOUSLY PRINTED
   2217                                 ;
   2218 032132  005737  002536          20$:    TST     RAM                     ; SEE IF RAM SIZE ALREADY PRINTED
   2219 032136  001014                          BNE     40$                     ; IF RAM SIZE PRINTED, BRANCH
   2220 032140                                  PRINTB  #FOR9,EXP               ; TELL OPERATOR HOW MUCH RAM PRESENT
        032140  013746  002530                  MOV     EXP,-(SP)
        032144  012746  004227                  MOV     #FOR9,-(SP)
        032150  012746  000002                  MOV     #2,-(SP)
        032154  010600                          MOV     SP,R0
        032156  104414                          TRAP    C$PNTB
        032160  062706  000006                  ADD     #6,SP
   2221 032164  005237  002536                  INC     RAM                     ; SET RAM FLAG MESSAGE PRINTED
   2222                                 ;
   2223                                 ; NOW LOOP THROUGH AND CHECK REST OF RAM
   2224                                 ;
   2225 032170                          40$:
   2226 032170  005201                          INC     R1                      ; INCREMENT COUNTER
   2227 032172  005237  002524                  INC     ADD                     ; INCREMENT RAM ADDRESS POINTER
   2228 032176  017737  150120  002526          MOV     @#VDA,VAL               ; READ NEXT RAM ADDRESS
   2229 032204  010137  002530                  MOV     R1,EXP                  ; GET EXPECTED IN EXP
   2230 032210  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED TO VALUE
   2231 032216  001433                          BEQ     50$                     ; IF OK, BRANCH
   2232                                 ;
   2233                                 ; ERROR CONDITION
   2234                                 ;
   2235 032220  005737  002532                  TST     ERROR1                  ; HAVE ANY ERRORS OCCURED YET?
   2236 032224  001004                          BNE     45$                     ; IF PREVIOUS ERROR, BRANCH
   2237                                 ;
   2238                                 ; NO PREVIOUS ERRORS
   2239                                 ;
   2240 032226                                  ERRHRD  2,MSG34,ERR5            ; PRINT ERROR HEADER
        032226  104456                          TRAP    C$ERHRD
        032230  000002                          .WORD   2
        032232  006673                          .WORD   MSG34
        032234  012136                          .WORD   ERR5
   2241 032236  022737  000023  002532  45$:    CMP     #19.,ERROR1             ; SEE IF TOO MANY ERRORS TO PRINT
   2242 032244  100416                          BMI     48$                     ; BRANCH IF MORE THEN 20 PRINTED
   2243 032246                                  PRINTB  #FOR4,ADD,VAL,EXP       ; PRINT ERROR LINE
        032246  013746  002530                  MOV     EXP,-(SP)
        032252  013746  002526                  MOV     VAL,-(SP)
        032256  013746  002524                  MOV     ADD,-(SP)
        032262  012746  003552                  MOV     #FOR4,-(SP)
        032266  012746  000004                  MOV     #4,-(SP)
        032272  010600                          MOV     SP,R0
        032274  104414                          TRAP    C$PNTB
        032276  062706  000012                  ADD     #12,SP
   2244 032302  005237  002532          48$:    INC     ERROR1                  ; INCREMENT ERROR COUNTER
   2245                                 ;
   2246                                 ; END OF ERROR ROUTINE, CHECK FOR END OF TEST
   2247                                 ;
   2248 032306                          50$:
```

TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```
   2249 032306  020127  003777                CMP     R1,#3777                ; END OF TEST
   2250 032312  001326                         BNE     40$                     ; BRANCH IF TEST NOT DONE
   2251                                ;
   2252                                ; TEST DONE, PRINT FINAL ERROR TOTALS IF ANY
   2253                                ;
   2254 032314  005737  002532                 TST     ERROR1                  ; ANY ERRORS?
   2255 032320  001422                         BEQ     60$                     ; IF NO, BRANCH
   2256                                ;
   2257                                ; PRINT FINAL ERROR TOTALS
   2258                                ;
   2259 032322                                 PRINTB  #FOR6,ERROR1            ; PRINT NUMBER OF ERRORS
        032322  013746  002532                 MOV     ERROR1,-(SP)
        032326  012746  004054                 MOV     #FOR6,-(SP)
        032332  012746  000002                 MOV     #2,-(SP)
        032336  010600                         MOV     SP,R0
        032340  104414                         TRAP    C$PNTB
        032342  062706  000006                 ADD     #6,SP
   2260 032346                                 PRINTB  #FOR0                   ; LINE FEED CARRIAGE RETURN
        032346  012746  003333                 MOV     #FOR0,-(SP)
        032352  012746  000001                 MOV     #1,-(SP)
        032356  010600                         MOV     SP,R0
        032360  104414                         TRAP    C$PNTB
        032362  062706  000004                 ADD     #4,SP
   2261 032366                         60$:
   2262 032366                                 ENDTST
        032366                         L10115:
        032366  104401                         TRAP    C$ETST
```

TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

```
                                .SBTTL  TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)
                               STATST  <CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)>
032370  040  103  110  T30MSG::        .ASCIZ  / CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)/

                               ;*****************************************************************************
                               ;++
                               ;TEST DESCRIPTION-
                               ; CHECKS VIEW DATA RAM ADDRESSING
                               ; LOADS ALL RAM LOCATIONS WITH ADDRESS
                               ; CHECKS ALL RAM LOCATIONS
                               ; USES QVAD (Q-REGISTER) FOR ADDRESS
                               ; USES LDQVDAT - LDUVDAT LOOP-BACK
                               ; USES QTSOUT - QTSIN LOOP-BACK
                               ;TEST STEPS-
                               ; SELECT FUNCTION UVADR
                               ; LOAD QVAD WITH ZERO'S THROUGH QMTO
                               ; DESELECT FUNCTION UVADR
                               ; SELECT FUNCTION UVDAT
                               ; LOAD QVDA WITH COMPLIMENTED ADDRESS VALUE THROUGH QMTO
                               ; REPEAT LOAD QVDA THROUGH QMTO FOR 2K ADDRESSES
                               ; CLEAR MIODIS MAINTENANCE LOOP-BACK BIT
                               ; LOAD QVAD WITH ZERO
                               ; READ QVDA, CHECK FOR ZERO OR 2000 (CONFIRMS 1K OR 2K)
                               ; TELL OPERATOR 1K RAM, 2K RAM OR UNKNOWN SIZE RAM
                               ; READ QVDA, CHECK DATA
                               ; REPEAT READ QVDA FOR ALL RAM
                               ;
                               ;--
                               ;*****************************************************************************

032452                                 BGNTST
032452                         T30::
032452                                 CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
032452  104421                         TRAP    C$RFLA
032454  022700  001000                 CMP     @PNT,R0                 ; IS PNT FLAG SET?
032460  001012                         BNE     64$                     ; BRANCH IF NOT SET
032462  012746  032370                 MOV     @T30MSG,-(SP)
032466  012746  002270                 MOV     @PNTMSG,-(SP)
032472  012746  000002                 MOV     @2,-(SP)
032476  010600                         MOV     SP,R0
032500  104417                         TRAP    C$PNTF
032502  062706  000006                 ADD     @6,SP
032506  005037  002532         64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
032512                                 INIT
                               ;INITIALIZE DEVICE
032512  012777  000100  147576         MOV     @#QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
032520  013777  002520  147570         MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
032526  005077  147546                 CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
032532  005037  002532                 CLR     ERROR1                  ;CLEAR ERROR FLAG
032536  012701  177777                 MOV     @-1,R1                  ; FIRST 1'S COMPLEMENT TO R1
                               ;
                               ; SET ADDRESS
                               ;
032542  042777  000020  147532         BIC     @QVREN,@QCTL            ; CLEAR READ ENABLE
032550  052777  000014  147540         BIS     @SELUVADR,@QMT1         ; SELECT UVADR
032556  012777  000000  147530         MOV     @0,@QMTO                ; LOAD U SIDE OF QVAD WITH RAM ADDRESS
                               ;
```

TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

```
2304                                        ; SET UP FOR WRITE TO RAM
2305                                        ;
2306 032564  042777  000014  147524                BIC     #SELUVADR,@QMT1         ; DESELECT UVADR
2307 032572  052777  000030  147516                BIS     #SELUVDAT,@QMT1         ; SELECT UVDAT TO WRITE TO RAM
2308                                        ;
2309                                        ; LOAD RAM WITH ADDRESSES
2310                                        ;
2311 032600                                 10$:
2312 032600  010177  147510                        MOV     R1,@QMTO                ; LOAD QVDA WITH ADDRESS DATA THROUGH QMTO
2313 032604  005301                                DEC     R1                      ; DECREMENT COUNTER
2314 032606  020127  173777                        CMP     R1,#^C4000              ; HAS IT LOOPED 2K YET?
2315 032612  001372                                BNE     10$                     ; IF NOT DONE, GO AGAIN
2316                                        ;
2317                                        ; ALL RAM LOADED, NOW CHECK FOR 1K OR 2K PRESENT
2318                                        ;
2319 032614  052777  000020  147460                BIS     #QVREN,@QCTL            ; READ ENABLE
2320 032622  042777  000200  147466                BIC     #MIODIS,@QMT1           ; CLEAR MAINTENANCE LOOP-BACK BIT
2321 032630  012737  000002  002530                MOV     #2,EXP                  ; ASSUME 2K RAM FOR MESSAGES
2322 032636  012701  000000                        MOV     #0,R1                   ; SET COUNTER FOR 2K RAM
2323 032642  012737  000000  002524                MOV     #0,ADD                  ; POINT ADD TO FIRST RAM ADDRESS
2324                                        ;
2325                                        ; SET ADDRESS
2326                                        ;
2327 032650  012777  000000  147442                MOV     #0,@QVAD                ; LOAD QVAD WITH RAM ADDRESS
2328 032656  017737  147440  002526                MOV     @QVDA,VAL               ; READ FIRST RAM ADDRESS
2329 032664  022737  177777  (   )26                CMP     #^C0,VAL                ; IS IT 2K RAM
2330 032672  001452                                BEQ     20$                     ; BRANCH IF 2K RAM
2331 032674  012737  000001  002530                MOV     #1,EXP                  ; AMOUNT OF RAM TO EXP FOR MESSAGE
2332 032702  012701  002000                        MOV     #2000,R1                ; LOAD R1 COUNTER WITH 1K
2333 032706  022737  175777  002526                CMP     #^C2000,VAL             ; IS IT 1K RAM
2334 032714  001441                                BEQ     20$                     ; IF 1K RAM, BRANCH
2335                                        ;
2336                                        ; ERROR CONDITION, UNKNOWN RAM SIZE
2337                                        ; ASSUME 1K FOR REST OF TEST
2338                                        ;
2339 032716  005237  002532                        INC     ERROR1                  ; INCREMENT ERROR COUNTER
2340 032722                                        ERRHRD  1,MSG34,ERR5            ; PRINT ERROR HEADER
     032722  104456                                TRAP    C$ERHRD
     032724  000001                                .WORD   1
     032726  006673                                .WORD   MSG34
     032730  012136                                .WORD   ERR5
2341                                        ;
2342                                        ; TELL OPERATOR UNKNOWN RAM SIZE
2343                                        ;
2344 032732                                        PRINTB  #FOR4.5                 ; PRINT UNKNOWN RAM SIZE MESSAGE
     032732  012746  003600                        MOV     #FOR4.5,-(SP)
     032736  012746  000001                        MOV     #1,-(SP)
     032742  010600                                MOV     SP,R0
     032744  104414                                TRAP    C$PNTB
     032746  062706  000004                        ADD     #4,SP
2345 032752  010137  002530                        MOV     R1,EXP                  ; EXPECTED VALUE FROM R1
2346 032756  005137  002530                        COM     EXP                     ; COMPLEMENT EXPRESION
2347 032762                                        PRINTB  #FOR4,ADD,VAL,EXP       ; PRINT ERROR LINE
     032762  013746  002530                        MOV     EXP,-(SP)
     032766  013746  002526                        MOV     VAL,-(SP)
     032772  013746  002524                        MOV     ADD,-(SP)
     032776  012746  003552                        MOV     #FOR4,-(SP)
```

```
        033002  012746  000004                  MOV     #4,-(SP)
        033006  010600                           MOV     SP,R0
        033010  104414                           TRAP    C$PNTB
        033012  062706  000012                   ADD     #12,SP
  2348  033016  000417                           BR      40$
  2349                                  ;
  2350                                  ; END ERROR CONDITION
  2351                                  ; PRINT RAM SIZE IF DETERMINED AND NOT PREVIOUSLY PRINTED
  2352                                  ;
  2353  033020  005737  002536          20$:    TST     RAM                     ; SEE IF RAM SIZE ALREADY PRINTED
  2354  033024  001014                           BNE     40$                     ; IF RAM SIZE PRINTED, BRANCH
  2355  033026                                   PRINTB  #FOR9,EXP               ; TELL OPERATOR HOW MUCH RAM PRESENT
        033026  013746  002530                   MOV     EXP,-(SP)
        033032  012746  004227                   MOV     #FOR9,-(SP)
        033036  012746  000002                   MOV     #2,-(SP)
        033042  010600                           MOV     SP,R0
        033044  104414                           TRAP    C$PNTB
        033046  062706  000006                   ADD     #6,SP
  2356  033052  005237  002536                   INC     RAM                     ; SET RAM FLAG MESSAGE PRINTED
  2357                                  ;
  2358                                  ; NOW LOOP THROUGH AND CHECK REST OF RAM
  2359                                  ;
  2360  033056                          40$:
  2361  033056  005201                           INC     R1                      ; INCREMENT COUNTER
  2362  033060  005237  002524                   INC     ADD                     ; INCREMENT RAM ADDRESS POINTER
  2363  033064  017737  147232  002526           MOV     @#VDA,VAL               ; READ NEXT RAM ADDRESS
  2364  033072  010137  002530                   MOV     R1,EXP                  ; GET EXPECTED IN EXP
  2365  033076  005137  002530                   COM     EXP                     ; COMPLEMENT EXPRESION
  2366  033102  023737  002530  002526           CMP     EXP,VAL                 ; COMPARE EXPECTED TO VALUE
  2367  033110  001432                           BEQ     50$                     ; IF OK, BRANCH
  2368                                  ;
  2369                                  ; ERROR CONDITION
  2370                                  ;
  2371  033112  005737  002532                   TST     ERROR1                  ; HAVE ANY ERRORS OCCURED YET?
  2372                                  ;
  2373                                  ; NO PREVIOUS ERRORS
  2374                                  ;
  2375  033116                                   ERRMRD  2,MSG34,ERR5            ; PRINT ERROR HEADER
        033116  104456                           TRAP    C$ERMRD
        033120  000002                           .WORD   2
        033122  006673                           .WORD   MSG34
        033124  012136                           .WORD   ERR5
  2376  033126  022737  000023  002532  43$:    CMP     #19.,ERROR1             ; SEE IF TOO MANY ERRORS TO PRINT
  2377  033134  100416                           BMI     48$                     ; BRANCH IF MORE THEN 20 PRINTED
  2378  033136                                   PRINTB  #FOR4,ADD,VAL,EXP       ; PRINT ERROR LINE
        033136  013746  002530                   MOV     EXP,-(SP)
        033142  013746  002526                   MOV     VAL,-(SP)
        033146  013746  002524                   MOV     ADD,-(SP)
        033152  012746  003552                   MOV     #FOR4,-(SP)
        033156  012746  000004                   MOV     #4,-(SP)
        033162  010600                           MOV     SP,R0
        033164  104414                           TRAP    C$PNTB
        033166  062706  000012                   ADD     #12,SP
  2379  033172  005237  002532          48$:    INC     ERROR1                  ; INCREMENT ERROR COUNTER
  2380                                  ;
  2381                                  ; END OF ERROR ROUTINE, CHECK FOR END OF TEST
  2382                                  ;
```

TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

```
2383 033176
2384 033176  020127  003777          50$:
                                             CMP     R1,#3777                ; END OF TEST
2385 033202  001325                          BNE     40$                     ; BRANCH IF TEST NOT DONE
2386                                         ;
2387                                         ; TEST DONE, PRINT FINAL ERROR TOTALS IF ANY
2388                                         ;
2389 033204  005737  002532                  TST     ERROR1                  ; ANY ERRORS?
2390 033210  001422                          BEQ     60$                     ; IF NO, BRANCH
2391                                         ;
2392                                         ; PRINT FINAL ERROR TOTALS
2393                                         ;
2394 033212                                  PRINTB  #FOR6,ERROR1            ; PRINT NUMBER OF ERRORS
     033212  013746  002532                  MOV     ERROR1,-(SP)
     033216  012746  004054                  MOV     #FOR6,-(SP)
     033222  012746  000002                  MOV     #2,-(SP)
     033226  010600                          MOV     SP,R0
     033230  104414                          TRAP    C$PNTB
     033232  062706  000006                  ADD     #6,SP
2395 033236                                  PRINTB  #FOR0                   ; LINE FEED CARRIAGE RETURN
     033236  012746  003333                  MOV     #FOR0,-(SP)
     033242  012746  000001                  MOV     #1,-(SP)
     033246  010600                          MOV     SP,R0
     033250  104414                          TRAP    C$PNTB
     033252  062706  000004                  ADD     #4,SP
2396 033256                          60$:
2397 033256                                  ENDTST
     033256                          L10116:
     033256  104401                          TRAP    C$ETST
```

TEST31: CHECK QBA INCREMENT

```
                                        .SBTTL  TEST31: CHECK QBA INCREMENT
                                        STATST  <CHECK QBA INCREMENT>
033260  040     103     110     T31MSG::        .ASCIZ  / CHECK QBA INCREMENT/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION
                                        ; CHECKS BUS ADDRESS COUNTER INCREMENT
                                        ;TEST STEPS-
                                        ; LOAD 177776 IN QBA
                                        ; LOAD 1'S IN QDMA BITS 8-13
                                        ; IF U-BRIDGE THEN LOAD BITS 8,9
                                        ; SET QDMA GO AND DMAWT BITS
                                        ; READ QBA, CHECK FOR 0
                                        ; READ QDMA, CHECK BITS 8-13 FOR 0
                                        ; IF U-BRIDGE THEN CHECK BITS 8,9
                                        ;
                                        ;--
                                        ;*****************************************************************************

033306                                  BGNTST
033306                                  T31::
033306                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
033306  104421                          TRAP    C$RFLA
033310  022700  001000                  CMP     @PNT,R0                 ; IS PNT FLAG SET?
033314  001012                          BNE     64$                     ; BRANCH IF NOT SET
033316  012746  033260                  MOV     @T31MSG,-(SP)
033322  012746  002270                  MOV     @PNTMSG,-(SP)
033326  012746  000002                  MOV     #2,-(SP)
033332  010600                          MOV     SP,R0
033334  104417                          TRAP    C$PNTF
033336  062706  000006                  ADD     #6,SP
033342  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
033346                                  INIT
                                        ;INITIALIZE DEVICE
033346  012777  000100  146742          MOV     #QHALT,@QMT1            ;INITIALIZE Q-BRIDGE
033354  013777  002520  146734          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
033362  005077  146712                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
033366  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
                                        ;
                                        ; CHECK QBA INCREMENT
                                        ;
033372  012777  177776  146710          MOV     #177776,@QBA            ; ALL ONES TO QBA
033400  012777  037400  146676          MOV     #37400,@QDMA            ; SET BITS 8-13 IN QDMA
033406  005737  002500                  TST     BRIDGETYPE              ; TEST BRIDGE TYPE
033412  001403                          BEQ     10$                     ; IF EQ THEN Q-BRIDGE
033414  012777  001400  146662          MOV     #01400,@QDMA            ; SET BITS 8,9 IN QDMA (U-BRIDGE)
033422  052777  000003  146654  10$:    BIS     #DMAWT!GO,@QDMA         ; INITIATE DMA TRANSFER
033430  017737  146654  002526          MOV     @QBA,VAL                ; MOVE QBA TO VALUE
033436  012737  000000  002530          MOV     #0,EXP                  ; MOVE EXPECTED TO EXP
033444  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
033452  001410                          BEQ     1$                      ; IF OK, BRANCH
                                        ;
                                        ; ERROR CONDITION
                                        ;
033454  012701  000006                  MOV     #6,R1                   ; REGISTER NUMBER TO R1
033460                                  ERRHRD  1,MSG36,ERR4            ; TELL OPERATOR ERROR
```

TEST31: CHECK QBA INCREMENT

```
        033460  104456                          TRAP    C$ERHRD
        033462  000001                          .WORD   1
        033464  006777                          .WORD   MSG36
        033466  011776                          .WORD   ERR4
  2439  033470  004737  013230                  JSR     PC,DUMP         ; DO OPTIONAL REGISTER DUMP
  2440                                  ;
  2441                                  ; NOW CHECK QDMA BITS 8-13 FOR 0
  2442                                  ;
  2443  033474                          1$:
  2444  033474  017737  146604  002526          MOV     @QDMA,VAL       ; READ QDMA INTO VAL
  2445  033502  042737  140377  002526          BIC     #140377,VAL     ; CLEAR UNTESTED BITS
  2446  033510  005737  002500                  TST     BRIDGETYPE      ; TEST BRIDGE TYPE
  2447  033514  001403                          BEQ     11$             ; IF EQ THEN Q-BRIDGE
  2448  033516  042737  176377  002526          BIC     #176377,VAL     ; CLEAR UNTESTED BITS (U-BRIDGE)
  2449  033524  012737  000000  002530  11$:    MOV     #0,EXP          ; MOVE EXPECTED TO EXP
  2450  033532  023737  002530  002526          CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
  2451  033540  001410                          BEQ     2$              ; IF OK, BRANCH
  2452                                  ;
  2453                                  ; ERROR CONDITION
  2454                                  ;
  2455  033542  012701  000004                  MOV     #4,R1           ; REGISTER NUMBER TO R1
  2456  033546                          .       ERRHRD  2,MSG36.5,ERR4  ; TELL OPERATOR ERROR
        033546  104456                          TRAP    C$ERHRD
        033550  000002                          .WORD   2
        033552  007023                          .WORD   MSG36.5
        033554  011776                          .WORD   ERR4
  2457  033556  004737  013230                  JSR     PC,DUMP         ; DO OPTIONAL REGISTER DUMP
  2458  033562                          2$:
  2459  033562                          .       ENDTST
        033562                          L10117:
        033562  104401                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST32: CHECK QWC OVERFLOW (READ)
                                        STATST  <CHECK QWC OVERFLOW (READ)>
033564      040     103     110         T32MSG::        .ASCIZ  / CHECK QWC OVERFLOW (READ)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT GO WITH DMAWT CAUSES IMMEDIATE DMA REQ
                                        ; CHECKS WC INCREMENT, OVERFLOW
                                        ;TEST STEPS-
                                        ; LOAD QWC WITH 1'S
                                        ; LOAD QBA WITH VALID ADDRESS
                                        ; SET QDMA AND GO BIT
                                        ; READ QWC, CHECK FOR 0
                                        ; CHECK QDMA RDY SET
                                        ;
                                        ;--
                                        ;*****************************************************************************

033620                                  T32::   BGNTST
033620                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
033620  104421                                  TRAP    C$RFLA
033622  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
033626  001012                                  BNE     64$                     ; BRANCH IF NOT SET
033630  012746  033564                          MOV     @T32MSG,-(SP)
033634  012746  002270                          MOV     @PNTMSG,-(SP)
033640  012746  000002                          MOV     #2,-(SP)
033644  010600                                  MOV     SP,R0
033646  104417                                  TRAP    C$PNTF
033650  062706  000006                          ADD     #6,SP
033654  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
033660                                          INIT
                                        ;INITIALIZE DEVICE
033660  012777  000100  146430                  MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
033666  013777  002520  146422                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
033674  005077  146400                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
033700  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
033704  012777  177777  146400                  MOV     #-1,@QWC                ; LOAD QWC WITH 1'S
033712  012777  041660  146370                  MOV     #$PATCH,@QBA            ; LOAD QBA WITH VALID ADDRESS
033720  052777  000001  146356                  BIS     #GO,@QDMA               ; SET QDMA AND GO BIT
033726  017737  146360  002526                  MOV     @QWC,VAL                ; READ QWC
033734  012737  000000  002530                  MOV     #0,EXP                  ; EXPECTED VALUE IN EXP
033742  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
033750  001410                                  BEQ     10$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
033752  012701  000007                          MOV     #7,R1                   ; REGISTER NUMBER TO R1
033756                                          ERRHRD  1,MSG40,ERR4            ; TELL OPERATOR ERROR
033756  104456                                  TRAP    C$ERHRD
033760  000001                                  .WORD   1
033762  007337                                  .WORD   MSG40
033764  011776                                  .WORD   ERR4
033766  004737  013230                          JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
                                        ;
                                        ; END ERROR CONDITION, CHECK FOR RDY SET
```

TEST32: CHE_K QWC OVERFLOW (READ)

```
2497                                            ;
2498 033772                                     10$:
2499 033772  017737  146306  002526                     MOV     @QOMA,VAL               ; READ QDMA REGISTER
2500 034000  032737  000200  002526                     BIT     #RDY,VAL                ; CHECK FOR RDY SET
2501 034006  001006                                     BNE     20$                     ; BRANCH IF OK
2502                                            ;
2503                                            ; ERROR CONDITION
2504                                            ;
2505 034010                                             ERRHRD  2,MSG41                 ; TELL OPERATOR ERROR
     034010  104456                                     TRAP    C$ERHRD
     034012  000002                                     .WORD   2
     034014  007423                                     .WORD   MSG41
     034016  000000                                     .WORD   0
2506 034020  004737  013230                            JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
2507                                            ;
2508                                            ; END ERROR CONDITION
2509                                            ;
2510 034024                                     20$:
2511 034024                                             ENDTST
     034024                                     L10120:
     034024  104401                                     TRAP    C$ETST
```

TEST33: CHECK QWC OVERFLOW (WRITE)

```
                                        .SBTTL  TEST33: CHECK QWC OVERFLOW (WRITE)
                                        STATST  <CHECK QWC OVERFLOW (WRITE)>
034026  040     103     110     T33MSG::        .ASCIZ  / CHECK QWC OVERFLOW (WRITE)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT GO CLEARS RDY BIT
                                        ; CHECKS THAT GO CAUSES DMA REQ
                                        ; CHECKS THAT DMA REQ CAUSES WC INCREMENT
                                        ; CHECKS THAT WC OVERFLOW SETS RDY BIT
                                        ;TEST STEPS
                                        ; LOAD QWC WITH 177776
                                        ; LOAD QBA WITH ANY VALID ADDRESS
                                        ; SET QMT1 FUNC = 5 (SELDMADAT)
                                        ; SET QDMA GO AND DMAWT BITS
                                        ; CHECK QDMA RDY CLEARED
                                        ; SET QDMA GO AND DMAWT BITS
                                        ; READ QWC, CHECK FOR 0
                                        ; CHECK QDMA RDY SET
                                        ;
                                        ;--
                                        ;*****************************************************************************

034062                                          BGNTST
034062                                  T33::
034062                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
034062  104421                                  TRAP    C$RFLA
034064  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
034070  001012                                  BNE     64$             ; BRANCH IF NOT SET
034072  012746  034026                          MOV     @T33MSG,-(SP)
034076  012746  002270                          MOV     @PNTMSG,-(SP)
034102  012746  000002                          MOV     @2,-(SP)
034106  010600                                  MOV     SP,R0
034110  104417                                  TRAP    C$PNTF
034112  062706  000006                          ADD     @6,SP
034116  005037  002532            64$:          CLR     ERROR1          ; CLEAR ERROR COUNTER
034122                                          INIT
                                        ;INITIALIZE DEVICE
034122  012777  000100  146166                  MOV     @#QHALT,@QMT1   ;INITIALIZE Q-BRIDGE
034130  013777  002520  146160                  MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
034136  005077  146136                          CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
034142  005037  002532                          CLR     ERROR1          ;CLEAR ERROR FLAG
034146  012777  177776  146136                  MOV     @177776,@QWC    ; LOAD QWC WITH ALL 1'S
034154  012777  041660  146126                  MOV     @#PATCH,@QBA    ; LOAD QBA WITH VALID ADDRESS
034162  052777  000024  146126                  BIS     @SELDMADAT,@QMT1 ; SEL QMT1 FUNC = 5
034170  052777  000003  146106                  BIS     @GO!DMAWT,@QDMA ; SET QDMA GO BIT
034176  017737  146102  002526                  MOV     @QDMA,VAL       ; READ QDMA REGISTER
034204  032737  000200  002526                  BIT     @RDY,VAL        ; CHECK FOR RDY BIT CLEARED
034212  001406                                  BEQ     10$             ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
034214                                          ERRHRD  1,MSG37         ; TELL OPERATOR ERROR
034214  104456                                  TRAP    C$ERHRD
034216  000001                                  .WORD   1
034220  007103                                  .WORD   MSG37
```

TEST33: CHECK QWC OVERFLOW (WRITE)

```
      034222  000000                          .WORD   0
2550  034224  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
2551                                          ;
2552                                          ; END ERROR CONDITION, CHECK FOR QWC INCREMENT
2553                                          ;
2554  034230                                  10$:
2555  034230  052777  000003  146046          BIS     #GO!DMAWT,@QDMA         ; SET QDMA GO BIT
2556  034236  017737  146050  002526          MOV     @QWC,VAL                ; READ QWC REGISTER
2557  034244  012737  000000  002530          MOV     #0,EXP                  ; ALL ZEROS FOR EXPECTED
2558  034252  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2559  034260  001410                          BEQ     20$                     ; BRANCH IF OK
2560                                          ;
2561                                          ; ERROR CONDITION
2562                                          ;
2563  034262  012701  000007                  MOV     #7,R1                   ; REGISTER NUMBER TO R1
2564  034266                                  ERRHRD  2,MSG38,ERR4            ; TELL OPERATOR ERROR
      034266  104456                          TRAP    C$ERHRD
      034270  000002                          .WORD   2
      034272  007172                          .WORD   MSG38
      034274  011776                          .WORD   ERR4
2565  034276  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
2566                                          ;
2567                                          ; END ERROR CONDITION, CHECK FOR QDMA RDY SET
2568                                          ;
2569  034302                                  20$:
2570  034302  017737  145776  002526          MOV     @QDMA,VAL               ; READ QDMA REGISTER
2571  034310  032737  000200  002526          BIT     #RDY,VAL                ; CHECK RDY BIT FOR SET
2572  034316  001006                          BNE     30$                     ; BRANCH IF OK
2573                                          ;
2574                                          ; ERROR CONDITION
2575                                          ;
2576  034320                                  ERRHRD  3,MSG39                 ; TELL OPERATOR ERROR
      034320  104456                          TRAP    C$ERHRD
      034322  000003                          .WORD   3
      034324  007254                          .WORD   MSG39
      034326  000000                          .WORD   0
2577  034330  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
2578  034334                                  30$:
2579  034334                                  ENDTST
      034334                                  L10121:
      034334  104401                          TRAP    C$ETST
```

TEST34: DMA 1 WORD (WRITE)

```
                                        .SBTTL  TEST34: DMA 1 WORD (WRITE)
                                        STATST  <DMA 1 WORD (WRITE)>
034336  040   104   115                 T34MSG::        .ASCIZ  / DMA 1 WORD (WRITE)/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS DMA FROM MEMORY FROM Q-BUS TO UNIBUS
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; LOAD DATA WORD IN BUFFER
                                        ; LOAD QWC WITH -1
                                        ; LOAD QBA WITH BUFFER ADDRESS
                                        ; SET QDMA DMAWT AND GO BITS
                                        ; SET QMT1 FUNC = 5 (SELDMADAT)
                                        ; READ QMTO, CHECK DATA WORD
                                        ; CLEAR QMT1 FUNC = 5 (SELDMADAT)
                                        ; REPEAT WITH COMPLEMENT DATA PATTERN
                                        ; LOOP 100 TIMES
                                        ;
                                        ;--
                                        ;*****************************************************************************
034362                                          BGNTST
034362                                  T34::
034362                                          CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
034362  104421                                  TRAP    C$RFLA
034364  022700  001000                          CMP     @PNT,R0         ; IS PNT FLAG SET?
034370  001012                                  BNE     64$             ; BRANCH IF NOT SET
034372  012746  034336                          MOV     @T34MSG,-(SP)
034376  012746  002270                          MOV     @PNTMSG,-(SP)
034402  012746  000002                          MOV     @2,-(SP)
034406  010600                                  MOV     SP,R0
034410  104417                                  TRAP    C$PNTF
034412  062706  000006                          ADD     @6,SP
034416  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
034422                                          INIT
                                        ;INITIALIZE DEVICE
034422  012777  000100  145666                  MOV     @MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
034430  013777  002520  145660                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
034436  005077  145636                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
034442  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
034446  012705  000144                          MOV     @100.,R5                ; NUMBER OF LOOPS
034452  012737  125252  002530                  MOV     @125252,EXP             ; STARTING DATA PATTERN TO DMA
034460                                          BGNSUB
034460                                  T34.1:
034460  104402                                  TRAP    C$BSUB
034462                                  L34:    BGNSEG
034462  104404                                  TRAP    C$BSEG
034464  012777  177777  145620                  MOV     @-1,@QWC                ; -1 TO QWC
034472  012777  002530  145610                  MOV     @EXP,@QBA               ; RECEIVING BUFFER TO QBA
034500  052777  000003  145576                  BIS     @DMAWT!GO,@QDMA         ; INITIATE DMA
034506  052777  000024  145602                  BIS     @SELDMADAT,@QMT1        ; SET QMT1 FUNC = 5
034514  017737  145574  002526                  MOV     @QMTO,VAL               ; READ QMTO (RECEIVED DMA DATA)
034522  042777  000024  145566                  BIC     @SELDMADAT,@QMT1        ; CLEAR QMT1 FUNC = 5
034530  023737  002530  002526                  CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
034536  001430                                  BEQ     20$                     ; BRANCH IF OK
```

TEST34: DMA 1 WORD (WRITE)

```
                                        ;
                                        ; ERROR CONDITION
                                        ;
                        ERRHRD  1,MSG42                 ; TELL OPERATOR ERROR
                        TRAP    C$ERHRD
                        .WORD   1
                        .WORD   MSG42
                        .WORD   0
                        MOV     #101.,R4                ; CALCULATE LOOP NUMBER
                        SUB     R5,R4
                        PRINTB  #FOR10,EXP,VAL,R4       ; PRINT OUT RESULTS
                        MOV     R4,-(SP)
                        MOV     VAL,-(SP)
                        MOV     EXP,-(SP)
                        MOV     #FOR10,-(SP)
                        MOV     #4,-(SP)
                        MOV     SP,R0
                        TRAP    C$PNTB
                        ADD     #12,SP
                        JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
                        INC     ERROR1                  ; SET ERROR FLAG
                20$:
                        ENDSEG
                10000$:
                        TRAP    C$ESEG
                        COM     EXP                     ; COMPLEMENT DATA PATTERN
                        DEC     R5                      ; DECREMENT LOOP COUNTER
                        BEQ     30$                     ; BRANCH IF DONE WITH LOOPS
                        TST     ERROR1                  ; CHECK FOR ERRORS
                        BNE     30$                     ; IF ANY ERRORS, END TEST
                        JMP     L34                     ; PREFORM TEST AGAIN
                30$:
                        ENDSUB
                L10123:
                        TRAP    C$ESUB
                        ENDTST
                L10122:
                        TRAP    C$ETST
```

TEST35: DMA 1 WORD (READ)

```
                                .SBTTL  TEST35: DMA 1 WORD (READ)
                               STATST  <DMA 1 WORD (READ)>
034650  040  104  115  T35MSG::        .ASCIZ  / DMA 1 WORD (READ)/

                               ;*****************************************************************************
                               ;++
                               ;TEST DESCRIPTION-
                               ; CHECKS DMA TO MEMORY FROM UNIBUS TO Q-BUS
                               ; USES UDMAENB - QDMAENB LOOP-BACK
                               ; USES QTSOUT - QTSIN LOOP-BACK
                               ;TEST STEPS-
                               ; SET QMT1 FUNC = 7 (SEL DMA READ)
                               ; CLEAR BUFFER
                               ; LOAD QMTO WITH DATA PATTERN
                               ; LOAD QWC WITH -1
                               ; LOAD QBA WITH BUFFER ADDRESS
                               ; SET QDMA GO BITS
                               ; READ DATA IN BUFFER, CHECK
                               ; REPEAT WITH COMPLEMENT DATA PATTERN
                               ; LOOP 100 TIMES
                               ;
                               ;--
                               ;*****************************************************************************

034674                                 BGNTST
034674                         T35::
034674                                 CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
034674  104421                         TRAP    C$RFLA
034676  022700  001000                 CMP     @PNT,R0                 ; IS PNT FLAG SET?
034702  001012                         BNE     64$                     ; BRANCH IF NOT SET
034704  012746  034650                 MOV     @T35MSG,-(SP)
034710  012746  002270                 MOV     @PNTMSG,-(SP)
034714  012746  000002                 MOV     @2,-(SP)
034720  010600                         MOV     SP,R0
034722  104417                         TRAP    C$PNTF
034724  062706  000006                 ADD     @6,SP
034730  005037  002532         64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
034734                                 INIT
                               ;INITIALIZE DEVICE
034734  012777  000100  145354         MOV     @MRQHALT,@QMT1          ;INITIALIZE Q-BRIDGE
034742  013777  002520  145346         MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
034750  005077  145324                 CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
034754  005037  002532                 CLR     ERROR1                  ;CLEAR ERROR FLAG
034760  012705  000144                 MOV     @100.,R5                ; NUMBER OF LOOPS
034764  012737  125252  002530         MOV     @125252,EXP             ; STARTING DATA PATTERN TO DMA
034772                                 BGNSUB
034772                         T35.1:
034772  104402                         TRAP    C$BSUB
034774                         L35:    BGNSEG
034774  104404                         TRAP    C$BSEG
034776  052777  000034  145312         BIS     @SELRDDMA,@QMT1         ; SET QMT1 FUNC = 7
035004  012737  000000  041660         MOV     @0,$PATCH               ; CLEAR BUFFER
035012  013777  002530  145274         MOV     EXP,@QMTO               ; LOAD DATA PATTERN INTO QMTO
035020  012777  177777  145264         MOV     @-1,@QWC                ; -1 TO QWC
035026  012777  041660  145254         MOV     @$PATCH,@QBA            ; RECEIVING BUFFER TO QBA
035034  052777  000001  145242         BIS     @GO,@QDMA               ; INITIATE DMA
035042  013737  041660  002526         MOV     $PATCH,VAL              ; MOVE BUFFER TO VALUE
```

TEST35: DMA 1 WORD (READ)

```
2676 035050  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2677 035056  001430                          BEQ     20$                     ; BRANCH IF OK
2678                                 ;
2679                                 ; ERROR CONDITION
2680                                 ;
2681 035060                                  ERRHRD  1,MSG43                 ; TELL OPERATOR ERROR
     035060  104456                          TRAP    C$ERHRD
     035062  000001                          .WORD   1
     035064  007626                          .WORD   MSG43
     035066  000000                          .WORD   0
2682 035070  012704  000145                  MOV     #101.,R4                ; CALCULATE LOOP NUMBER
2683 035074  160504                          SUB     R5,R4
2684 035076                                  PRINTB  #FOR10,EXP,VAL,R4       ; PRINT OUT RESULTS
     035076  010446                          MOV     R4,-(SP)
     035100  013746  002526                  MOV     VAL,-(SP)
     035104  013746  002530                  MOV     EXP,-(SP)
     035110  012746  004261                  MOV     #FOR10,-(SP)
     035114  012746  000004                  MOV     #4,-(SP)
     035120  010600                          MOV     SP,R0
     035122  104414                          TRAP    C$PNTB
     035124  062706  000012                  ADD     #12,SP
2685 035130  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
2686 035134  005237  002532                  INC     ERROR1                  ; SET ERROR FLAG
2687 035140                          20$:
2688 035140                                  ENDSEG
     035140                          10000$:
     035140  104405                          TRAP    C$ESEG
2689 035142  005137  002530                  COM     EXP                     ; COMPLEMENT DATA PATTERN
2690 035146  005305                          DEC     R5                      ; DECREMENT LOOP COUNTER
2691 035150  001405                          BEQ     30$                     ; BRANCH IF DONE WITH LOOPS
2692 035152  005737  002532                  TST     ERROR1                  ; CHECK FOR ERRORS
2693 035156  001002                          BNE     30$                     ; IF ANY ERRORS, END TEST
2694 035160  000137  034774                  JMP     L35                     ; PREFORM TEST AGAIN
2695 035164                          30$:
2696 035164                                  ENDSUB
     035164                          L10125:
     035164  104403                          TRAP    C$ESUB
2697 035166                                  ENDTST
     035166                          L10124:
     035166  104401                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST36: DMA 1 WORD HI MEMORY (WRITE)
035170                                  STATST  <DMA 1 WORD HI MEMORY (WRITE)>
035170  040   104   115                 T36MSG::        .ASCIZ  / DMA 1 WORD HI MEMORY (WRITE)/

                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS DMA FROM MEMORY FROM Q-BUS HIGHEST AVAILABLE
                                        ;   MEMORY TO UNIBUS
                                        ; USES QTSOUT - QTSIN LOOP-BACK
                                        ;TEST STEPS-
                                        ; CHECK IF > 32K WORD MEMORY
                                        ; IF YES-
                                        ;       SET UP MEMORY MANAGEMENT REGISTERS
                                        ;       GET TOP OF PROGRAM ADDRESS, CALCULATE NEXT 4KW BLOCK
                                        ;       GET TOP THREE BITS OF BLOCK, USE THAT FOR INDEX TO MM REGISTER
                                        ;       THIS ADDRESS STORED (TRANSFER ADDRES``
                                        ;       GET HIGHEST AVAILABLE ADDRESS
                                        ;       SET EXTENDED ADDRESS BITS 16 AND 17 FROM HIGHEST ADDRESS
                                        ;       CONVERT HIGHEST AVAILABLE ADDRESS TO 16 BIT ADDRESS
                                        ;       ENABLE MAPPING
                                        ;       WRITE EXPECTED DATA PATTERN TO MEMORY AT TRANSFER ADDRESS
                                        ;
                                        ; ELSE TRANSFER AT TOP OF BUFFER
                                        ;
                                        ; LOAD QWC WITH -1
                                        ; LOAD QBA WITH BUFFER ADDRESS
                                        ; SET QDMA DMAWT AND GO BITS
                                        ; SET QMT1 FUNC = 5 (SELDMADAT)
                                        ; READ QMT0, CHECK DATA WORD
                                        ; CLEAR QMT1 FUNC = 5 (SELDMADAT)
                                        ; REPEAT WITH COMPLEMENT DATA PATTERN
                                        ; LOOP 100 TIMES
                                        ;--
                                        ;*****************************************************************************

                                                BGNTST
035226                                  T36::
035226                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
035226  104421                                  TRAP    C$RFLA
035230  022700  001000                          CMP     #PNT,R0                 ; IS PNT FLAG SET?
035234  001012                                  BNE     64$                     ; BRANCH IF NOT SET
035236  012746  035170                          MOV     #T36MSG,-(SP)
035242  012746  002270                          MOV     #PNTMSG,-(SP)
035246  012746  000002                          MOV     #2,-(SP)
035252  010600                                  MOV     SP,R0
035254  104417                                  TRAP    C$PNTF
035256  062706  000006                          ADD     #6,SP
035262  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
035266                                          INIT
                                        ;INITIALIZE DEVICE
035266  012777  000100  145022                  MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
035274  013777  002520  145014                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
035302  005077  144772                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
035306  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
035312  012705  000144                          MOV     #100.,R5                ; NUMBER OF LOOPS
035316  012737  125252  002530                  MOV     #125252,EXP             ; STARTING DATA PATTERN TO DMA
```

TEST36: DMA 1 WORD MI MEMORY (WRITE)

```
2739 035324  005002                          CLR     R2
2740 035326                                  BGNSUB
     035326                          T36.1:
     035326  104402                          TRAP    C$BSUB
2741 035330  022737  002000  002120          CMP     #2000,L$MIME            ; > 32K WORD MEMORY?
2742 035336  002404                          BLT     2$                      ; IF YES BRANCH
2743                                 ;
2744                                 ; MEMORY LESS THEN 32KW, NO MEMORY MANAGEMENT SET UP
2745                                 ;
2746 035340  013701  002552                   MOV     FREMEM,R1               ; READ BUFFER ADDRESS
2747 035344  010103                          MOV     R1,R3
2748 035346  000451                          BR      L36                     ; GO DO DMA                     ; V C.0
2749                                 ;
2750                                 ; MEMORY MANAGEMENT IN USE, SET UP TEST
2751                                 ;
2752 035350                          2$:
2753 035350  004737  014370                   JSR     PC,MMINIT               ; SET UP MM REGISTERS
2754 035354  013700  002552                   MOV     FREMEM,R0               ; GET TO TOP OF PROGRAM
2755 035360  062700  020000                   ADD     #20000,R0               ; NEXT 4KW BLOCK
2756 035364  042700  017777                   BIC     #17777,R0
2757 035370  010003                          MOV     R0,R3                   ; SAVE FOR BUFFER ADDRESS
2758 035372  000241                          CLC                             ; CLEAR CARRY FLAG
2759 035374  006100                          ROL     R0                      ; GET TOP 3 BITS
2760 035376  006100                          ROL     R0
2761 035400  006100                          ROL     R0
2762 035402  006100                          ROL     R0
2763 035404  006100                          ROL     R0                      ; TIMES 2 FOR INDEX USE
2764 035406  013760  002120  172340          MOV     L$MIME,KISAR0(R0)       ; INDEX TO MM REGISTER TO USE
2765 035414  005002                          CLR     R2
2766 035416  032737  004000  002120          BIT     #BIT11,L$MIME           ; BIT 11 OF HIGHEST AVAILABLE MEMORY
2767 035424  001402                          BEQ     3$                      ; IF NOT ON
2768 035426  052702  001000                   BIS     #QXAD17,R2              ; SET BIT 17
2769 035432  032737  002000  002120  3$:     BIT     #BIT10,L$MIME           ; BIT 10 OF HIGHEST AVAILABLE MEMORY
2770 035440  001402                          BEQ     4$                      ; IF NOT ON
2771 035442  052702  (00400                  BIS     #QXAD16,R2              ; SET BIT 16
2772 035446  012700  (00006          4$:     MOV     #6,R0
2773 035452  013701  002120                   MOV     L$MIME,R1               ; GET BUS ADDRESS
2774 035456  006301                  5$:     ASL     R1                      ; CHANGE TO 16 BIT ADDRESS
2775 035460  005300                          DEC     R0
2776 035462  001375                          BNE     5$
2777 035464  012737  000001  177572          MOV     #BIT0,MMR0              ; ENABLE MAPPING
2778 035472  013713  002530          L36:    MOV     EXP,(R3)                ; WRITE DATA TO MEMORY          ; V C.0
2779 035476  012777  177777  144606          MOV     #-1,BQMC                ; -1 TO QMC
2780 035504  010177  144600                   MOV     R1,BQBA                 ; XMITTING BUFFER ADDR TO QBA
2781 035510  052702  000003                   BIS     #DMAWT!GO,R2            ; SET UP INITIATE DMA WORD
2782 035514  010277  144564                   MOV     R2,BQDMA                ; INITIATE DMA
2783 035520  052777  000024  144570          BIS     #SELDMADAT,BQMT1        ; SET QMT1 FUNC = 5
2784 035526  017737  144562  002526          MOV     BQMTO,VAL               ; READ QMTO (RECEIVED DMA DATA)
2785 035534  042777  000024  144554          BIC     #SELDMADAT,BQMT1        ; CLEAR QMT1 FUNC = 5
2786 035542  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2787 035550  001430                          BEQ     20$                     ; BRANCH IF OK
2788                                 ;
2789                                 ; ERROR CONDITION
2790                                 ;
2791 035552                                  ERRHRD  1,MSG42.5               ; TELL OPERATOR ERROR
     035552  104456                          TRAP    C$ERHRD
     035554  000001                          .WORD   1
```

TEST36: DMA 1 WORD HI MEMORY (WRITE)

```
                035556  007556                          .WORD   MSG42.5
                035560  000000                          .WORD   0
     2792       035562  012704  000145                  MOV     #101.,R4                ; CALCULATE LOOP NUMBER
     2793       035566  160504                          SUB     R5,R4
     2794       035570                                  PRINTB  #FOR10,EXP,VAL,R4       ; PRINT OUT RESULTS
                035570  010446                          MOV     R4,-(SP)
                035572  013746  002526                  MOV     VAL,-(SP)
                035576  013746  002530                  MOV     EXP,-(SP)
                035602  012746  004261                  MOV     #FOR10,-(SP)
                035606  012746  000004                  MOV     #4,-(SP)
                035612  010600                          MOV     SP,R0
                035614  104414                          TRAP    C$PNTB
                035616  062706  000012                  ADD     #12,SP
     2795       035622  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
     2796       035626                                  EXIT    TST
                035626  104432                          TRAP    C$EXIT
                035630  000026                          .WORD   L10126 .
     2797       035632                          20$:
     2798       035632                                  ENDSUB
                035632                          L10127:
                035632  104403                          TRAP    C$ESUB
     2799       035634  005137  002530                  COM     EXP                     ; COMPLEMENT DATA PATTERN       ; V C.0
     2800       035640  005305                          DEC     R5                      ; DECREMENT LOOP COUNTER                ; V C.0
     2801       035642  001405                          BEQ     30$                     ; BRANCH IF DONE WITH LOOPS     ; V C.0
     2802       035644  000137  035472                  JMP     L36                     ; PREFORM TEST AGAIN            ; V C.0
     2803       035650  012737  000000  177572          MOV     #0,MMR0                 ; DISABLE MAPPING
     2804
     2805       035656                          30$:
     2806       035656                                  ENDTST
                035656                          L10126:
                035656  104401                          TRAP    C$ETST
```

TEST37: CHECK WDTO TIME

```
                                        .SBTTL  TEST37: CHECK WDTO TIME
                                        STATST  <CHECK WDTO TIME>
035660  040  103  110                   T37MSG::        .ASCIZ  / CHECK WDTO TIME/

                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS QCTL BITS
                                        ; CHECKS THAT TIMER IS OPERATING
                                        ; ALLOWS OPERATOR TO CHECK TIMES
                                        ; USES UPWRFAIL - QWDTO LOOP BACK
                                        ;TEST STEPS-
                                        ; SET TIME CODE IN QCTL
                                        ; READ QCTL TIME CODE BITS, CHECK
                                        ; SET QCTL TMRENB
                                        ; READ QCTL TMRENB, CHECK
                                        ; WAIT FOR QINT WPWRFL BIT SET ON TIME OUT
                                        ; RING BELL
                                        ; REPEAT 5 TIMES
                                        ; REPEAT WITH ALL TIMES CODES
                                        ;
                                        ;--
                                        ;*****************************************************************************

035702                                  BGNTST
035702                                  T37::
035702                                  CKPNT                   ; PRINT OUT TEST TITLE IF PNT SET
035702  104421                          TRAP    C$RFLA
035704  022700  001000                  CMP     @PNT,R0         ; IS PNT FLAG SET?
035710  001012                          BNE     64$             ; BRANCH IF NOT SET
035712  012746  035660                  MOV     @T37MSG,-(SP)
035716  012746  002270                  MOV     @PNTMSG,-(SP)
035722  012746  000002                  MOV     @2,-(SP)
035726  010600                          MOV     SP,R0
035730  104417                          TRAP    C$PNTF
035732  062706  000006                  ADD     @6,SP
035736  005037  002532          64$:    CLR     ERROR1          ; CLEAR ERROR COUNTER
035742                                  INIT
                                        ;INITIALIZE DEVICE
035742  012777  000100  144346          MOV     @MQHALT,@QMT1   ;INITIALIZE Q-BRIDGE
035750  013777  002520  144340          MOV     INITWORD,@QMT1  ;SET UP LOOPBACK OPTION
035756  005077  144316                  CLR     @QINT           ;CLEAR QINT AFTER LOOPBACK ENABLED
035762  005037  002532                  CLR     ERROR1          ;CLEAR ERROR FLAG
035766  012701  000004                  MOV     @4,R1           ; R1 WITH NUMBER OF TIME CODES
035772  012702  000005                  MOV     @5,R2           ; R2 WITH NUMBER OF LOOPS PER CODE
035776  012703  036362                  MOV     @TABLE36,R3     ; R3 WITH TIME CODE TABLE ADDRESS
036002  005737  036372                  TST     P37             ; HAS MESSAGE ALREADY BEEN PRINTED
036006  001012                          BNE     L37             ; IF ALREADY PRINTED, SKIP PRINTING
036010                                  PRINTB  @FOR11          ; PRINT ABOUT USING BELLS IN TESTS
036010  012746  004353                  MOV     @FOR11,-(SP)
036014  012746  000001                  MOV     @1,-(SP)
036020  010600                          MOV     SP,R0
036022  104414                          TRAP    C$PNTB
036024  062706  000004                  ADD     @4,SP
036030  005237  036372                  INC     P37             ; INCREMENT PRINTED ABOVE MESSAGE FLAG
036034                          L37:
036034  012777  000000  144236          MOV     @0,@QINT        ; MAKE SURE UPWRFL IS CLEAR
```

TEST37: CHECK WDTO TIME

```
2843 036042  012777  000000  144232          MOV     #0,@QCTL                ; MAKE SURE TIMER ENABLE IS CLEAR
2844 036050  051377  144226                  BIS     (R3),@QCTL              ; SET TIME CODE BITS IN QCTL
2845 036054  017737  144222  002526          MOV     @QCTL,VAL               ; READ QCTL
2846 036062  011337  002530                  MOV     (R3),EXP                ; MOVE EXPECTED TO EXP
2847 036066  005137  002530                  COM     EXP                     ; COMPLEMENT EXP FOR MASK
2848 036072  043737  002530  002526          BIC     EXP,VAL                 ; MASK OUT UNTESTED BITS
2849 036100  005137  002530                  COM     EXP                     ; RESTORE EXPECTED VALUE
2850 036104  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
2851 036112  001410                          BEQ     10$                     ; BRANCH IF OK
2852                                        ;
2853                                        ; ERROR CONDITION
2854                                        ;
2855 036114  012701  000003                  MOV     #3,R1                   ; REGISTER NUMBER TO R1
2856 036120                                  ERRHRD  1,MSG44,ERR4            ; TELL OPERATOR ERROR AND ABORT
     036120  104456                          TRAP    C$ERHRD
     036122  000001                          .WORD   1
     036124  007663                          .WORD   MSG44
     036126  011776                          .WORD   ERR4
2857 036130                                  EXIT    TST                     ; ABORT TEST
     036130  104432                          TRAP    C$EXIT
     036132  000242                          .WORD   L10130-.
2858                                        ;
2859                                        ; END ERROR CONDITION
2860                                        ;
2861 036134                                 10$:
2862 036134  052777  000002  144140          BIS     #TMRENB,@QCTL           ; ENABLE TIMER
2863 036142  017737  144134  002526          MOV     @QCTL,VAL               ; READ QCTL
2864 036150  032737  000002  002526          BIT     #TMRENB,VAL             ; TEST FOR TMRENB BIT SET
2865 036156  001016                          BNE     20$                     ; BRANCH IF OK
2866                                        ;
2867                                        ; ERROR CONDITION
2868                                        ;
2869 036160  012701  000003                  MOV     #3,R1                   ; REGISTER NUMBER TO R1
2870 036164  012737  000002  002530          MOV     #TMRENB,EXP             ; EXPECTED VALUE TO EXP
2871 036172  042737  177775  002526          BIC     #^CTMRENB,VAL           ; CLEAR UNTESTED BIT FOR MESSAGE
2872 036200                                  ERRHRD  2,MSG45,ERR4            ; TELL OPERATOR ERROR AND ABORT
     036200  104456                          TRAP    C$ERHRD
     036202  000002                          .WORD   2
     036204  007775                          .WORD   MSG45
     036206  011776                          .WORD   ERR4
2873 036210                                  EXIT    TST                     ; ABORT TEST
     036210  104432                          TRAP    C$EXIT
     036212  000162                          .WORD   L10130-.
2874                                        ;
2875                                        ; END ERROR CONDITION
2876                                        ; NOW WAIT FOR MAXIMUM OF 8 SECONDS IN .025 SEC INCREMENTS FOR WDTO COMPLETE
2877                                        ;
2878 036214                                 20$:
2879 036214  012704  000360                  MOV     #240.,R4                ; NUMBER OF .025 SEC TO LOOP
2880 036220                                 30$:
2881 036220                                  DELAY   25.                     ; .025 SEC DELAY
     036220  012727  000031                  MOV     #25.,(PC)+
     036224  000000                          .WORD   0
     036226  013727  002116                  MOV     L$DLY,(PC)+
     036232  000000                          .WORD   0
     036234  005367  177772                  DEC     -6(PC)
     036240  001375                          BNE     .-4
```

TEST37: CHECK WDTO TIME

```
        036242  005367  177756                  DEC     -22(PC)
        036246  001367                          BNE     .-20
   2882 036250                                  BREAK                           ; CHECK FOR OPERATOR INPUT
        036250  104422                          TRAP    C$BRK
   2883 036252  032777  010000  144020          BIT     @UPWRFL,@QINT           ; TEST FOR WDTO COMPLETE (BIT SET)
   2884 036260  001012                          BNE     40$                     ; IF TIMER COMPLETE, BRANCH
   2885 036262  005304                          DEC     R4                      ; DECREMENT R4 COUNTER
   2886 036264  001355                          BNE     30$                     ; BRANCH IF 6 SECONDS NOT UP
   2887                                 ;
   2888                                 ; ERROR CONDITION, 8 SECONDS UP WITH NO WDTO COMPLETE
   2889                                 ;
   2890 036266                                  ERRHRD  3,MSG46                 ; TELL OPERATOR ERROR AND ABORT
        036266  104456                          TRAP    C$ERHRD
        036270  000003                          .WORD   3
        036272  010077                          .WORD   MSG46
        036274  000000                          .WORD   0
   2891 036276  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
   2892 036302                                  EXIT    TST                     ; ABORT TEST
        036302  104432                          TRAP    C$EXIT
        036304  000070                          .WORD   L10130-.
   2893                                 ;
   2894                                 ; END ERROR CONDITION
   2895                                 ; WATCH DOG TIMER ENDED PROPERLY, RING BELL AND SET UP FOR LOOPING
   2896                                 ;
   2897 036306                          40$:
   2898 036306                                  PRINTB  @FOR12                  ; RING BELL
        036306  012746  004462                  MOV     @FOR12,-(SP)
        036312  012746  000001                  MOV     #1,-(SP)
        036316  010600                          MOV     SP,R0
        036320  104414                          TRAP    C$PNTB
        036322  062706  000004                  ADD     #4,SP
   2899 036326  005302                          DEC     R2                      ; DECREMENT LOOP COUNTER
   2900 036330  001402                          BEQ     50$                     ; IF DONE, BRANCH
   2901 036332  000137  036034                  JMP     L37                     ; IF NOT DONE, JUMP BACK
   2902                                 ;
   2903                                 ; SET UP FOR NEXT TIME CODE AND CHECK FOR DONE
   2904                                 ;
   2905 036336                          50$:
   2906 036336  062703  000002                  ADD     #2,R3                   ; POINT R3 TO NEXT TIME CODE VALUE
   2907 036342  012702  000005                  MOV     #5,R2                   ; R2 WITH NUMBER OF LOOPS PER CODE
   2908 036346  005301                          DEC     R1                      ; DECREMENT CODE COUNTER
   2909 036350  001402                          BEQ     60$                     ; BRANCH IF DONE
   2910 036352  000137  036034                  JMP     L37                     ; GO TEST NEXT TIME CODE
   2911 036356                          60$:
   2912 036356                                  EXIT    TST
        036356  104432                          TRAP    C$EXIT
        036360  000014                          .WORD   L10130-.
   2913                                 ;
   2914                                 ; TABLE OT TIME CODES
   2915                                 ;
   2916 036362                          TABLE36:
   2917 036362  000000                          .WORD   TIME.5                  ; 1/2 SECOND TIME CODE
   2918 036364  000004                          .WORD   TIME1                   ; 1 SECOND TIME CODE
   2919 036366  000010                          .WORD   TIME2                   ; 2 SECONDS TIME CODE
   2920 036370  000014                          .WORD   TIME4                   ; 4 SECONDS TIME CODE
   2921 036372  000000                  P37:    .WORD   0                       ; FLAG FOR MESSAGE ABOUT BELL
   2922 036374                                  ENDTST
```

SEQ 0149

TEST37· CHECK WDTO TIME

```
036374                          L10130:
036374  104401                          TRAP    C$ETST
```

TEST38: CHECK WDTO RESET

```
                                        .SBTTL  TEST38: CHECK WDTO RESET
                                        STATST  <CHECK WDTO RESET>
036376  040     103     110     T38MSG::        .ASCIZ  / CHECK WDTO RESET/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT TIMER CAN BE RESET
                                        ; USES UPWRFAIL - QWDTO LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET QCTL TMRENB
                                        ; WAIT < 10 MILLISECONDS
                                        ; CHECK QINT WPWRFL FOR 0
                                        ; SET QCTL WDTRST BIT
                                        ; REPEAT 1000 TIMES
                                        ;
                                        ;--
                                        ;*****************************************************************************
036420                                  BGNTST
036420                                  T38::
036420                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
036420  104421                          TRAP    C$RFLA
036422  022700  001000                  CMP     #PNT,R0                 ; IS PNT FLAG SET?
036426  001012                          BNE     64$                     ; BRANCH IF NOT SET
036430  012746  036376                  MOV     #T38MSG,-(SP)
036434  012746  002270                  MOV     #PNTMSG,-(SP)
036440  012746  000002                  MOV     #2,-(SP)
036444  010600                          MOV     SP,R0
036446  104417                          TRAP    C$PNTF
036450  062706  000006                  ADD     #6,SP
036454  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
036460                                  INIT
                                        ;INITIALIZE DEVICE
036460  012777  000100  143630          MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
036466  013777  002520  143622          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
036474  005077  143600                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
036500  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
036504  012701  001750                  MOV     #1000.,R1               ; NUMBER OF LOOPS IN TEST
036510                                  BGNSUB
036510                                  T38.1:
036510  104402                          TRAP    C$BSUB
036512  052777  000002  143562  L38:    BIS     #TMRENB,@QCTL           ; SET TMRENB BIT IN QCTL
036520                                  DELAY   3                       ; WAIT ABOUT 7.5 MS
036520  012727  000003                  MOV     #3,(PC)+
036524  000000                          .WORD   0
036526  013727  002116                  MOV     L$DLY,(PC)+
036532  000000                          .WORD   0
036534  005367  177772                  DEC     -6(PC)
036540  001375                          BNE     .-4
036542  005367  177756                  DEC     -22(PC)
036546  001367                          BNE     .-20
036550  032777  010000  143522          BIT     #UPWRFL,@QINT           ; CHECK UPWRFL FOR CLEAR
036556  001410                          BEQ     10$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
```

TEST38: CHECK WDTO RESET

```
2954 036560                                        ERRHRD  1,MSG47          ; TELL OPERATOR ERROR
     036560  104456                               TRAP    C$ERHRD
     036562  000001                               .WORD   1
     036564  010140                               .WORD   MSG47
     036566  000000                               .WORD   0
2955 036570  004737  013230                       JSR     PC,DUMP          ; DO OPTIONAL REGISTER DUMP
2956 036574                                        EXIT    TST              ; ABORT TEST IF ERROR
     036574  104432                               TRAP    C$EXIT
     036576  000016                               .WORD   L10131-.
2957                                       ;
2958                                       ; END ERROR CONDITION
2959                                       ;
2960 036600                                10$:
2961 036600  052777  000001  143474               BIS     #WDTRST,@QCTL    ; SET WDTRST BIT
2962 036606                                        ENDSUB
     036606                                L10132:
     036606  104403                               TRAP    C$ESUB
2963 036610  005301                               DEC     R1               ; DECREMENT LOOP COUNTER
2964 036612  001337                               BNE     L38              ; IF NOT DONE, LOOP
2965 036614                                        ENDTST
     036614                                L10131:
     036614  104401                               TRAP    C$ETST
```

```
                                        .SBTTL  TEST39: CHECK VNTACK INTERRUPT
                                       STATST  <CHECK VNTACK INTERRUPT>
036616  040     103     110           T39MSG::        .ASCIZ  / CHECK VNTACK INTERRUPT/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCIPTION-
                                        ; CHECKS THAT VNTACK SETTING CAN CAUSE INTERRUPT
                                        ; USES QCMDACK - UEVENTACK LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET INTERRUPT VECTOR
                                        ; SET QVNT ACKIE BIT
                                        ; CHECK FOR BIT SET
                                        ; SET QCMD CMDACK BIT
                                        ; CHECK FOR IMMEDIATE INTERRUPT
                                        ; RESTORE INTERRUPT VECTOR
                                        ;
                                        ;--
                                        ;*****************************************************************************

036646                                          BGNTST
036646                                  T39::
036646                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
036646  104421                                  TRAP    C$RFLA
036650  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
036654  001012                                  BNE     64$                     ; BRANCH IF NOT SET
036656  012746  036616                          MOV     @T39MSG,-(SP)
036662  012746  002270                          MOV     @PNTMSG,-(SP)
036666  012746  000002                          MOV     @2,-(SP)
036672  010600                                  MOV     SP,R0
036674  104417                                  TRAP    C$PNTF
036676  062706  000006                          ADD     @6,SP
036702  005037  002532          64$:    CLR     ERROR1                          ; CLEAR ERROR COUNTER
036706                                          INIT
                                        ;INITIALIZE DEVICE
036706  012777  000100  143402                  MOV     @MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
036714  013777  002520  143374                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
036722  005077  143352                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
036726  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
036732                                          SETVEC  VECTOR,#L39,#PRI07      ; SET INTERRUPT VECTOR
036732  012746  000340                          MOV     @PRI07,-(SP)
036736  012746  037060                          MOV     @L39,-(SP)
036742  013746  002512                          MOV     VECTOR,-(SP)
036746  012746  000003                          MOV     @3,-(SP)
036752  104437                                  TRAP    C$SVEC
036754  062706  000010                          ADD     @10,SP
036760  052777  000100  143310                  BIS     @ACKIE,@QVNT            ; SET ACKIE BIT
036766  017737  143304  002526                  MOV     @QVNT,VAL               ; READ QVNT
036774  032737  000100  002526                  BIT     @ACKIE,VAL              ; IS ACKIE BIT REALLY SET
037002  001012                                  BNE     10$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
037004  012701  000001                          MOV     @1,R1                   ; REGISTER NUMBER TO R1
037010  012737  000100  002530                  MOV     @ACKIE,EXP              ; EXPECTED PATTERN TO EXP
037016                                          ERRHRD  1,MSG48,ERR4            ; TELL OPERATOR ERROR
037016  104456                                  TRAP    C$ERHRD
```

TEST39: CHECK VNTACK INTERRUPT

```
        037020  000001                          .WORD   1
        037022  010177                          .WORD   MSG48
        037024  011776                          .WORD   ERR4
   3000 037026  000417                          BR      E39                     ; BRANCH TO END
   3001                                 ;
   3002                                 ; END ERROR CONDITION
   3003                                 ;
   3004 037030                          10$:
   3005 037030  052777  000001  143236          BIS     #CMDACK,@QCMD           ; SET QMDACK BIT, SHOULD INTERRUPT NOW
   3006 037036  000240                          NOP                             ; ALLOW SMALL AMOUNT OF TIME
   3007 037040  000240                          NOP
   3008                                 ;
   3009                                 ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
   3010                                 ;
   3011 037042                          ERRHRD  2,MSG49                         ; TELL OPERATOR ERROR
        037042  104456                  TRAP    C$ERHRD
        037044  000002                  .WORD   2
        037046  010247                  .WORD   MSG49
        037050  000000                  .WORD   0
   3012 037052  004737  013230          JSR     PC,DUMP                         ; DO OPTIONAL REGISTER DUMP
   3013 037056  000403                  BR      E39                             ; SKIP INTERRUPT SERVICE ROUTINE
   3014                                 ;
   3015                                 ; END ERROR CONDITION
   3016                                 ; INTERRUPT SERVICE ROUTINE
   3017                                 ;
   3018 037060                          BGNSRV  L39
        037060                          L39::
   3019 037060  012716  037066          MOV     #E39,(SP)
   3020 037064                          ENDSRV
        037064                          L10134:
        037064  000002                  RTI
   3021                                 ;
   3022                                 ; END INTERRUPT SERVICE ROUTINE
   3023                                 ;
   3024 037066                          E39:
   3025 037066                          CLRVEC  VECTOR                          ; CLEAR VECTOR
        037066  013700  002512          MOV     VECTOR,R0
        037072  104436                  TRAP    C$CVEC
   3026 037074                          ENDTST
        037074                          L10133:
        037074  104401                  TRAP    C$ETST
```

TEST40: CHECK CMDPRS INTERRUPT

```
                                        .SBTTL  TEST40: CHECK CMDPRS INTERRUPT
                                        STATST  <CHECK CMDPRS INTERRUPT>
037076  040  103  110                   T40MSG::        .ASCIZ  / CHECK CMDPRS INTERRUPT/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCIPTION-
                                        ; CHECKS THAT CMDPRS SETTING CAN CAUSE INTERRUPT
                                        ; USES QEVENTPRS - UCMDPRS LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET INTERRUPT VECTOR
                                        ; SET QCMD PRSIE BIT
                                        ; CHECK FOR BIT SET
                                        ; SET QVNT VNTPRS BIT
                                        ; CHECK FOR IMMEDIATE INTERRUPT
                                        ; RESTORE INTERRUPT VECTOR
                                        ;
                                        ;--
                                        ;*****************************************************************************
                                                BGNTST
                                        T40::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
037126  104421                                  TRAP    C$RFLA
037130  022700  001000                          CMP     #PNT,R0                 ; IS PNT FLAG SET?
037134  001012                                  BNE     64$                     ; BRANCH IF NOT SET
037136  012746  037076                          MOV     #T40MSG,-(SP)
037142  012746  002270                          MOV     #PNTMSG,-(SP)
037146  012746  000002                          MOV     #2,-(SP)
037152  010600                                  MOV     SP,R0
037154  104417                                  TRAP    C$PNTF
037156  062706  000006                          ADD     #6,SP
037162  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
037166                                          INIT
                                        ;INITIALIZE DEVICE
037166  012777  000100  143122                  MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
037174  013777  002520  143114                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
037202  005077  143072                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
037206  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
037212                                          SETVEC  VECTOR,#L40,#PRI07      ; SET INTERRUPT VECTOR
037212  012746  000340                          MOV     #PRI07,-(SP)
037216  012746  037340                          MOV     #L40,-(SP)
037222  013746  002512                          MOV     VECTOR,-(SP)
037226  012746  000003                          MOV     #3,-(SP)
037232  104437                                  TRAP    C$SVEC
037234  062706  000010                          ADD     #10,SP
037240  012777  000100  143026                  MOV     #PRSIE,@QCMD            ; SET QCMD PRSIE BIT
037246  017737  143022  002526                  MOV     @QCMD,VAL               ; READ QCMD
037254  032737  000100  002526                  BIT     #PRSIE,VAL              ; IS PRSIE BIT SET
037262  001012                                  BNE     10$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
037264  012701  000000                          MOV     #0,R1                   ; REGISTER NUMBER TO R1
037270  012737  000100  002530                  MOV     #PRSIE,EXP              ; EXPECTED PATTERN TO EXP
037276                                          ERRHRD  1,MSG50,ERR4            ; TELL OPERATOR ERROR
037276  104456                                  TRAP    C$ERHRD
```

TEST40: CHECK CMDPRS INTERRUPT

```
        037300  000001                          .WORD   1
        037302  010337                          .WORD   MSG50
        037304  011776                          .WORD   ERR4
   3061 037306  000417                          BR      E40                     ; BRANCH TO END
   3062                                 ;
   3063                                 ; END ERROR CONDITION
   3064                                 ;
   3065 037310                          10$:
   3066 037310  052777  000001  142760          BIS     #VNTPRS,@QVNT           ; SET VNTPRS BIT, SHOULD INTERRUPT NOW
   3067 037316  000240                          NOP                             ; ALLOW SMALL AMOUNT OF TIME
   3068 037320  000240                          NOP
   3069                                 ;
   3070                                 ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
   3071                                 ;
   3072 037322                          ERRHRD  2,MSG51                         ; TELL OPERATOR ERROR
        037322  104456                  TRAP    C$ERHRD
        037324  000002                  .WORD   2
        037326  010407                  .WORD   MSG51
        037330  000000                  .WORD   0
   3073 037332  004737  013230          JSR     PC,DUMP                         ; DO OPTIONAL REGISTER DUMP
   3074 037336  000403                  BR      E40                             ; SKIP INTERRUPT SERVICE ROUTINE
   3075                                 ;
   3076                                 ; END ERROR CONDITION
   3077                                 ; INTERRUPT SERVICE ROUTINE
   3078                                 ;
   3079 037340                          BGNSRV  L40
        037340                          L40::
   3080 037340  012716  037346          MOV     #E40,(SP)
   3081 037344                          ENDSRV
        037344                          L10136:
        037344  000002                  RTI
   3082                                 ;
   3083                                 ; END INTERRUPT SERVICE ROUTINE
   3084                                 ;
   3085 037346                          E40:
   3086 037346                          CLRVEC  VECTOR                          ; CLEAR VECTOR
        037346  013700  002512          MOV     VECTOR,R0
        037352  104436                  TRAP    C$CVEC
   3087 037354                          ENDTST
        037354                          L10135:
        037354  104401                  TRAP    C$ETST
```

TEST41: CHECK RDY INTERRUPT

```
                                        .SBTTL  TEST41: CHECK RDY INTERRUPT
                                        STATST  <CHECK RDY INTERRUPT>
037356  040     103     110     T41MSG::        .ASCIZ  / CHECK RDY INTERRUPT/
                                        ;*****************************************************************************
                                        ;++
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT RDY BIT SET CAUSES INTERRUPT
                                        ;TEST STEPS-
                                        ; SET INTERRUPT VECTOR
                                        ; SET QDMA RDYIE BIT
                                        ; CHECK FOR BIT SET
                                        ; SET QWC = -1
                                        ; SET QBA = (ANY LEGAL ADDRESS)
                                        ; SET QDMA DMAWT, GO BITS
                                        ; CHECK FOR IMMEDIATE INTERRUPT
                                        ; RESTORE INTERRUPT VECTOR
                                        ;
                                        ;--
                                        ;*****************************************************************************

037404                                  BGNTST
037404                          T41::
037404                                  CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
037404  104421                          TRAP    C$RFLA
037406  022700  001000                  CMP     #PNT,R0                 ; IS PNT FLAG SET?
037412  001012                          BNE     64$                     ; BRANCH IF NOT SET
037414  012746  037356                  MOV     #T41MSG,-(SP)
037420  012746  002270                  MOV     #PNTMSG,-(SP)
037424  012746  000002                  MOV     #2,-(SP)
037430  010600                          MOV     SP,R0
037432  104417                          TRAP    C$PNTF
037434  062706  000006                  ADD     #6,SP
037440  005037  002532          64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
037444                                  INIT
                                        ;INITIALIZE DEVICE
037444  012777  000100  142644          MOV     #MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
037452  013777  002520  142636          MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
037460  005077  142614                  CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
037464  005037  002532                  CLR     ERROR1                  ;CLEAR ERROR FLAG
037470                                  SETVEC  VECTOR,#L41,#PRI07      ; SET INTERRUPT VECTOR
037470  012746  000340                  MOV     #PRI07,-(SP)
037474  012746  037632                  MOV     #L41,-(SP)
037500  013746  002512                  MOV     VECTOR,-(SP)
037504  012746  000003                  MOV     #3,-(SP)
037510  104437                          TRAP    C$SVEC
037512  062706  000010                  ADD     #10,SP
037516  052777  000100  142560          BIS     #RDYIE,@QDMA            ; SET QDMA RDYIE BIT
037524  017737  142554  002526          MOV     @QDMA,VAL               ; READ QDMA INTO VAL
037532  032737  000100  002526          BIT     #RDYIE,VAL              ; IS RDYIE BIT REALLY SET
037540  001012                          BNE     10$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
037542  012701  000004                  MOV     #4,R1                   ; REGISTER NUMBER TO R1
037546  012737  000100  002530          MOV     #RDYIE,EXP              ; EXPECTED PATTERN TO EXP
037554                                  ERRHRD  1,MSG52,ERR4            ; TELL OPERATOR ERROR
```

TEST41: CHECK RDY INTERRUPT

```
        037554  104456                          TRAP    C$ERMRD
        037556  000001                          .WORD   1
        037560  010477                          .WORD   MSG52
        037562  011776                          .WORD   ERR4
  3123  037564  000425                          BR      E41                     ; BRANCH TO END
  3124                                  ;
  3125                                  ; END ERROR CONDITION
  3126                                  ;
  3127  037566                          10$:
  3128  037566  012777  177777  142516          MOV     #-1,@QWC                ; SET QWC = -1
  3129  037574  012777  041660  142506          MOV     #$PATCH,@QBA            ; LOAD QBA WITH ANY ADDRESS     ;VD.2
  3130  037602  052777  000003  142474          BIS     #GO!DMAWT,@QDMA         ; CAUSE INTERRUPT
  3131  037610  000240                          NOP                             ; TIME DELAY
  3132  037612  000240                          NOP
  3133                                  ;
  3134                                  ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
  3135                                  ;
  3136  037614                                  ERRMRD  2,MSG53                 ; TELL OPERATOR ERROR
        037614  104456                          TRAP    C$ERMRD
        037616  000002                          .WORD   2
        037620  010547                          .WORD   MSG53
        037622  000000                          .WORD   0
  3137  037624  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
  3138  037630  000403                          BR      E41                     ; BRANCH OVER INTERRUPT AREA
  3139                                  ;
  3140                                  ; END ERROR CONDITION
  3141                                  ; INTERRUPT SERVICE ROUTINE
  3142                                  ;
  3143  037632                                  BGNSRV  L41
        037632                          L41::
  3144  037632  012716  037640                  MOV     #E41,(SP)                       ; ADJUST STACK
  3145  037636                                  ENDSRV
        037636                          L10140:
        037636  000002                          RTI
  3146  037640                          E41:
  3147  037640                                  CLRVEC  VECTOR                  ; CLEAR VECTOR
        037640  013700  002512                  MOV     VECTOR,R0
        037644  104436                          TRAP    C$CVEC
  3148  037646                                  ENDTST
        037646                          L10137:
        037646  104401                          TRAP    C$ETST
```

TEST42: CHECK ERR INTERRUPT (UNEX)

```
                                        .SBTTL  TEST42: CHECK ERR INTERRUPT (UNEX)
                                        STATST  <CHECK ERR INTERRUPT (UNEX)>
037650  040  103  110                   T42MSG::        .ASCIZ  / CHECK ERR INTERRUPT (UNEX)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT UNEX CAN CAUSE INTERRUPT
                                        ; USES QPURFL - UNEX LOOP-BACK
                                        ;TEST STEPS-
                                        ; SET INTERRUPT VECTOR
                                        ; SET QINT ERRIE BIT, CHECK
                                        ; SET QMT1 MOPFL BIT
                                        ; CHECK FOR IMMEDIATE INTERRUPT
                                        ; CHECK QINT ERR, UNEX BITS
                                        ; CLEAR QMT1 MOPFL BIT
                                        ; RESTORE INTERRUPT VECTOR
                                        ;
                                        ;--
                                        ;*****************************************************************************
                                                BGNTST
                                        T42::
                                                CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
037704  104421                                  TRAP    C$RFLA
037706  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
037712  001012                                  BNE     64$                     ; BRANCH IF NOT SET
037714  012746  037650                          MOV     @T42MSG,-(SP)
037720  012746  002270                          MOV     @PNTMSG,-(SP)
037724  012746  000002                          MOV     #2,-(SP)
037730  010600                                  MOV     SP,R0
037732  104417                                  TRAP    C$PNTF
037734  062706  000006                          ADD     #6,SP
037740  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
                                                INIT
                                        ;INITIALIZE DEVICE
037744  012777  000100  142344                  MOV     @#QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
037752  013777  002520  142336                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
037760  005077  142314                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
037764  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
                                                SETVEC  VECTOR,#L42,#PRI07      ; SET INTERRUPT VECTOR
037770  012746  000340                          MOV     #PRI07,-(SP)
037774  012746  040116                          MOV     #L42,-(SP)
040000  013746  002512                          MOV     VECTOR,-(SP)
040004  012746  000003                          MOV     #3,-(SP)
040010  104437                                  TRAP    C$SVEC
040012  062706  000010                          ADD     #10,SP
040016  052777  000100  142254                  BIS     #ERRIE,@QINT            ; SET QINT ERRIE BIT
040024  017737  142250  002526                  MOV     @QINT,VAL               ; READ QINT INTO VAL
040032  032737  000100  002526                  BIT     #ERRIE,VAL              ; IS ERRIE BIT REALLY SET
040040  001012                                  BNE     10$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
040042  012701  000002                          MOV     #2,R1                   ; REGISTER NUMBER TO R1
040046  012737  000100  002530                  MOV     #ERRIE,EXP              ; EXPECTED PATTERN TO EXP
040054                                          ERRHRD  1,MSG54,ERR4            ; TELL OPERATOR ERROR
```

TEST42: CHECK ERR INTERRUPT (UNEX)

```
        040054  104456                          TRAP    C$ERHRD
        040056  000001                          .WORD   1
        040060  010653                          .WORD   MSG54
        040062  011776                          .WORD   ERR4
  3184  040064  000450                          BR      E42                     ; BRANCH TO END
  3185                                  ;
  3186                                  ; END ERROR CONDITION
  3187                                  ;
  3188  040066                          10$:
  3189  040066  052777  000040  142222          BIS     #MQPFL,@QMT1            ; ENABLE INTERRUPT
  3190  040074  000240                          NOP                             ; TIME DELAY
  3191  040076  000240                          NOP
  3192                                  ;
  3193                                  ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
  3194                                  ;
  3195  040100                                  ERRHRD  2,MSG55                 ; TELL OPERATOR ERROR
        040100  104456                          TRAP    C$ERHRD
        040102  000002                          .WORD   2
        040104  010723                          .WORD   MSG55
        040106  000000                          .WORD   0
  3196  040110  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
  3197  040114  000434                          BR      E42                     ; BRANCH OVER INTERRUPT AREA
  3198                                  ;
  3199                                  ; END ERROR CONDITION
  3200                                  ; INTERRUPT SERVICE ROUTINE
  3201                                  ;
  3202  040116                                  BGNSRV  L42
        040116                          L42::
  3203  040116  017737  142156  002526          MOV     @QINT,VAL               ; READ QINT REGISTER
  3204  040124  012737  140000  002530          MOV     #UNEX!ERR,EXP           ; EXPECTED PATTERN TO EXP
  3205  040132  005137  002530                  COM     EXP                     ; COMPLEMENT EXP FOR CLEARING
  3206  040136  043737  002530  002526          BIC     EXP,VAL                 ; CLEAR UNTESTED BITS
  3207  040144  005137  002530                  COM     EXP                     ; RESTORE EXPECTED PATTERN
  3208  040150  023737  002530  002526          CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
  3209  040156  001410                          BEQ     20$                     ; BRANCH IF OK
  3210                                  ;
  3211                                  ; ERROR CONDITION
  3212                                  ;
  3213  040160  012701  000002                  MOV     #2,R1                   ; REGISTER NUMBER TO R1
  3214  040164                                  ERRHRD  3,MSG56,ERR4            ; TELL OPERATOR ERROR
        040164  104456                          TRAP    C$ERHRD
        040166  000003                          .WORD   3
        040170  011012                          .WORD   MSG56
        040172  011776                          .WORD   ERR4
  3215  040174  004737  013230                  JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
  3216                                  ;
  3217                                  ; END ERROR CONDITION
  3218                                  ;
  3219  040200  012716  040206          20$:    MOV     #E42,(SP)                       ; ADJUST STACK
  3220  040204                                  ENDSRV
        040204                          L10142:
        040204  000002                          RTI
  3221  040206                          E42:
  3222  040206  042777  000040  142102          BIC     #MQPFL,@QMT1            ; CLEAR MQPFL BIT IN QMT1
  3223  040214                                  CLRVEC  VECTOR                  ; CLEAR VECTOR
        040214  013700  002512                  MOV     VECTOR,R0
        040220  104436                          TRAP    C$CVEC
```

TEST42: CHECK ERR INTERRUPT (UNEX)

```
3224 040222                                ENDTST
     040222                          L10141:
     040222  104401                         TRAP    C$ETST
```

TEST43: CHECK ERR INTERRUPT (QNEX)

```
                                        .SBTTL  TEST43: CHECK ERR INTERRUPT (QNEX)
                                        STATST  <CHECK ERR INTERRUPT (QNEX)>
040224  040     103     110     T43MSG::        .ASCIZ  / CHECK ERR INTERRUPT (QNEX)/
                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT QNEX CAN CAUSE INTERRUPT
                                        ;TEST STEPS-
                                        ; SET INTERRUPT VECTOR
                                        ; SET QINT ERRIE BIT
                                        ; SET QWC = -1
                                        ; SET QBA = 160000 (ILLEGAL ADDRESS)
                                        ; SET QDMA = 37400
                                        ; SET QDMA DMAWT, GO BITS
                                        ; CHECK FOR IMMEDIATE INTERRUPT (CAN BE 10 MICROSEC.)
                                        ; CHECK QINT ERR, QNEX BITS
                                        ; RESTORE INTERRUPT VECTOR
                                        ;
                                        ;--
                                        ;*****************************************************************************

040260                                          BGNTST
040260                                  T43::
040260                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
040260  104421                                  TRAP    C$RFLA
040262  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
040266  001012                                  BNE     64$                     ; BRANCH IF NOT SET
040270  012746  040224                          MOV     @T43MSG,-(SP)
040274  012746  002270                          MOV     @PNTMSG,-(SP)
040300  012746  000002                          MOV     @2,-(SP)
040304  010600                                  MOV     SP,R0
040306  104417                                  TRAP    C$PNTF
040310  062706  000006                          ADD     @6,SP
040314  005037  002532          64$:    CLR     ERROR1                          ; CLEAR ERROR COUNTER
040320                                          INIT
                                        ;INITIALIZE DEVICE
040320  012777  000100  141770                  MOV     @MQHALT,@QMT1           ;INITIALIZE Q-BRIDGE
040326  013777  002520  141762                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
040334  005077  141740                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
040340  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
040344                                          SETVEC  VECTOR,#L43,#PRIO7      ; SET INTERRUPT VECTOR
040344  012746  000340                          MOV     @PRIO7,-(SP)
040350  012746  040454                          MOV     @L43,-(SP)
040354  013746  002512                          MOV     VECTOR,-(SP)
040360  012746  000003                          MOV     @3,-(SP)
040364  104437                                  TRAP    C$SVEC
040366  062706  000010                          ADD     @10,SP
040372  052777  000100  141700                  BIS     @ERRIE,@QINT            ; SET ERRIE BIT IN QINT
040400  012777  177777  141704                  MOV     @-1,@QWC                ; ALL 1'S TO QWC
040406  012777  160000  141674                  MOV     @160000,@QBA            ; ILLEGAL ADDRESS 160000 TO QBA
040414  012777  037400  141662                  MOV     @37400,@QDMA            ; 37400 TO QDMA
040422  052777  000003  141654                  BIS     @DMAWT!GO,@QDMA         ; INITIATE INTERRUPT
040430  000240                                  NOP                             ; SMALL TIME DELAY
040432  000240                                  NOP
040434  000240                                  NOP
                                        ;
```

TEST43: CHECK ERR INTERRUPT (QNEX)

```
                                        ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
                                        ;
040436                                          ERRHRD  1,MSG57                 ; TELL OPERATOR ERROR
040436  104456                                 TRAP    C$ERHRD
040440  000001                                 .WORD   1
040442  011101                                 .WORD   MSG57
040444  000000                                 .WORD   0
040446  004737  013230                         JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
040452  000434                                 BR      E43                     ; BRANCH OVER INTERRUPT AREA
                                        ;
                                        ; END ERROR CONDITION
                                        ; INTERRUPT SERVICE ROUTINE
                                        ;
                                                BGNSRV  L43
040454                                  L43::
040454  017737  141620  002526                   MOV     @QINT,VAL               ; READ QINT REGISTER
040462  012737  120000  002530                   MOV     #QNEX!ERR,EXP           ; EXPECTED PATTERN TO EXP
040470  005137  002530                         COM     EXP                     ; COMPLEMENT EXP FOR CLEARING
040474  043737  002530  002526                 BIC     EXP,VAL                 ; CLEAR UNTESTED BITS
040502  005137  002530                         COM     EXP                     ; RESTORE EXPECTED PATTERN
040506  023737  002530  002526                 CMP     EXP,VAL                 ; COMPARE EXPECTED WITH VALUE
040514  001410                                 BEQ     20$                     ; BRANCH IF OK
                                        ;
                                        ; ERROR CONDITION
                                        ;
040516  012701  000002                         MOV     #2,R1                   ; REGISTER NUMBER TO R1
040522                                          ERRHRD  2,MSG58,ERR4            ; TELL OPERATOR ERROR
040522  104456                                 TRAP    C$ERHRD
040524  000002                                 .WORD   2
040526  011205                                 .WORD   MSG58
040530  011776                                 .WORD   ERR4
040532  004737  013230                         JSR     PC,DUMP                 ; DO OPTIONAL REGISTER DUMP
                                        ;
                                        ; END ERROR CONDITION
                                        ;
040536  012716  040544                 20$:    MOV     #E43,(SP)                       ; ADJUST STACK
040542                                          ENDSRV
040542                                 L10144:
040542  000002                                 RTI
040544                                 E43:
040544  042777  000040  141544                 BIC     #MQPFL,@QMT1            ; CLEAR MQPFL BIT IN QMT1
040552                                          CLRVEC  VECTOR                  ; CLEAR VECTOR
040552  013700  002512                         MOV     VECTOR,R0
040556  104436                                 TRAP    C$CVEC
040560                                          ENDTST
040560                                 L10143:
040560  104401                                 TRAP    C$ETST
```

TEST44: CHECK ERR INTERRUPT (UPWRFL)

```
                                        .SBTTL  TEST44: CHECK ERR INTERRUPT (UPWRFL)
040562                                  STATST  <CHECK ERR INTERRUPT (UPWRFL)>
040562      040     103     110         T44MSG::        .ASCIZ  / CHECK ERR INTERRUPT (UPWRFL)/

                                        ;*****************************************************************************
                                        ;**
                                        ;TEST DESCRIPTION-
                                        ; CHECKS THAT UPWRFL CAN CAUSE INTERRUPT
                                        ; USED WDTO - UPWFL LOOP-BACK
                                        ;TEST STEPS
                                        ; SET INTERRUPT VECTOR
                                        ; SET QINT ERRIE BIT
                                        ; SET QCTL TMRENB
                                        ; WAIT FOR INTERRUPT ON TIME-OUT
                                        ; CHECKS QINT UPWRFL, ERR BITS
                                        ; RESTORE INTERRUPT VECTOR
                                        ;
                                        ;--
                                        ;*****************************************************************************

040620                                          BGNTST
040620                                  T44::
040620                                          CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
040620  104421                                  TRAP    C$RFLA
040622  022700  001000                          CMP     @PNT,R0                 ; IS PNT FLAG SET?
040626  001012                                  BNE     64$                     ; BRANCH IF NOT SET
040630  012746  040562                          MOV     @T44MSG,-(SP)
040634  012746  002270                          MOV     @PNTMSG,-(SP)
040640  012746  000002                          MOV     #2,-(SP)
040644  010600                                  MOV     SP,R0
040646  104417                                  TRAP    C$PNTF
040650  062706  000006                          ADD     #6,SP
040654  005037  002532                  64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
040660                                          INIT
                                        ;INITIALIZE DEVICE
040660  012777  000100  141430                  MOV     @#MQHALT,@QMT1          ;INITIALIZE Q-BRIDGE
040666  013777  002520  141422                  MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
040674  005077  141400                          CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
040700  005037  002532                          CLR     ERROR1                  ;CLEAR ERROR FLAG
040704                                          SETVEC  VECTOR,#L44,#PRI07      ; SET INTERRUPT VECTOR
040704  012746  000340                          MOV     #PRI07,-(SP)
040710  012746  041032                          MOV     #L44,-(SP)
040714  013746  002512                          MOV     VECTOR,-(SP)
040720  012746  000003                          MOV     #3,-(SP)
040724  104437                                  TRAP    C$SVEC
040726  062706  000010                          ADD     #10,SP
040732  052777  000100  141340                  BIS     #ERRIE,@QINT            ; SET ERRIE BIT IN QINT
040740  052777  000002  141334                  BIS     #TMRENB,@QCTL           ; SET TMRENB IN QCTL
                                        ;
                                        ; NOW WAIT FOR MAXIMUM OF 8 SECONDS FOR WDTO INTERRUPT
                                        ;
040746  012704  000360                          MOV     #240.,R4                ; NUMBER OF .025 SEC TO LOOP
040752                                  10$:
040752                                          DELAY   25.                     ; .025 SEC DELAY
040752  012727  000031                          MOV     #25.,(PC)+
040756  000000                                  .WORD   0
040760  013727  002116                          MOV     L$DLY,(PC)+
```

TEST44: CHECK ERR INTERRUPT (UPWRFL)

```
        040764  000000                          .WORD   0
        040766  005367  177772                  DEC     6(PC)
        040772  001375                          BNE     .-4
        040774  005367  177756                  DEC     -22(PC)
        041000  001367                          BNE     .-20
  3324  041002                                  BREAK                   ; CHECK FOR OPERATOR INPU
        041002  104422                          TRAP    C$BRK
  3325  041004  001002                          BNE     20$             ; IF TIMER COMPLETE, BRANCH
  3326  041006  005304                          DEC     R4              ; DECREMENT R4 COUNTER
  3327  041010  001360                          BNE     10$             ; BRANCH IF 6 SECONDS NOT UP
  3328                                  ;
  3329                                  ; ERROR CONDITION, 8 SECONDS UP WITH NO WDTO INTERRUPT
  3330                                  ;
  3331  041012                          20$:
  3332  041012                                  ERRHRD  1,MSG59         ; TELL OPERATOR ERROR AND ABORT
        041012  104456                          TRAP    C$ERHRD
        041014  000001                          .WORD   1
        041016  011274                          .WORD   MSG59
        041020  000000                          .WORD   0
  3333  041022  004737  013230                  JSR     PC,DUMP         ; DO OPTIONAL REGISTER DUMP
  3334  041026                                  EXIT    TST             ; ABORT TEST
        041026  104432                          TRAP    C$EXIT
        041030  000100                          .WORD   L10145-.
  3335                                  ;
  3336                                  ; END ERROR CONDITION
  3337                                  ; INTERRUPT SERVICE ROUTINE
  3338                                  ;
  3339  041032                                  BGNSRV  L44
        041032                          L44::
  3340  041032  017737  141242  002526          MOV     @QINT,VAL       ; READ QINT REGISTER
  3341  041040  012737  110000  002530          MOV     #UPWRFL!ERR,EXP ; EXPECTED PATTERN TO EXP
  3342  041046  005137  002530                  COM     EXP             ; COMPLEMENT EXP FOR CLEARING
  3343  041052  043737  002530  002526          BIC     EXP,VAL         ; CLEAR UNTESTED BITS
  3344  041060  005137  002530                  COM     EXP             ; RESTORE EXPECTED PATTERN
  3345  041064  023737  002530  002526          CMP     EXP,VAL         ; COMPARE EXPECTED WITH VALUE
  3346  041072  001410                          BEQ     30$             ; BRANCH IF OK
  3347                                  ;
  3348                                  ; ERROR CONDITION
  3349                                  ;
  3350  041074  012701  000002                  MOV     #2,R1           ; REGISTER NUMBER TO R1
  3351  041100                                  ERRHRD  2,MSG60,ERR4    ; TELL OPERATOR ERROR
        041100  104456                          TRAP    C$ERHRD
        041102  000002                          .WORD   2
        041104  011351                          .WORD   MSG60
        041106  011776                          .WORD   ERR4
  3352  041110  004737  013230                  JSR     PC,DUMP         ; DO OPTIONAL REGISTER DUMP
  3353                                  ;
  3354                                  ; END ERROR CONDITION
  3355                                  ;
  3356  041114  012716  041122          30$:    MOV     #E44,(SP)               ; ADJUST STACK
  3357  041120                                  ENDSRV
        041120                          L10146:
        041120  000002                          RTI
  3358  041122                          E44:
  3359  041122                                  CLRVEC  VECTOR          ; CLEAR VECTOR
        041122  013700  002512                  MOV     VECTOR,R0
        041126  104436                          TRAP    C$CVEC
```

```
3360 041130                              ENDTST
     041130                    L10145:
     041130  104401                      TRAP    C$ETST
```

TEST45: CHECK BR LEVEL

```
3362                                    .SBTTL  TEST45: CHECK BR LEVEL
3363 041132                             STATST  <CHECK BR LEVEL>
     041132     040     103     110     T45MSG::        .ASCIZ  / CHECK BR LEVEL/
3364
3365                                    ;*****************************************************************************
3366                                    ;**
3367                                    ;TEST DESCRIPTION-
3368                                    ; CHECKS FOR INTERRUPT WHEN PS IS BELOW DEVICE BR LEVEL
3369                                    ; CHECKS FOR NO INTERRUPT WHEN PS IS EQUAL OR ABOVE DEVICE LEVEL
3370                                    ;TEST STEPS-
3371                                    ; SET INTERRUPT VECTOR
3372                                    ; SET CPU PRIORITY = 7
3373                                    ; SET QVNT ACKIE BIT
3374                                    ; SET QCMD CMDACK BIT TO CAUSE INTERRUPT
3375                                    ; IF NOT INTERRUPT:
3376                                    ;  CHECK FOR DEVICE LEVEL < OR = CPU LEVEL
3377                                    ; IF INTERRUPT:
3378                                    ;  CHECK FOR DEVICE LEVEL > CPU LEVEL
3379                                    ; DROP CPU LEVEL AND REPEAT
3380                                    ; RESTORE INTERRUPT VECTOR
3381                                    ;
3382                                    ;--
3383                                    ;*****************************************************************************
3384
3385 041152                                   BGNTST
     041152                             T45::
3386 041152                                   CKPNT                           ; PRINT OUT TEST TITLE IF PNT SET
     041152  104421                           TRAP    C$RFLA
     041154  022700  001000                   CMP     @PNT,R0                 ; IS PNT FLAG SET?
     041160  001012                           BNE     64$                     ; BRANCH IF NOT SET
     041162  012746  041132                   MOV     @T45MSG,-(SP)
     041166  012746  002270                   MOV     @PNTMSG,-(SP)
     041172  012746  000002                   MOV     #2,-(SP)
     041176  010600                           MOV     SP,R0
     041200  104417                           TRAP    C$PNTF
     041202  062706  000006                   ADD     #6,SP
     041206  005037  002532           64$:    CLR     ERROR1                  ; CLEAR ERROR COUNTER
3387 041212                                   INIT
                                        ;INITIALIZE DEVICE
     041212  012777  000100  141076           MOV     @#QHALT,@QMT1           ;INITIALIZE Q-BRIDGE
     041220  013777  002520  141070           MOV     INITWORD,@QMT1          ;SET UP LOOPBACK OPTION
     041226  005077  141046                   CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
     041232  005037  002532                   CLR     ERROR1                  ;CLEAR ERROR FLAG
3388 041236                                   SETVEC  VECTOR,#L45,BRLEVEL     ; SET INTERRUPT VECTOR
     041236  013746  002516                   MOV     BRLEVEL,-(SP)
     041242  012746  041432                   MOV     #L45,-(SP)
     041246  013746  002512                   MOV     VECTOR,-(SP)
     041252  012746  000003                   MOV     #3,-(SP)
     041256  104437                           TRAP    C$SVEC
     041260  062706  000010                   ADD     #10,SP
3389 041264  012737  000340  002540           MOV     @#PRI07,VALUE           ; STORE CURRENT CPU LEVEL
3390 041272                             10$:
3391 041272                                   SETPRI  VALUE                   ; SET UP CURRENT CPU PRIORITY LEVEL
     041272  013700  002540                   MOV     VALUE,R0
     041276  104441                           TRAP    C$SPRI
3392 041300  005037  002542                   CLR     INTFLG                  ; CLEAR INTRRUPT FLAG
3393 041304  052777  000100  140764           BIS     #ACKIE,@QVNT            ; SET ACKIE BIT IN QVNT
```

```
TEST45: CHECK BR LEVEL

 3394 041312  052777  000001  140754          BIS     #CMDACK,@QCMD           ; CAUSE INTERRUPT
 3395 041320  000240                          NOP                             ; LITTLE BIT OF TIME
 3396 041322  005737  002542                  TST     INTFLG                  ; DID WE GET INTERRUPT
 3397 041326  001011                          BNE     20$                     ; BRANCH IF WE GOT INTERRUPT
 3398                                 ;
 3399                                 ; NO INTERRUPT, SEE CHECK CPU LEVEL VS. BR LEVEL
 3400                                 ;
 3401 041330  023737  002540  002516          CMP     VALUE,BRLEVEL           ; COMPARE CPU LEVEL VS. BR
 3402 041336  002015                          BGE     30$                     ; IF CPU >=BR, OK
 3403                                 ;
 3404                                 ; ERROR CONDITION, NO INTERRUPT
 3405                                 ;
 3406 041340                                  ERRHRD  1,MSG61,ERR6            ; TELL OPERATOR ERROR
      041340  104456                          TRAP    C$ERHRD
      041342  000001                          .WORD   1
      041344  011434                          .WORD   MSG61
      041346  012160                          .WORD   ERR6
 3407 041350  000410                          BR      30$                     ; BRANCH OVER NEXT SECTION
 3408                                 ;
 3409                                 ; INTERRUPT OCCURRED-CHECK CPU LEVEL VS. BR LEVEL
 3410                                 ;
 3411 041352  023737  002540  002516  20$:    CMP     VALUE,BRLEVEL           ; COMPARE CPU LEVEL VS. BR
 3412 041360  002404                          BLT     30$                     ; IF CPU<BR, OK; BRANCH
 3413                                 ;
 3414                                 ; ERROR CONDITION, INTERRUPT OCCURED WHEN IT SHOULDN'T
 3415                                 ;
 3416 041362                                  ERRHRD  2,MSG62,ERR6            ; TELL OPERATOR ERROR
      041362  104456                          TRAP    C$ERHRD
      041364  000002                          .WORD   2
      041366  011521                          .WORD   MSG62
      041370  012160                          .WORD   ERR6
 3417                                 ;
 3418                                 ; END ERROR CONDITION, TRY NEXT LOWER CPU LEVEL
 3419                                 ;
 3420 041372                          30$:
 3421 041372  012777  000001  140676          MOV     #VNTPRS,@QVNT           ; RESET VNTACK FOR NEXT LOOP
 3422 041400  162737  000040  002540          SUB     #PRI01,VALUE            ; TRY NEXT CPU LEVEL
 3423 041406  100401                          BMI     40$                     ; IF DONE, BRANCH
 3424 041410  000730                          BR      10$                     ; GO DO NEXT TEST
 3425                                 ;
 3426                                 ; END OF TEST, RESTORE
 3427                                 ;
 3428 041412                          40$:
 3429 041412                                  CLRVEC  VECTOR                  ; CLEAR VECTOR
      041412  013700  002512                  MOV     VECTOR,R0
      041416  104436                          TRAP    C$CVEC
 3430 041420                                  SETPRI  #PRI00                  ; LOWER CPU PRIORITY
      041420  012700  000000                  MOV     #PRI00,R0
      041424  104441                          TRAP    C$SPRI
 3431 041426                                  EXIT    TST
      041426  104432                          TRAP    C$EXIT
      041430  000010                          .WORD   L10147-.
 3432                                 ;
 3433                                 ; INTERRUPT SERVICE ROUTINE
 3434                                 ;
 3435 041432                                  BGNSRV  L45
      041432                         L45::
```

TEST45: CHECK BR LEVEL

```
3436 041432  005237  002542                    INC     INTFLG                  ; INCREMENT INTERRUPT FLG
3437 041436                                     ENDSRV
     041436                             L10150:
     041436  000002                             RTI
3438                                    ;
3439                                    ; END INTERRUPT SERVICE ROUTINE
3440                                    ;
3441 041440                                     ENDTST
     041440                             L10147:
     041440  104401                             TRAP    C$ETST
3442
3443 041442                             ENDMOD
3444
```

```
      1                                         .ENABL  AMA
     13
     14                                 .SBTTL  HARDWARE PARAMETER CODING SECTION
     42
     43 041442                                 BGNMOD
     44
     45                                 ;++
     46                                 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
     47                                 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
     48                                 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
     49                                 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
     50                                 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
     51                                 ; WITH THE OPERATOR.
     52                                 ;--
     53
     54 041442                                 BGNHRD
        041442  000020                         .WORD L10151-L$HARD/2
        041444                           L$HARD::
     55 041444                                 GPRMA   G1,0,0,0,177776,YES     ;BASE ADDRESS
        041444  000031                         .WORD   T$CODE
        041446  041504                         .WORD   G1
        041450  000000                         .WORD   T$LOLIM
        041452  177776                         .WORD   T$HILIM
     56 041454                                 GPRMA   G2,2,0,0,1000,YES       ;VECTOR
        041454  001031                         .WORD   T$CODE
        041456  041537                         .WORD   G2
        041460  000000                         .WORD   T$LOLIM
        041462  001000                         .WORD   T$HILIM
     57 041464                                 GPRML   G3,4,1,YES              ;LOOPBACK CABLE INSTALLED?
        041464  002130                         .WORD   T$CODE
        041466  041572                         .WORD   G3
        041470  000001                         .WORD   1
     58 041472                                 GPRMD   G4,6,0,340,0,7,YES      ;BR LEVEL
        041472  003032                         .WORD   T$CODE
        041474  041625                         .WORD   G4
        041476  000340                         .WORD   340
        041500  000000                         .WORD   T$LOLIM
        041502  000007                         .WORD   T$HILIM
     59
     69
     70
     71 041504                                 ENDHRD
                                               .EVEN
        041504                           L10151:
     72 041504      102     122     111  G1:     .ASCIZ  /BRIDGE MODULE BASE ADDRESS/
     73 041537      102     122     111  G2:     .ASCIZ  /BRIDGE MODULE VECTOR       /
     74 041572      114     117     117  G3:     .ASCIZ  /LOOPBACK CABLE INSTALLED? /
     75 041625      102     122     040  G4:     .ASCIZ  /BR LEVEL                   /
     76                                         .EVEN
     77
     84
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ; -

                                ;SOFTWARE PARAMETER SECTION NOT NEEDED
                                        .EVEN


                                $PATCH::
                                        .BLKW   100


                                        LASTAD
                                        .EVEN
042060  042100                          .WORD T$FREE
042062  000006                          .WORD T$SIZE
042064                          L$LAST::
                                ;HARDCODED P-TABLE
                                        BGNSETUP 1
                                        BGNPTAB
042064  000000                          .WORD   0
042066  000004                          .WORD   L10154-./2-1
042070                          L10152:
042070  160500                  .WORD   QCMDAD          ;HARDWARE UNIT 0 DEFAULT BASE ADDRESS
042072  000500                  .WORD   VEC             ;HARDWARE UNIT 0 DEFAULT VECTOR
042074  000000                  .WORD   NO              ;LOOPBACK CABLE INSTALLED DEFAULT
042076  000200                  .WORD   ^B10000000      ;DEFAULT BR INTERRUPT LEVEL 4
042100                          ENDPTAB
042100                  L10154:
042100                          ENDSETUP
000001                  .END
```

SYMBOL TABLE

```
ABORT   015030          C$CLEA- 000012  EF.RES- 000037 G  FUNC1 - 000010 G  IER   - 020000 G
ACKIE - 000100 G        C$CLOS- 000035  EF.STA- 000040 G  FUNC2 - 000020 G  INITWO  002520 G
ADD     002524 G        C$CLP1- 000006  ENDI    015032    F$    - 000001    INITO   014474
ADDRES  002274 G        C$CVEC- 000036  ERR   - 100000 G  F$AU  - 000015    INIT1   014510
ADR   - 000020 G        C$DCLN- 000044  ERRIE - 000100 G  F$AUTO- 000020    INIT2   014520
ASSEMB- 000010          C$DODU- 000051  ERROR1  002532 G  F$BGN - 000040    INIT3   014606
BAD     015112          C$DRPT- 000024  ERROR2  002534 G  F$CLEA- 000007    INTB    013610
BASEAD  002510 G        C$DU  - 000053  ERR2    011702 G  F$DU  - 000016    INTBIT  002712 G
BITS$   012506 G        C$EDIT- 000003  ERR4    011776 G  F$END - 000041    INTFLG  002542 G
BIT$    012466 G        C$ERDF- 000055  ERR5    012136 G  F$HARD- 000004    ISR   - 000100 G
BIT0  - 000001 G        C$ERHR- 000056  ERR6    012160 G  F$HW  - 000013    IXE   - 004000 G
BIT00 - 000001 G        C$ERRO- 000060  ERR7    012246 G  F$INIT- 000006    I$AU  - 000041
BIT01 - 000002 G        C$ERSF- 000054  EVL   - 000004 G  F$JMP - 000050    I$AUTO- 000041
BIT02 - 000004 G        C$ERSO- 000057  EXP     002530 G  F$MOD - 000000    I$CLN - 000041
BIT03 - 000010 G        C$ESCA- 000010  EXPECT  002354 G  F$MSG - 000011    I$DU  - 000041
BIT04 - 000020 G        C$ESEG- 000005  EXQBA   002370 G  F$PROT- 000021    I$HRD - 000041
BIT05 - 000040 G        C$ESUB- 000003  EXQCMD  002354 G  F$PWR - 000017    I$INIT- 000041
BIT06 - 000100 G        C$ETST- 000001  EXQCTL  002362 G  F$RPT - 000012    I$MOD - 000040
BIT07 - 000200 G        C$EXIT- 000032  EXQDMA  002364 G  F$SEG - 000003    I$MSG - 000041
BIT08 - 000400 G        C$GETB- 000026  EXQINT  002360 G  F$SOFT- 000005    I$PROT- 000040
BIT09 - 001000 G        C$GETW- 000027  EXQMTO  002374 G  F$SRV - 000010    I$PTAB- 000041
BIT1  - 000002 G        C$GMAN- 000043  EXQMT1  002376 G  F$SUB - 000002    I$PWR - 000041
BIT10 - 002000 G        C$GPWR- 000042  EXQUBA  002366 G  F$SW  - 000014    I$RPT - 000041
BIT11 - 004000 G        C$GPLO- 000030  EXQVAD  002400 G  F$TEST- 000001    I$SEG - 000041
BIT12 - 010000 G        C$GPRI- 000040  EXQVDA  002402 G  GO    - 000001 G  I$SETU- 000041
BIT13 - 020000 G        C$INIT- 000011  EXQVNT  002356 G  G$CNTO- 000200    I$SRV - 000041
BIT14 - 040000 G        C$INLP- 000020  EXQWC   002372 G  G$DELM- 000372    I$SUB - 000041
BIT15 - 100000 G        C$MANI- 000050  E$END - 002100    G$DISP- 000003    I$TST - 000041
BIT2  - 000004 G        C$MEM - 000031  E$LOAD- 000035    G$EXCP- 000400    J$JMP - 000167
BIT3  - 000010 G        C$MSG - 000023  E10$    013306    G$HILI- 000002    KISAR0- 172340 G
BIT4  - 000020 G        C$OPEN- 000034  E39     037066    G$LOLI- 000001    KISDR0- 172300 G
BIT5  - 000040 G        C$PNTB- 000014  E40     037346    G$NO  - 000000    LOE   - 040000 G
BIT6  - 000100 G        C$PNTF- 000017  E41     037640    G$OFFS- 000400    LOGUNI  002504 G
BIT7  - 000200 G        C$PNTS- 000016  E42     040206    G$OFSI- 000376    LOOPBK  002514 G
BIT8  - 000400 G        C$PNTX- 000015  E43     040544    G$PRMA- 000001    LOT   - 000010 G
BIT9  - 001000 G        C$QIO - 000377  E44     041122    G$PRMD- 000002    L$ACP   002110 G
BOE   - 000400 G        C$RDBU- 000007  E5$     013334    G$PRML- 000000    L$APT   002036 G
BRDGFL  002502 G        C$REFG- 000047  E6$     013372    G$RADA- 000140    L$AU    015132 G
BRESET  016002          C$RESE- 000033  FLAG    014170    G$RADB- 000000    L$AUT   002070 G
BRIDGE  002500 G        C$REVI- 000003  FOR0    003333 G  G$RADD- 000040    L$AUTO  015040 G
BRLEVE  002516 G        C$RFLA- 000021  FOR1    003337 G  G$RADL- 000120    L$CCP   002106 G
BUILD$  012530          C$RPT - 000025  FOR10   004261 G  G$RADO- 000020    L$CLEA  015044 G
CMDACK- 000001 G        C$SEFG- 000046  FOR11   004353 G  G$XFER- 000004    L$CO    002032 G
CMDB    013560          C$SPRI- 000041  FOR12   004462 G  G$YES - 000010    L$DEPO  002011 G
CMDBIT  002624 G        C$SVEC- 000037  FOR13   004466 G  G1      041504    L$DESC  002560 G
CMDPRS- 004000 G        C$TPRI- 000013  FOR2    003454 G  G2      041537    L$DESP  002076 G
CONFIG  003244 G        DETAIL  012366  FOR3    003504 G  G3      041572    L$DEVP  002060 G
CONT    015344          DFPTBL  002260 G  FOR4    003552 G  G4      041625    L$DISP  002124 G
CTLB    013624          DIAGMC- 000000  FOR4.5  003600 G  HELP  - 000000    L$DLY   002116 G
CTLBIT  002771 G        DMAB    013640  FOR5    003670 G  HELP1   011576 G  L$DTP   002040 G
C$AU  - 000052          DMABIT  003042 G  FOR5.5  003762 G  HELP2   011645 G  L$DTYP  002034 G
C$AUTO- 000061          DMAWT - 000002 G  FOR6    004054 G  HIUBM - 170370 G  L$DU    015122 G
C$BRK - 000022          DUMP    013230 G  FOR7    004114 G  HOE   - 100000 G  L$DUT   002072 G
C$BSEG- 000004          EDUMP   013556  FOR8    004154 G  HP1     002544 G  L$DVTY  002554 G
C$BSUB- 000002          EF.CON- 000036 G  FOR9    004227 G  HP2     002546 G  L$EF    002052 G
C$CEFG- 000045          EF.NEW- 000035 G  FREMEM  002552 G  IBE   - 010000 G  L$ENVI  002044 G
C$CLCK- 000062          EF.PWR- 000034 G  FUNC0 - 000004 G  IDU   - 000040 G  L$ETP   002102 G
```

SYMBOL TABLE

L$EXP1	002046 G	L10035	020076	L10126	035656	MSG19	006006 G	NAMES	002404 G		
L$EXP4	002064 G	L10036	020046	L10127	035632	MSG2	004571 G	NEXT	014614		
L$EXP5	002066 G	L10037	020414	L10130	036374	MSG20	006033 G	NO	= 000000 G		
L$HARD	041444 G	L10040	020364	L10131	036614	MSG21	006060 G	NO$CHA	013224		
L$HIME	002120 G	L10041	020716	L10132	036606	MSG22	006125 G	ONEFIL	= 000000		
L$HPCP	002016 G	L10042	020666	L10133	037074	MSG23	006172 G	O$APTS	= 000000		
L$HPTP	002022 G	L10043	021242	L10134	037064	MSG24	006241 G	O$AU	= 000000		
L$HW	002260 G	L10044	021212	L10135	037354	MSG25	006267 G	O$BGNR	= 000001		
L$ICP	002104 G	L10045	021536	L10136	037344	MSG26	006315 G	O$BGNS	= 000000		
L$INIT	014466 G	L10046	021506	L10137	037646	MSG27	006342 G	O$DU	= 000000		
L$LADP	002026 G	L10047	022054	L10140	037636	MSG28	006367 G	O$ERRT	= 000000		
L$LAST	042064 G	L10050	022024	L10141	040222	MSG29	006430 G	O$GNSW	= 000000		
L$LOAD	002100 G	L10051	022452	L10142	040204	MSG3	004632 G	O$POIN	= 000001		
L$LUN	002074 G	L10052	022726	L10143	040560	MSG30	006471 G	O$SETU	= 000001		
L$MREV	002050 G	L10053	022676	L10144	040542	MSG31	006532 G	PLOC	002506 G		
L$NAME	002000 G	L10054	023216	L10145	041130	MSG32	006573 G	PNT	= 001000 G		
L$PRIO	002042 G	L10055	023166	L10146	041120	MSG33	006633 G	PNTMSG	002270 G		
L$PROT	014460 G	L10056	023546	L10147	041440	MSG34	006673 G	PRI	= 002000 G		
L$PRT	002112 G	L10057	023516	L10150	041436	MSG35	006736 G	PRINTI	013666		
L$REPP	002062 G	L10060	024144	L10151	041504	MSG[illegible]	006777 G	PRI00	= 000000 G		
L$REV	002010 G	L10061	024114	L10152	042070	MSG36.	007023 G	PRI01	= 000040 G		
L$RPT	014446 G	L10062	024544	L10154	042100	MSG37	007103 G	PRI02	= 000100 G		
L$SPC	002056 G	L10063	024336	L28	030426	MSG38	007172 G	PRI03	= 000140 G		
L$SPCP	002020 G	L10064	024412	L34	034462	MSG39	007254 G	PRI04	= 000200 G		
L$SPTP	002024 G	L10065	024466	L35	034774	MSG4	004672 G	PRI05	= 000240 G		
L$STA	002030 G	L10066	024542	L36	035472	MSG40	007337 G	PRI06	= 000300 G		
L$TEST	002114 G	L10067	025030	L37	036034	MSG41	007423 G	PRI07	= 000340 G		
L$TIML	002014 G	L10070	025000	L38	036512	MSG42	007520 G	PRSIE	= 000100 G		
L$UNIT	002012 G	L10071	025330	L39	037060 G	MSG42.	007556 G	P1	013720		
L10000	002270	L10072	025300	L40	037340 G	MSG43	007626 G	P2	013772		
L10001	011774	L10073	025604	L41	037632 G	MSG44	007663 G	P37	036372		
L10002	012134	L10074	025554	L42	040116 G	MSG45	007775 G	QBA	002310 G		
L10003	012156	L10075	026074	L43	040454 G	MSG46	010077 G	QBA00	= 000001 G		
L10004	012244	L10076	026044	L44	041032 G	MSG47	010140 G	QBA01	= 000002 G		
L10005	012364	L10077	026376	L45	041432 G	MSG48	010177 G	QBA02	= 000004 G		
L10006	014444	L10100	026346	MAXUNI	= 000010 G	MSG49	010247 G	QBA03	= 000010 G		
L10007	014456	L10101	026714	MEMSIZ	002550 G	MSG5	004731 G	QBA04	= 000020 G		
L10011	015036	L10102	026664	MES1$	012766 G	MSG50	010337 G	QBA05	= 000040 G		
L10012	015042	L10103	027254	MES2$	013015 G	MSG51	010407 G	QBA06	= 000100 G		
L10013	015120	L10104	027224	MES2$L	= 000204	MSG52	010477 G	QBA07	= 000200 G		
L10014	015130	L10105	027604	MIODIS	= 000200 G	MSG53	010547 G	QBA08	= 000400 G		
L10015	015140	L10106	027554	MMINIT	014370 G	MSG54	010653 G	QBA09	= 001000 G		
L10016	016020	L10107	030016	MMR0	= 177572 G	MSG55	010723 G	QBA10	= 002000 G		
L10017	015424	L10110	030014	MMR3	= 172516 G	MSG56	011012 G	QBA11	= 004000 G		
L10020	015506	L10111	030300	MQHALT	= 000100 G	MSG57	011101 G	QBA12	= 010000 G		
L10021	015562	L10112	030276	MQPFL	= 000040 G	MSG58	011205 G	QBA13	= 020000 G		
L10022	015636	L10113	031520	MRDY	= 001000 G	MSG59	011274 G	QBA14	= 040000 G		
L10023	015712	L10114	031516	MSG1	004544 G	MSG6	004771 G	QBA15	= 100000 G		
L10024	015774	L10115	032366	MSG10	005161 G	MSG60	011351 G	Q$US	003316 G		
L10025	016256	L10116	033256	MSG11	005224 G	MSG61	011434 G	QCMD	002274 G		
L10026	016166	L10117	033562	MSG12	005266 G	MSG62	011521 G	QCMDAD	= 160500 G		
L10027	016254	L10120	034024	MSG13	005330 G	MSG7	005012 G	QCTL	002302 G		
L10030	016750	L10121	034334	MSG14	005373 G	MSG8	005054 G	QDMA	002304 G		
L10031	017256	L10122	034646	MSG15	005436 G	MSG9	005116 G	QINT	002300 G		
L10032	017226	L10123	034644	MSG16	005531 G	MT1B	013654	QMT0	002314 G		
L10033	017602	L10124	035166	MSG17	005622 G	MT1BIT	003161 G	QMT1	002316 G		
L10034	017552	L10125	035164	MSG18	005715 G	MVADR	= 000002 G	QNEX	= 020000 G		

SEQ 0173

SYMBOL TABLE

```
QUBA     002306 G    QWC08 = 000400 G    TRPFLG  002522 G    T1.5    015640      T26MSG  027606 G
QUBA00= 000001 G    QWC09 = 001000 G    T$ARGC= 000002      T1.6    015714      T26.1   027724
QUBA01= 000002 G    QWC10 = 002000 G    T$CODE= 003032      T10     021304 G    T27     030064 G
QUBA02= 000004 G    QWC11 = 004000 G    T$ERRN= 000002      T10MSG  021244 G    T27MSG  030020 G
QUBA03= 000010 G    QWC12 = 010000 G    T$EXCP= 000000      T10.1   021406      T27.1   030154
QUBA04= 000020 G    QWC13 = 020000 G    T$FLAG= 000040      T11     021600 G    T28     030334 G
QUBA05= 000040 G    QWC14 = 040000 G    T$FREE= 042100      T11MSG  021540 G    T28MSG  030302 G
QUBA06= 000100 G    QWC15 = 100000 G    T$GMAN= 000000      T11.1   021710      T28.1   030424
QUBA07- 000200 G    QXAD16= 000400 G    T$HILI= 000007      T12     022104 G    T29     031570 G
QUBA08= 000400 G    QXAD17= 001000 G    T$LAST= 000001      T12MSG  022056 G    T29MSG  031522 G
QUBA09= 001000 G    QXAD18= 002000 G    T$LOLI= 000000      T13     022502 G    T3      016302 G
QUBA10= 002000 G    QXAD19= 004000 G    T$LSYM= 010000      T13MSG  022454 G    T3MSG   016260 G
QUBA11= 004000 G    QXAD20= 010000 G    T$LTNO= 000055      T13.1   022604      T30     032452 G
QUBA12= 010000 G    QXAD21= 020000 G    T$NEST= 000000      T14     022756 G    T30MSG  032370 G
QUBA13= 020000 G    RAM     002536 G    T$NS0 = 000000      T14MSG  022730 G    T31     033306 G
QUBA14= 040000 G    RDY   = 000200 G    T$NS1 = 000004      T14.1   023066      T31MSG  033260 G
QUBA15= 100000 G    RDYIE = 000100 G    T$NS2 = 000010      T15     023266 G    T32     033620 G
QVAD     002320 G    RP0     014044 G    T$NS3 = 000003      T15MSG  023220 G    T32MSG  033564 G
QVAD00= 000001 G    RP1     014050 G    T$NUM = 000055      T15.1   023402      T33     034062 G
QVAD01= 000002 G    RP2     014140 G    T$PCNT= 000000      T16     023616 G    T33MSG  034026 G
QVAD02= 000004 G    RP4     014165 G    T$PTAB= 010153      T16MSG  023550 G    T34     034362 G
QVAD03= 000010 G    SELDMA= 000024 G    T$PTHV= 000001      T16.1   023740      T34MSG  034336 G
QVAD04= 000020 G    SELQCM= 000010 G    T$PTNU= 000001      T17     024174 G    T34.1   034460
QVAD05= 000040 G    SELQEV= 000004 G    T$SAVL= 177777      T17MSG  024146 G    T35     034674 G
QVAD06= 000100 G    SELRDD= 000034 G    T$SEGL= 177777      T17.1   024264      T35MSG  034650 G
QVAD07= 000200 G    SELUBA= 000020 G    T$SEKO= 010000      T17.2   024340      T35.1   034772
QVAD08= 000400 G    SELUVA= 000014 G    T$SIZE= 000006      T17.3   024414      T36     035220 G
QVAD09= 001000 G    SELUVD= 000030 G    T$SUBN= 000000      T17.4   024470      T36MSG  035170 G
QVAD10= 002000 G    SETUP   014606      T$TAGL= 177777      T18     024576 G    T36.1   035326
QVDA     002322 G    SQBA    002340 G    T$TAGN= 010155      T18MSG  024546 G    T37     035702 G
QVDA00= 000001 G    SQCMD   002324 G    T$TEMP= 000004      T18.1   024706      T37MSG  035660 G
QVDA01= 000002 G    SQCTL   002332 G    T$TEST= 000055      T19     025062 G    T38     036420 G
QVDA02= 000004 G    SQDMA   002334 G    T$TSTM= 177777      T19MSG  025032 G    T38MSG  036376 G
QVDA03= 000010 G    SQINT   002330 G    T$TSTS= 000001      T19.1   025200      T38.1   036510
QVDA04= 000020 G    SQMTO   002344 G    T$$AU = 010015      T2      016050 G    T39     036646 G
QVDA05= 000040 G    SQMT1   002346 G    T$$AUT= 010012      T2MSG   016022 G    T39MSG  036616 G
QVDA06= 000100 G    SQUBA   002336 G    T$$CLE= 010013      T2.1    016110      T4      017012 G
QVDA07= 000200 G    SQVAD   002350 G    T$$DAT= 010154      T2.2    016170      T4MSG   016752 G
QVDA08= 000400 G    SQVDA   002352 G    T$$DU = 010014      T20     025360 G    T4.1    017126
QVDA09= 001000 G    SQVNT   002326 G    T$$HAR= 010151      T20MSG  025332 G    T40     037126 G
QVDA10= 002000 G    SQWC    002342 G    T$$HW = 010000      T20.1   025462      T40MSG  037076 G
QVDA11= 004000 G    STATUS  013226      T$$INI= 010011      T21     025634 G    T41     037404 G
QVDA12= 010000 G    STORE$  013222      T$$MSG= 010005      T21MSG  025606 G    T41MSG  037356 G
QVDA13= 020000 G    SVCGBL= 000000      T$$PC = 000001      T21.1   025744      T42     037704 G
QVDA14= 040000 G    SVCINS= 000000      T$$PRO= 010010      T22     026142 G    T42MSG  037650 G
QVDA15= 100000 G    SVCSUB= 000000      T$$PTA= 010153      T22MSG  026076 G    T43     040260 G
QVNT     002276 G    SVCTAG= 00C000      T$$RPT= 010007      T22.1   026232      T43MSG  040224 G
QVREN = 000020 G    SVCTST= 000000      T$$SEG= 010000      T23     026444 G    T44     040620 G
QWC      002312 G    S$LSYM= 010000      T$$SRV= 010150      T23MSG  026400 G    T44MSG  040562 G
QWC00 = 000001 G    TABLE3  036362      T$$SUB= 010132      T23.1   026542      T45     041152 G
QWC01 = 000002 G    TE$   = 000002      T$$TES= 010147      T24     027006 G    T45MSG  041132 G
QWC02 = 000004 G    TIME.5= 000000 G    T1      015152 G    T24MSG  026742 G    T5      017320 G
QWC03 = 000010 G    TIME1 = 000004 G    T1MSG   015142 G    T24.1   027116      T5MSG   017260 G
QWC04 = 000020 G    TIME2 = 000010 G    T1.1    015344      T25     027322 G    T5.1    017436
QWC05 = 000040 G    TIME4 = 000014 G    T1.2    015426      T25MSG  027256 G    T6      017644 G
QWC06 = 000100 G    TMRENB= 000002 G    T1.3    015510      T25.1   027440      T6MSG   017604 G
QWC07 = 000200 G    TRAP4   014440 G    T1.4    015564      T26     027652 G    T6.1    017746
```

```
CZFPA-AO CTRL BRIDGE INTFC DIAG MACRO M1200  09 JAN 84 12:33  PAGE 91 4

                                                                                SEQ 0174
SYMBOL TABLE

T7       020140 G        T9.1     021076          VALUE    002540 G        WDTRST= 000001 G        X$TRUE= 000020
T7MSG    020100 G        UAM    = 000200 G        VALUES   002324 G        WDTO   = 000004 G        YES   = 000001 G
T7.1     020250          UBUS     003324 G        VEC    = 000500 G        WDT1   = 000010 G        $HLP     014172 G
T8       020456 G        UNEX   = 040000 G        VECTOR   002512 G        WSPACE= 020040 G        $HLPFO   014362 G
T8MSG    020416 G        UPWRFL= 010000 G        VNTACK= 002000 G        X$       013510          $MESLN   014364
T8.1     020566          UXAD16= 000020 G        VNTB     013574          X$ALWA= 000000          $PATCH   041660 G
T9       020760 G        UXAD17= 000040 G        VNTBIT   002657 G        X$FALS= 000040          $STORE   014366
T9MSG    020720 G        VAL      002526 G        VNTPRS= 000001 G        X$OFFS= 000400

. ABS.   042100      000
         000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  29816 WORDS  ( 117 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:12:57
ZFPAAO.BIC,CZFPAAO/-SP=SVC/ML,ZFPAA01,ZFPAA02,ZFPAA03,ZFPAA04,ZFPAA05,ZFPAA06
.
```