

FEPCH

CTAL BRIDGE INTFC
CZFPAAO

COPYRIGHT (c) 1982-84

AH-T859A-MC

FICHE 1 OF 1

APR 1984

digital

Made In USA

.REM 6

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

COPYRIGHT (C) 1982,1984
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED
AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE
AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS
SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR
OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO
AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY
OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC
LSI

PDP
DECUS
VAX

UNIBUS
DECTAPE

MASSBUS
Q-BUS

REVISION HISTORY

- A.0 DALE PROCTOR 22-DEC-1983
CHANGED THE NAME TO CZFPA-A0 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,BQBA TO
MOV #PATCH,BQBA 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 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

FEPCH TECHNICAL MANUAL
AC-T860A-MC CZFPBAO CTRL BRIDGE LINK
AC-T866AS-MC CXFPCAO 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 HIERARCHY 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 ADDRESS OF Q-COMMAND REGISTER	DEFAULT=160500
LSI BRIDGE MODULE VECTOR ADDRESS OF INTERRUPT VECTOR	DEFAULT=500
LOOPBACK CABLE INSTALLED INTERNAL LOOPBACK OR PHYSICAL CABLE USED	DEFAULT=NO
BR LEVEL INTERRUPT PRIORITY LEVEL	DEFAULT=4

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 EXPLANATORY. 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 CHDPRS INTERRUPT

TEST41:
CHECK RDY INTERRUPT

TEST42:
CHECK ERR INTERRUPT (UNEX)

TEST43:
CHECK ERR INTERRUPT (QEX)

TEST44:
CHECK ERR INTERRUPT (UPWRFL)

TEST45:
CHECK BR LEVEL

```

12      .TITLE CZFPA-AO CTRL BRIDGE INTFC DIAGNOSTIC
13      .SBITL  PROGRAM HEADER
39
41      000000      .ENABL  ABS,AMA
42      002000      "      2000
47
48      002000      BGNMOD
49
50      ;**
51      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
52      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
53      ;--
54
55      002000      POINTER BGNSETUP,BGNRPT
56
73
74      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$DEPO::      ;0
      002011      060      .ASCII  /O/
      002012      L$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::      .WORD   0
      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	L\$EXP1::	.WORD	0	;EXPANSION WORD
002046		L\$MREV::	.WORD	0	;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 ERR_TBL
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	

DISPATCH TABLE

```

87
88
89
90
91
92
93
94 002122
    002122 000055
    002124
    002124 015152
    002126 016050
    002130 016302
    002132 017012
    002134 017320
    002136 017644
    002140 020140
    002142 020456
    002144 020760
    002146 021304
    002150 021600
    002152 022104
    002154 022502
    002156 022756
    002160 023266
    002162 023616
    002164 024174
    002166 024576
    002170 025062
    002172 025360
    002174 025634
    002176 026142
    002200 026444
    002202 027006
    002204 027322
    002206 027652
    002210 030064
    002212 030334
    002214 031570
    002216 032452
    002220 033306
    002222 033620
    002224 034062
    002226 034362
    002230 034674
    002232 035226
    002234 035702
    002236 036420
    002240 036646
    002242 037126
    002244 037404
    002246 037704
    002250 040260
    002252 040620
    002254 041152

```

95

.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
.LDISPATCH::
    .WORD 45
    .WORD T1
    .WORD T2
    .WORD T3
    .WORD T4
    .WORD T5
    .WORD T6
    .WORD T7
    .WORD T8
    .WORD T9
    .WORD T10
    .WORD T11
    .WORD T12
    .WORD T13
    .WORD T14
    .WORD T15
    .WORD T16
    .WORD T17
    .WORD T18
    .WORD T19
    .WORD T20
    .WORD T21
    .WORD T22
    .WORD T23
    .WORD T24
    .WORD T25
    .WORD T26
    .WORD T27
    .WORD T28
    .WORD T29
    .WORD T30
    .WORD T31
    .WORD T32
    .WORD T33
    .WORD T34
    .WORD T35
    .WORD T36
    .WORD T37
    .WORD T38
    .WORD T39
    .WORD T40
    .WORD T41
    .WORD T42
    .WORD T43
    .WORD T44
    .WORD T45

```


DEFAULT HARDWARE P-TABLE

```
103      .SBTTL  DEFAULT HARDWARE P-TABLE
104
105      ;**
106      ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
107      ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
108      ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES.
109      ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
110      ;--
111
112 002256      BGNHW  DFPTBL
      002256      .WORD  L10000-L$HW/2
      002260      L$HW::
      002260      DFPTBL::
113
114 002260      160500      .WORD  QCMDAD      ;HARDWARE UNIT 0 DEFAULT BASE ADDRESS
115 002262      000500      .WORD  VEC        ;HARDWARE UNIT 0 DEFAULT VECTOR
116 002264      000000      .WORD  NO         ;LOOPBACK CABLE INSTALLED DEFAULT
117 002266      000200      .WORD  *B000000000100000000 ;DEFAULT BR INTERRUPT LEVEL 4
127
128 002270      ENDHW
      002270      L10000:
```

SOFTWARE P-TABLE

131
132
133
134
135
136
137
138
139
140
148
149
150 002270

.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

ENDMOD

E2

SEQ 0017

SOFTWARE P-TABLE

11
12
40
50
51 002270

.ENABL AMA
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

MACRO DEFINITIONS

```

53 .SBTTL MACRO DEFINITIONS
54 ;*****
55 ;**
56 ; MACRO DEFINITION SECTION
57 ;--
58 ;*****
59
60 .MACRO INIT
61 .LIST
62 ;INITIALIZE DEVICE
63     MOV     @M0HALT,@QMT1           ;INITIALIZE Q-BRIDGE
64     MOV     INITWORD,@QMT1         ;SET UP LOOPBACK OPTION
65     CLR     @QINT                   ;CLEAR QINT AFTER LOOPBACK ENABLED
66     CLR     ERROR1                 ;CLEAR ERROR FLAG
67 .NLIST
68 .ENDM
69
70 ; MACROS TO CALL BITS SUBROUTINE
71
72 ; MACRO SETBITS
73 ; USED TO CALL ROUTINE TO PRINT OUT REGISTER BIT NAMES
74
75 .MACRO SETBITS NAME,POINTER,VALUE
76 .NARG     TE#                       ;GET NUMBER OF ARGUMENTS PASSED
77 .IF NE    TE#-3                     ;SEE IF RIGHT NUMBER
78 .ERROR    ;SETBITS MACRO REQUIRES 3 PARAMETERS
79 .ENDC
80 .NCHR     $L1 NAME                  ;GET NUMBER OF CHARACTERS IN NAME PARAMETER
81 $L1 = $L1+1                        ;ALLOW FOR ZERO BYTE IN ASCIZ
82 $TEMP = $L1 & 1                    ;CHECK FOR ODD OR EVEN
83 .IF EQ $TEMP                          ;IF EVEN LENGTH OF NAME
84 $LEN = 4+$L1
85 .ENDC
86 .IF NE $TEMP                          ;IF ODD LENGTH OF NME
87 $LEN = 5+$L1
88 .ENDC
89     MOV     VALUE,R1                ;GET VALUE REGISTER TO READ
90     JSR     RS,BIT#                 ;SUBROUTINE TO OUTPUT VALUES
91     .WORD   $LEN
92     .WORD   POINTER
93     .ASCIZ  /NAME/
94     .EVEN
95 .ENDM
96
97 ; MACRO S#BITS
98 ; USED TO CALL ROUTINE TO FORMAT PRINT BUFFER OF REGISTER BIT NAMES
99
100 .MACRO S#BITS POINTER,VALUE
101 .NARG     TE#                       ;GET NUMBER OF ARGUMENTS PASSED
102 .IF NE    TE#-2                     ;SEE IF RIGHT NUMBER
103 .ERROR    ;S#BITS MACRO REQUIRES 2 PARAMETERS
104 .ENDC
105     MOV     VALUE,R1                ;GET VALUE REGISTER TO READ
106     JSR     RS,BIT#                 ;SUBROUTINE TO OUTPUT VALUES
107     .WORD   POINTER
108     .EVEN
109 .ENDM

```


MACRO DEFINITIONS

```

111      ;
112      ; MACROS USED TO PRINT OUT THE TEST'S TITLE IF THE PNT FLAG IS SET
113      ;
114      .MACRO STATST P1
115      .MCALL PASTST
116      .IF EQ F$
117      .ERROR ;2 STATST MACRO'S WITHOUT CKPNT MACRO CALL IN BETWEEN
118      .MEXIT
119      .ENDC
120      F$ = 0
121      .RADIX 10
122      T$NUM = T$NUM+1
123      PASTST \T$NUM,<P1>
124      .RADIX 8
125      .ENDM
126      .MACRO PASTST Z,P2
127      .LIST
128      T'Z'MSG: .ASCIZ / P2/
129      .NLIST
130      .LIST MEB
131      .EVEN
132      .ENDM
133      .MACRO CKPNT
134      .MCALL PTPNT
135      .IF NE F$
136      .ERROR ;CKPNT MACRO MUST FOLLOW STATST MACRO, NOT ANOTHER CKPNT CALL
137      .MEXIT
138      .ENDC
139      .RADIX 10
140      F$ = 1
141      PTPNT \T$NUM
142      .RADIX 8
143      .ENDM
144      .MACRO PTPNT N,?A
145      RFLAGS RO
146      CMP #PNT,RO
147      BNE A
148      PRINTF #PNTMSG,#T'N'MSG
149      CLR ERROR1
150      .ENDM
151      .MCALL PTPNT
152      .ENDM
153      .MCALL PTPNT

```

002270 045 124 000

000000

000001

PNTMSG: .ASCIZ /#T/

T\$NUM = 0

F\$ = 1

: READ CURRENT FLAGS
: IS PNT FLAG SET?
: BRANCH IF NOT SET
: PRINT MESSAGE IF PNT SET
: CLEAR ERROR COUNTER

MACRO DEFINITIONS

```

155
156
157
158
159
160
161
162
163
164
165
166
167
168
169 002273
170
171
172
173 002273
174
175 002273

;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

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

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
;
; STANDARD REGISTER BASE ADDRESS CONFIGURATION
;
160500 QCMDAD == 160500 ; STANDARD BASE ADDRESS FOR UNIT 0
000500 VEC == 500 ; STANDARD VECTOR ADDRESS FOR UNIT 0
;
; BIT DEFINITIONS
;
; QCMD - Q-BUS COMMAND REGISTER
;
000001 CMDACK == BIT00 ; NO COMMAND ACKNOWLEDGE
000100 PRSIE == BIT06 ; R/W INTERRUPT ENABLE
;
; QVNT - Q-BUS EVENT REGISTER
;
000001 VNTPRS == BIT00 ; NO EVENT PRESENT
000100 ACKIE == BIT06 ; R/W INTERRUPT ENABLE
;
; QINT - Q-BUS INTERRUPT SUMMARY REGISTER
;
000100 ERRIE == BIT06 ; R/W INTERRUPT ENABLE
001000 MRDY == BIT09 ; RO MIRROR OF QDMA-07, READY

```

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      ;
273      ;      QWC - Q-BUS DMA WORD COUNT REGISTER
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      ;
292      ;      QMT1 - Q-BUS MAINTENANCE REGISTER 1
293      000002      MVADR      -- BIT01      ; R/W MAINTENANCE VIEW ADDRESS
294      000004      FUNC0      -- BIT02      ; R/W SELECT MAINTENANCE MODE BIT 02 FOR QMT0
295      000010      FUNC1      -- BIT03      ; R/W SELECT MAINTENANCE MODE BIT 03 FOR QMT0
296      000020      FUNC2      -- BIT04      ; R/W SELECT MAINTENANCE MODE BIT 04 FOR QMT0
297      000040      MQPFL      -- BIT05      ; R/W MAINTENANCE Q-BUS POWER FAIL
298      000100      MQHALT      -- BIT06      ; NO MAINTENANCE Q-BUS HALT
299      000200      MIODIS      -- BIT07      ; R/W MAINTENANCE I/O DISABLE
300      ;
301      ;
302      ;      QVAD - Q-BUS VIEW ADDRESS
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

```

313      002000      QVAD10  == BIT10      ; R/W Q-BUS VIEW ADDRESS REG. BIT 10
314      ;
315      ;          QVDA - Q-BUS VIEW DATA REGISTER
316      ;
317      000001      QVDA00  == BIT00      ; R/W Q-BUS VIEW DATA REG. BIT00
318      000002      QVDA01  == BIT01      ; R/W Q-BUS VIEW DATA REG. BIT01
319      000004      QVDA02  == BIT02      ; R/W Q-BUS VIEW DATA REG. BIT02
320      000010      QVDA03  == BIT03      ; R/W Q-BUS VIEW DATA REG. BIT03
321      000020      QVDA04  == BIT04      ; R/W Q-BUS VIEW DATA REG. BIT04
322      000040      QVDA05  == BIT05      ; R/W Q-BUS VIEW DATA REG. BIT05
323      000100      QVDA06  == BIT06      ; R/W Q-BUS VIEW DATA REG. BIT06
324      000200      QVDA07  == BIT07      ; R/W Q-BUS VIEW DATA REG. BIT07
325      000400      QVDA08  == BIT08      ; R/W Q-BUS VIEW DATA REG. BIT08
326      001000      QVDA09  == BIT09      ; R/W Q-BUS VIEW DATA REG. BIT09
327      002000      QVDA10  == BIT10      ; R/W Q-BUS VIEW DATA REG. BIT10
328      004000      QVDA11  == BIT11      ; R/W Q-BUS VIEW DATA REG. BIT11
329      010000      QVDA12  == BIT12      ; R/W Q-BUS VIEW DATA REG. BIT12
330      020000      QVDA13  == BIT13      ; R/W Q-BUS VIEW DATA REG. BIT13
331      040000      QVDA14  == BIT14      ; R/W Q-BUS VIEW DATA REG. BIT14
332      100000      QVDA15  == BIT15      ; R/W Q-BUS VIEW DATA REG. BIT15
333
334      ;
335      ;          MAINTENANCE FUNCTION, BIT IN QMT1 TO DEFINE USE OF QMT0
336      ;
337      000004      SELQEVENT == 000004      ; SELECT QEVENT
338      000010      SELQCMD   == 000010      ; SELECT QCMD
339      000014      SELUVADR   == 000014      ; SELECT UVADR
340      000020      SELUBAR    == 000020      ; SELECT UBAR
341      000024      SELDMADAT  == 000024      ; SELECT DMADAT
342      000030      SELUVDAT   == 000030      ; SELECT UVDAT
343      000034      SELRDDMA   == 000034      ; SELECT DMA READ
344
345      ;
346      ;          MEMORY MANAGEMENT MAPPING ADDRESSES
347      ;
348      177572      MMRO       == 177572
349      172516      MMR3       == 172516
350      172300      KISDRO     == 172300
351      172340      KISARO     == 172340
352      170370      HIUBM      == 170370
353
354      ;
355      ;          MISC. EQUATES
356      ;
357      000010      MAXUNITS    == 8.          ; MAXIMUM NUMBER OF UNITS ALLOWED
358      000000      NO         == 0           ; LOGICAL VALUE FOR NO
359      000001      YES        == 1           ; LOGICAL VALUE FOR YES
360      020040      WSPACE     == 20040       ; WORD OF ASCII SPACES
361
362
377

```

GLOBAL DATA SECTION

```

379      .SBTTL  GLOBAL DATA SECTION
380
381 002273
382
383      STARS
384      ;*****
385      ;*      GLOBAL DATA SECTION
386      ;*
387      STARS
388      ;*****
389
390      ;
391      ; CURRENT UNIT Q-BUS BRIDGE INTERFACE REGISTERS ADDRESSES ARE STORED HERE
392      ;
393      .EVEN
394      ADDRESSES::
395      QCMD::      .WORD      0      ; Q-BUS COMMAND REGISTER
396      QVNT::      .WORD      0      ; Q-BUS EVENT REGISTER
397      QINT::      .WORD      0      ; Q-BUS INTERRUPT SUMMARY REGISTER
398      QCTL::      .WORD      0      ; Q-BUS CONTROL REGISTER
399      QDMA::      .WORD      0      ; Q-BUS DMA CONTROL & STATUS REGISTER
400      QUBA::      .WORD      0      ; Q-BUS BUS ADDRESS UNIBUS REGISTER
401      QBA::       .WORD      0      ; Q-BUS BUS ADDRESS Q-BUS REGISTER
402      QWC::       .WORD      0      ; Q-BUS WORD COUNT REGISTER
403      QMT0::      .WORD      0      ; Q-BUS MAINTENANCE REGISTER 0
404      QMT1::      .WORD      0      ; Q-BUS MAINTENANCE REGISTER 1
405      QVAD::      .WORD      0      ; Q-BUS VIEW ADDRESS REGISTER
406      QVDA::      .WORD      0      ; Q-BUS DATA REGISTER
407
408      ;
409      ; CURRENT UNIT Q-BUS BRIDGE INTERFACE REGISTERS VALUES ARE STORED HERE
410      ;
411      VALUES::
412      SQCMD::      .WORD      0      ; COMMAND REGISTER
413      SQVNT::      .WORD      0      ; EVENT REGISTER
414      SQINT::      .WORD      0      ; INTERRUPT SUMMARY REGISTER
415      SQCTL::      .WORD      0      ; CONTROL REGISTER
416      SQDMA::      .WORD      0      ; DMA CONTROL & STATUS REGISTER
417      SQUBA::      .WORD      0      ; BUS ADDRESS UNIBUS REGISTER
418      SQBA::       .WORD      0      ; BUS ADDRESS REGISTER
419      SQWC::       .WORD      0      ; WORD COUNT REGISTER
420      SQMT0::      .WORD      0      ; MAINTENANCE REGISTER 0
421      SQMT1::      .WORD      0      ; MAINTENANCE REGISTER 1
422      SQVAD::      .WORD      0      ; VIEW ADDRESS REGISTER
423      SQVDA::      .WORD      0      ; DATA REGISTER
424
425      EXPECTED::
426      EXQCMD::      .WORD      0      ; USED FOR TABLE PROCESSING OF EXPECTED RESULTS
427      EXQVNT::      .WORD      0      ; COMMAND REGISTER
428      EXQINT::      .WORD      0      ; EVENT REGISTER
429      EXQCTL::      .WORD      0      ; INTERRUPT SUMMARY REGISTER
430      EXQDMA::      .WORD      0      ; CONTROL REGISTER
431      EXQUBA::      .WORD      0      ; DMA CONTROL & STATUS REGISTER
432      EXQBA::       .WORD      0      ; BUS ADDRESS UNIBUS REGISTER
433      EXQWC::       .WORD      0      ; BUS ADDRESS REGISTER
434      EXQMT0::      .WORD      0      ; WORD COUNT REGISTER
435      EXQMT1::      .WORD      0      ; MAINTENANCE REGISTER 0
436      EXQVAD::      .WORD      0      ; MAINTENANCE REGISTER 1
437      EXQVDA::      .WORD      0      ; VIEW ADDRESS REGISTER

```


GLOBAL DATA SECTION

```

434 002402 000000      EXQVDA::      .WORD  0      ; DATA REGISTER
435
436      ;
437      ;Q-BRIDGE REGISTER NAMES ARE STORED HERE. ACCESS IS INDEXED, EACH TAKING 2 WORDS
438      ;
439      .EVEN
440 002404      121      103      115 NAMES:: .ASCIZ  /QCMD/
      002407      104      000
441 002411      121      126      116      .ASCIZ  /QVNT/
      002414      124      000
442 002416      121      111      116      .ASCIZ  /QINT/
      002421      124      000
443 002423      121      103      124      .ASCIZ  /QCTL/
      002426      114      000
444 002430      121      104      115      .ASCIZ  /QDMA/
      002433      101      000
445 002435      121      125      102      .ASCIZ  /QUBA/
      002440      101      000
446 002442      121      102      101      .ASCIZ  /QBA /
      002445      040      000
447 002447      121      127      103      .ASCIZ  /QWC /
      002452      040      000
448 002454      121      115      124      .ASCIZ  /QMT0/
      002457      060      000
449 002461      121      115      124      .ASCIZ  /QMT1/
      002464      061      000
450 002466      121      126      101      .ASCIZ  /QVAD/
      002471      104      000
451 002473      121      126      104      .ASCIZ  /QVDA/
      002476      101      000
452      .EVEN
453
454      ;
455      ;      MISC STORAGE AREAS
456      ;
457
458 002500 000000      BRIDGETYPE::      .WORD  0      ;BRIDGE TYPE (0=Q-BRIDGE 1=U-BRIDGE)
459 002502 000000      BRDGFLG::      .WORD  0      ;BRIDGE CONFIGURED FLAG
460 002504 000000      LOGUNIT::      .WORD  0      ;USED TO STORE CURRENT LOGICAL UNIT
461 002506 000000      PLOC::      .WORD  0      ;POINTER TO CURRENT HARDWARE P-TABLE
462 002510 000000      BASEADD::      .WORD  0      ;STORES BASE ADDRESS
463 002512 000000      VECTOR::      .WORD  0      ;STORES VECTOR ADDRESS
464 002514 000000      LOOPBK::      .WORD  0      ;STORES SOFTWARE LOOPBACK QUES. VALUE FOR CURRENT UNIT
465 002516 000000      BRLEVEL::      .WORD  0      ;STORES BR LEVEL
466 002520 000000      INITWORD::      .WORD  0      ;STORES INITIALIZE WORD SENT TO QMT1
467 002522 000000      TRPFLG::      .WORD  0      ;TRAP FLAG INDICATOR
468 002524 000000      ADD::      .WORD  0      ;STORES ADDRESS OF REGISTER FOR MESSAGES
469 002526 000000      VAL::      .WORD  0      ;STORES VALUE OF REGISTER FOR MESSAGES
470 002530 000000      EXP::      .WORD  0      ;STORES EXPECTED VALUE OF TEST FOR MESSAGES
471 002532 000000      ERROR1::      .WORD  0      ;STORES ERROR COUNTS
472 002534 000000      ERROR2::      .WORD  0      ;STORES ERROR COUNTS
473 002536 000000      RAM::      .WORD  0      ;TELLS WHETHER RAM MESSAGE WAS PRINTED OUT
474 002540 000000      VALUE::      .WORD  0      ;STORES CURRENT CPU PRIORITY LEVEL
475 002542 000000      INTFLG::      .WORD  0      ;INTERRUPT FLAG
476 002544 000000      HP1::      .WORD  0      ;TELLS WHETHER DRS HELP MESSAGE ALREADY PRINTED
477 002546 000000      HP2::      .WORD  0      ;TELLS WHETHER DIA HELP MESSAGE ALREADY PRINTED
478 002550 000000      MEMSIZ::      .WORD  0      ;STORE FREE MEMORY SIZE IN WORDS ABOVE DIAGNOSTIC

```


B3

CZFA-A0 CTRL BRIDGE INTFC DIAG MACRO M1200 09 JAN-84 12:33 PAGE 19-2

SEQ 0027

GLOBAL DATA SECTION

479 002552 000000
480

FREMEM::

.WORD 0

;POINTS TO STARTING ADDRESS OF FREE MEMORY

GLOBAL TEXT SECTION

```

482          .SBTTL  GLOBAL TEXT SECTION
483
484          ;**
485          ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
486          ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
487          ; MORE THAN ONE TEST.
488          ;--
489
490          ;
491          ; NAMES OF DEVICES SUPPORTED BY PROGRAM
492          ;
493          002554          DEVTYP  <FEP>
          002554          L$DVTYP::
          002554          .ASCIZ  /FEP/
          002557          .EVEN
494
495          ; TEST DESCRIPTION
496          ;
497          ;
498          002560          DESCRIPT      <COMMAND BRIDGE INTERFACE DIAGNOSTIC>
          002560          L$DESC::
          002560          .ASCIZ  /COMMAND BRIDGE INTERFACE DIAGNOSTIC/
          002563          103          117          115
          002566          115          101          116
          002571          104          040          102
          002574          122          111          104
          002577          107          105          040
          002602          111          116          124
          002605          105          122          106
          002610          101          103          105
          002613          040          104          111
          002616          101          107          116
          002621          117          123          124
          111          103          000
          .EVEN
          .EVEN
499

```

GLOBAL TEXT SECTION

```

501      .NLIST BEX
502      ;
503      ; BIT NAMES IN THE REGISTERS
504      ;
505 002624      054      054      054      CMDBIT:: .ASCIZ /.....PRsie.....CMDACK/
506 002657      054      054      054      VNTBIT:: .ASCIZ /.....ACKIE.....VNTPRS/
507 002712      105      122      122      INTBIT:: .ASCIZ /ERR,UNEX,QNEX,UPWRFL,CMDPRS,VNTACK,RDY,,,ERRIE/
508 002771      054      054      054      CTLBIT:: .ASCIZ /.....QVREN,WDT1,WDTO,THRENB,WDTRST/
509 003042      054      054      121      DMABIT:: .ASCII /,,QXAD21,QXAD20,QXAD19,QXAD18,QXAD17,QXAD16,RDY,RDYIE/
510 003127      054      125      130      .ASCIZ /,UXAD17,UXAD16,,,DMAWT,GO/
511 003161      054      054      054      MT1BIT:: .ASCIZ /.....MIODIS,MQHLT,MQPFL,FUNC2,FUNC1,FUNCO,MVADR/
512
513 003244      STARS
                    ;*****
514
515      ; FORMAT STATEMENTS USED IN PRINT CALLS
516      ;
517 003244      STARS
                    ;*****
518
519      ;.IF NDF ONEFILE
520      .NLIST BEX
521      ;.ENDC
522
523 003244      045      116      061      CONFIG:: .ASCIZ /#N1#ACONFIGURED FOR #T#A BRIDGE MODULE#N1/
524 003316      121      055      102      QBUS:: .ASCIZ /Q-BUS/
525 003324      125      116      111      UBUS:: .ASCIZ /UNIBUS/
526
527 003333      045      116      061      FOR0:: .ASCIZ /#N1/
528 003337      045      116      061      FOR1:: .ASCIZ /#N1#ANAME#S4#AADDRESS#S4#ACONTENTS#S4#AEXPECTED (UNDETERMINED BITS ARE ZERO
) /
529 003454      045      116      061      FOR2:: .ASCIZ /#N1#T#S5#06#S5#06#S6#06/
530 003504      045      124      045      FOR3:: .ASCIZ /#T#A REGISTER AT #06#A CANNOT BE READ/
531 003552      045      116      061      FOR4:: .ASCIZ /#N1#S4#04#S9#06#S7#06/
532 003600      FOR4.5::
533 003600      045      116      061      .ASCIZ /#N1#AUNKNOWN VIEW RAM SIZE, 1K ASSUMED FOR REST OF TEST/
534 003670      045      116      061      FOR5:: .ASCIZ /#N1#AQVAD Q RAM ADDRESS INCREMENT SHOULD BE #04#A WAS #04/
535 003762      FOR5.5::
536 003762      045      116      061      .ASCIZ /#N1#AQVAD U RAM ADDRESS INCREMENT SHOULD BE #04#A WAS #04/
537 004054      045      116      061      FOR6:: .ASCIZ /#N1#ATOTAL RAM CHECK ERRORS #D5/
538 004114      045      116      061      FOR7:: .ASCIZ /#N1#ATOTAL INCREMENT ERRORS #D5/
539 004154      045      116      061      FOR8:: .ASCIZ /#N1#ARAM ADDRESS CONTENTS EXPECTED/
540 004227      045      116      061      FOR9:: .ASCIZ /#N1#AVIEW RAM SIZE #D1#AK/
541 004261      045      116      061      FOR10:: .ASCIZ /#N1#ASHOULD BE #06#A, WAS #06#A LOOP #D3#A OUT OF 100#N1/
542 004353      045      116      061      FOR11:: .ASCII /#N1#ATEST 37: BELL WILL RING EACH TIME WDTO OCCURS (20 TIMES /
543 004450      124      117      124      .ASCIZ /TOTAL)#N1/
544 004462      045      101      007      FOR12:: .ASCIZ /#A/<7>
545 004466      045      116      045      FOR13:: .ASCIZ /#N#ABR LEVEL #01#A CPU PRIORITY LEVEL #01#N1/
546      .EVEN

```


GLOBAL TEXT SECTION

548 004544

STARS

;*****

549

;

550

; ASCII MESSAGES USED IN ERROR REPORTING

551

;

552 004544

STARS

;*****

553

554

555 004544	102	122	111	MSG1::	.ASCIIZ	/BRIDGE REGISTER DUMP/
556 004571	102	101	104	MSG2::	.ASCIIZ	/BAD ADDRESS ERROR - PASS ABORTED/
557 004632	122	105	107	MSG3::	.ASCIIZ	/REGISTER READ AFTER RESET ERROR/
558 004672	114	117	101	MSG4::	.ASCIIZ	/LOAD WITH ONE'S AND READ ERROR/
559 004731	114	117	101	MSG5::	.ASCIIZ	/LOAD WITH ZERO'S AND READ ERROR/
560 004771	121	103	114	MSG6::	.ASCIIZ	/QCLR RESET ERROR/
561 005012	121	103	115	MSG7::	.ASCIIZ	/QCMD ROTATING 1 IN BITS 1-5 ERROR/
562 005054	121	103	115	MSG8::	.ASCIIZ	/QCMD ROTATING 0 IN BITS 1-5 ERROR/
563 005116	121	103	115	MSG9::	.ASCIIZ	/QCMD ROTATING 1 IN BITS 8-15 ERROR/
564 005161	121	103	115	MSG10::	.ASCIIZ	/QCMD ROTATING 0 IN BITS 8-15 ERROR/
565 005224	121	126	116	MSG11::	.ASCIIZ	/QVNT ROTATING 1 IN BITS 1-5 ERROR/
566 005266	121	126	116	MSG12::	.ASCIIZ	/QVNT ROTATING 0 IN BITS 1-5 ERROR/
567 005330	121	126	116	MSG13::	.ASCIIZ	/QVNT ROTATING 1 IN BITS 8-15 ERROR/
568 005373	121	126	116	MSG14::	.ASCIIZ	/QVNT ROTATING 0 IN BITS 8-15 ERROR/
569 005436	121	111	116	MSG15::	.ASCIIZ	/QINT BIT 11 (CMDPRS) SHOULD BE CLEAR AFTER SET QCMD CMDACK/
570 005531	121	111	116	MSG16::	.ASCIIZ	/QINT BIT 10 (VNTACK) SHOULD BE SET AFTER SET QCMD CMDACK/
571 005622	121	111	116	MSG17::	.ASCIIZ	/QINT BIT 10 (VNTACK) SHOULD BE CLEAR AFTER SET QVNT VNTPRS/
572 005715	121	111	116	MSG18::	.ASCIIZ	/QINT BIT 11 (CMDPRS) SHOULD BE SET AFTER SET QVNT VNTPRS/
573 006006	121	102	101	MSG19::	.ASCIIZ	/QBA ROTATING 1 ERROR/
574 006033	121	102	101	MSG20::	.ASCIIZ	/QBA ROTATING 0 ERROR/
575 006060	121	104	115	MSG21::	.ASCIIZ	/QDMA HI BITS (8-13) ROTATING 1 ERROR/
576 006125	121	104	115	MSG22::	.ASCIIZ	/QDMA HI BITS (8-13) ROTATING 0 ERROR/
577 006172	121	104	115	MSG23::	.ASCIIZ	/QDMA EXTENDED ADDRESS BITS 4 & 5 ERROR/
578 006241	121	125	102	MSG24::	.ASCIIZ	/QUBA ROTATING 1 ERROR/
579 006267	121	125	102	MSG25::	.ASCIIZ	/QUBA ROTATING 0 ERROR/
580 006315	121	127	103	MSG26::	.ASCIIZ	/QMC ROTATING 1 ERROR/
581 006342	121	127	103	MSG27::	.ASCIIZ	/QMC ROTATING 0 ERROR/
582 006367	121	126	101	MSG28::	.ASCIIZ	/QVAD Q-REGISTER ROTATING 1 ERROR/
583 006430	121	126	101	MSG29::	.ASCIIZ	/QVAD Q-REGISTER ROTATING 0 ERROR/
584 006471	121	126	101	MSG30::	.ASCIIZ	/QVAD U-REGISTER ROTATING 1 ERROR/
585 006532	121	126	101	MSG31::	.ASCIIZ	/QVAD U-REGISTER ROTATING 0 ERROR/
586 006573	121	126	101	MSG32::	.ASCIIZ	/QVAD Q-REGISTER INCREMENT ERROR/
587 006633	121	126	101	MSG33::	.ASCIIZ	/QVAD U-REGISTER INCREMENT ERROR/
588 006673	121	126	104	MSG34::	.ASCIIZ	/QVDA VIEW REGISTER RAM CHECK ERROR/
589 006736	121	126	101	MSG35::	.ASCIIZ	/QVAD RAM ADDRESS INCREMENT ERROR/
590 006777	121	102	101	MSG36::	.ASCIIZ	/QBA INCREMENT ERROR/
591 007023				MSG36.5::		
592 007023	121	104	115		.ASCIIZ	/QDMA INCREMENT BITS 8-13 EXTENDED ADDRESS ERROR/
593 007103	121	104	115	MSG37::	.ASCIIZ	/QDMA RDY BIT 7 SHOULD BE CLEARED AFTER SET QDMA GO BIT/
594 007172	121	127	103	MSG38::	.ASCIIZ	/QMC SHOULD HAVE INCREMENTED AFTER SET QDMA GO BIT/
595 007254	121	104	115	MSG39::	.ASCIIZ	/QDMA RDY BIT 7 SHOULD BE SET AFTER SET QDMA GO BIT/
596 007337	121	127	103	MSG40::	.ASCIIZ	/QMC INCREMENT AFTER QDMA DMAWT AND GO BIT SET ERROR/
597 007423	121	104	115	MSG41::	.ASCIIZ	/QDMA RDY BIT 7 SHOULD BE SET AFTER QDMA DMAWA AND GO BIT SET/
598 007520	104	115	101	MSG42::	.ASCIIZ	/DMA WRITE DATA TRANSFER ERROR/
599 007556	104	115	101	MSG42.5::	.ASCIIZ	/DMA WRITE HI MEMORY DATA TRANSFER ERROR/
600 007626	104	115	101	MSG43::	.ASCIIZ	/DMA READ DATA TRANSFER ERROR/
601 007663	127	101	124	MSG44::	.ASCIIZ	/WATCH DOG TIMER READ AFTER WRITE TIME CODE BITS 2 & 3 ERROR/
602 007756	015	012	124		.ASCIIZ	<15><12>/TEST ABORTED/

GLOBAL TEXT SECTION

603 007775	127	101	124	MSG45::	.ASCII	/WATCH DOG TIMER READ AFTER WRITE TMREN8 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

```

632
633
634
635
636
637
638
639
640
641
642
643
644 011702
    011702
645 011702 010246
646 011704 005002
647 011706 060102
648 011710 060102
649 011712 060102
650 011714 060102
651 011716 060102
652 011720 062702 002404
653 011724
    011724 013746 002524
    011730 010246
    011732 012746 003504
    011736 012746 000003
    011742 010600
    011744 104414
    011746 062706 000010
654 011752
    011752 012746 003333
    011756 012746 000001
    011762 010600
    011764 104414
    011766 062706 000004
655 011772 012602
656 011774
    011774
    011774 104423
657
658
659
660
661
662
663
664
665 011776
    011776
666 011776 010146
667 012000 010246
668 012002 005002
669 012004 060102
670 012006 060102
671 012010 060102
672 012012 060102

```

.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

```

```

ERR2:: BGNMSG ERR2
        MOV     R2, -(SP)
        CLR     R2                                ;CLEAR ACCUMULATOR
        ADD     R1,R2                              ;MULTIPLY REGISTER POINTER TIMES 5
        ADD     R1,R2
        ADD     R1,R2
        ADD     R1,R2
        ADD     R1,R2
        ADD     #NAMES,R2                          ;POINT R2 TO REGISTER NAME
        PRINTB  #FOR3,R2,ADD                       ;TELL OPERATOR BAD ADDRESS
        MOV     ADD, -(SP)
        MOV     R2, -(SP)
        MOV     #FOR3, -(SP)
        MOV     #3, -(SP)
        MOV     SP,R0
        TRAP    C#PNTB
        ADD     #10,SP
        PRINTB  #FOR0                              ;INSERT LINE FEED CARRIAGE RETURN
        MOV     #FOR0, -(SP)
        MOV     #1, -(SP)
        MOV     SP,R0
        TRAP    C#PNTB
        ADD     #4,SP
        MOV     (SP)+,R2                          ;RESTORE REGISTERS
        ENDMMSG
L10001: 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

```

```

ERR4:: BGNMSG ERR4
        MOV     R1, -(SP)                          ;SAVE REGISTERS USED IN ROUTINES
        MOV     R2, -(SP)
        CLR     R2                                ;CLEAR ACCUMULATOR
        ADD     R1,R2                              ;MULTIPLY REGISTER POINTER TIMES 5
        ADD     R1,R2
        ADD     R1,R2
        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			L10002:		
	012134	104423		TRAP	C#MSG	
683						
684						
685						
686						
687	012136					
	012136					
688	012136			ERR5::		
	012136	012746	004154	PRINTX	#FOR8	;PRINT HEADER
	012142	012746	000001	MOV	#FOR8,-(SP)	
	012146	010600		MOV	#1,-(SP)	
	012150	104415		MOV	SP,R0	
	012152	062706	000004	TRAP	C#PNTX	
689	012156			ADD	#4,SP	
	012156			ENDMSG		
	012156	104423		L10003:		
				TRAP	C#MSG	
690						
691						
692						
693	012160					
	012160					
694	012160	010146		ERR6::		
695	012162	010246		MOV	R1,-(SP)	;SAVE REGISTERS USED IN ROUTINES
696	012164	010346		MOV	R2,-(SP)	
697	012166	013701	002516	MOV	R3,-(SP)	
698	012172	013702	002540	MOV	BRLEVEL,R1	;DEVICE PRIORITY TO R1
699	012176	012703	000005	MOV	VALUE,R2	;CPU PRIORITY TO R2
				MOV	#5,R3	;LOOP COUNTER TO R3

GLOBAL ERROR REPORT SECTION

```

700 ;SHIFT PRIORITY RIGHT 5 BITS FOR MESSAGE
701 012202 006201 10$: ASR R1 ;SHIFT R1 RIGHT
702 012204 006202 ASR R2 ;SHIFT R2 RIGHT
703 012206 005303 DEC R3 ;DECREMENT COUNTER
704 012210 001374 BNE 10$ ;IF NOT DONE, BRANCH
705 ;PRINT ERROR MESSAGE
706 012212 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
707 012236 012603 MOV (SP)+,R3
708 012240 012602 MOV (SP)+,R2 ;RESTORE REGISTERS
709 012242 012601 MOV (SP)+,R1
710 012244 ENDMMSG
    012244 L10004: TRAP C#MSG
    012244 104423 ; BAD REGISTER VALUE ERROR (SINGLE REGISTER TEST)
711 ; USED WITH ROTATING BITS TESTS
712 ;REGISTER NUMBER (0-11.) EXPECTED TO BE IN R1
713 ;REGISTER ADDRESS EXPECTED TO BE IN LOCATION ADD
714 ;REGISTER VALUE EXPECTED TO BE IN LOCATION VAL
715 ;REGISTER EXPECTED VALUE EXPECTED TO BE IN LOCATION EXP
716
717
718 012246 BGNMSG ERR7
    012246 ERR7:: MOV R1,-(SP) ;SAVE REGISTERS USED IN ROUTINES
719 012246 010146 MOV R2,-(SP)
720 012250 010246 CLR R2 ;CLEAR ACCUMULATOR
721 012252 005002 ADD R1,R2 ;MULTIPLY REGISTER POINTER TIMES 5
722 012254 060102 ADD R1,R2
723 012256 060102 ADD R1,R2
724 012260 060102 ADD R1,R2
725 012262 060102 ADD R1,R2
726 012264 060102 ADD R1,R2
727 012266 062702 002404 ADD #NAMES,R2 ;POINT R2 TO REGISTER NAME
728 012272 006301 ASL R1 ;REGISTER POINTER TIMES 2
729 012274 016137 002274 002524 MOV ADDRESSES(R1),ADD ;MOVE CURRENT ADDRESS TO MESSAGE AREA
730 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
731 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
732 012360 012602 MOV (SP)+,R2 ;RESTORE REGISTERS

```

J3

CZFPA-AO CTRL BRIDGE INTFC DIAG MACRO M1200 09-JAN-84 12:33 PAGE 23-3

SEQ 0035

GLOBAL ERROR REPORT SECTION

733 012362 012601
734 012364
012364
012364 104423

MOV (SP)+,R1
ENDMSG
L10005:
TRAP C\$MSG

GLOBAL SUBROUTINES SECTION

```

736      .SBTTL  GLOBAL SUBROUTINES SECTION
737
738      ;**
739      ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
740      ; THAT ARE USED IN MORE THAN ONE TEST.
741      ;--
742
743      ;
744      ;      THIS SUBROUTINE PRINTS A ONE LINE DETAIL LINE WITHOUT THE
745      ;      THE PRIMARY REGISTER ERROR HEADER LINE.
746      ;
747      ; INPUTS:
748      ; REGISTER NUMBER (0-11.) EXPECTED TO BE IN R1
749      ; REGISTER ADDRESS EXPECTED TO BE IN LOCATION ADD
750      ; REGISTER VALUE EXPECTED TO BE IN LOCATION VAL
751      ; REGISTER EXPECTED VALUE EXPECTED TO BE IN LOCATION EXP
752      ;
753      ; CALLING SEQUENCE:
754      ;      JSR      PC.DETAIL
755      ;
756
757      012366
758      012366 010146
759      012370 010246
760      012372 005002
761      012374 060102
762      012376 060102
763      012400 060102
764      012402 060102
765      012404 060102
766      012406 062702 002404
767      012412 006301
768      012414 016137 002274 002524
769      012422
          012422 013746 002530
          012426 013746 002526
          012432 013746 002524
          012436 010246
          012440 012746 003454
          012444 012746 000005
          012450 010600
          012452 104415
          012454 062706 000014
770      012460 012602
771      012462 012601
772      012464 000207

      DETAIL:
      MOV      R1,-(SP)          ;SAVE REGISTERS USED IN ROUTINES
      MOV      R2,-(SP)
      CLR      R2                ;CLEAR ACCUMULATOR
      ADD      R1,R2             ;MULTIPLY REGISTER POINTER TIMES 5
      ADD      R1,R2
      ADD      R1,R2
      ADD      R1,R2
      ADD      R1,R2
      ADD      #NAMES,R2        ;POINT R2 TO REGISTER NAME
      ASL      R1                ;REGISTER POINTER TIMES 2
      MOV      ADDRESSES(R1),ADD ;MOVE CURRENT ADDRESS TO MESSAGE AREA
      PRINTX   #FOR2,R2,ADD,VAL,EXP ;PRINT FULL ERROR DETAILS
      MOV      EXP,-(SP)
      MOV      VAL,-(SP)
      MOV      ADD,-(SP)
      MOV      R2,-(SP)
      MOV      #FOR2,-(SP)
      MOV      #5,-(SP)
      MOV      SP,R0
      TRAP     C#PNTX
      ADD      #14,SP
      MOV      (SP)+,R2          ;RESTORE REGISTERS
      MOV      (SP)+,R1
      RETURN

```

```

774 ;*****
775 ;**
776 ;               BITS.MAC
777 ;   PROVIDES CODE FOR A SUBROUTINE TO INTERPRET THE BITS SET
778 ;   IN A WORD (OR REGISTER), AND OPTIONALLY FOR EVERY BIT SET PRINT OUT
779 ;   THE NAME OF THAT BIT IF DEFINED IN AN ASCIZ STATEMENT, AND ALWAYS
780 ;   PUT THE NAMES IN A BUFFER.
781 ;
782 ;   AUTHOR:  DALE PROCTOR
783 ;   DATE:    12/23/81
784 ;   VERSION: 1.00
785 ;
786 ;   ASCIZ STATEMENT FORMAT -
787 ;       LABEL: .ASCIZ /BIT15,BIT14,...,BIT10...BIT7/
788 ;   MAXIMUM NUMBER OF DEFINITIONS IS 16
789 ;
790 ;   TO USE THE FULL ROUTINE, CALL BY USING THE MACRO SETBITS.
791 ;   SETBITS IS IN THE FORMAT-
792 ;       SETBITS      NAME,POINTER,VALUE
793 ;       WHERE NAME IS AN ASCII NAME THAT WILL BE PRINTED IN THE
794 ;       MESSAGE LINE, POINTER IS AN ADDRESS THAT POINTS TO THE
795 ;       ASCIZ FORMAT STATEMENT THAT CONTAINS THE BIT NAMES, AND
796 ;       VALUE IS THE WORD TO INTERPRET AND CONVERT TO BIT NAMES.
797 ;
798 ;   TO CREATE ONLY A PRINT BUFFER WITH THE SET BIT NAMES CALL
799 ;   THIS ROUTINE WITH THE MACRO S$BITS.
800 ;   S$BITS IS IN THE FORMAT-
801 ;       S$BITS      POINTER,VALUE
802 ;       ON RETURN FROM S$BITS ROUTINE R0 CONTAINS THE ADDRESS
803 ;       OF THE BUFFER.
804 ;
805 ;--
806 ;*****
807 ;CALL BY S$BITS MACRO, NO PRINTING
808 012466 BIT$::
809 ;SAVE SOME REGISTERS
810 012466 010246      MOV     R2,-(SP)
811 012470 010346      MOV     R3,-(SP)
812 012472 010446      MOV     R4,-(SP)
813 012474 012503      MOV     (R5)+,R3      ;GET POINTER OF FORMAT IN R3
814 012476 012737 177777 013226      MOV     #-1,STATU$      ;TELL ROUTINE THAT THE S$BITS MACRO CALLED
815 012504 000411      BR      BUILD$      ;BRANCH OVER SETBITS STUFF
816 ;CALL BY SETBITS MACRO, PRINT LINE IF SOMETHING TO PRINT
817 012506 BITS$::
818 012506 005037 013226      CLR      STATU$      ;TELL ROUTINE THAT THE SETBITS MACRO CALLED
819 ;SAVE SOME REGISTERS
820 012512 010246      MOV     R2,-(SP)
821 012514 010346      MOV     R3,-(SP)
822 012516 010446      MOV     R4,-(SP)
823 012520 010502      MOV     R5,R2      ;GET ADDRESS IN R2
824 012522 061505      ADD     (R5),R5      ;ADD LENGTH OF DATA TO RETURN PC
825 012524 005722      TST     (R2)+      ;SKIP PAST LENGTH
826 012526 012203      MOV     (R2)+,R3      ;GET POINTER OF FORMAT IN R3
827 ;REGISTER USAGE
828 ;   R0 = LOOP COUNTER
829 ;   R1 = CONTAINS REGISTER CONTENTS
830 ;   R2 = POINTS TOWARD REGISTER NAME

```


GLOBAL SUBROUTINES SECTION

```

831      ;      R3 = POINTS TO DEFINITION FORMAT
832      ;      R4 = POINTS TO BUFFER
833      ;      R5 = CONTAINS RETURN ADDRESS
834      ;NOW BUILD BIT NAME BUFFER
835      BUILD$:
836      MOV     #MES2$,R4      ;POINT R4 TO BUFFER
837      MOV     #MES2$L,R0     ;LENGTH OF BUFFER TO CLEAR
838      ;ZERO OUT THE BUFFER
839      1$:      CLRB      (R4)+      ;CLEAR BYTE
840      DEC     R0
841      BNE     1$
842      MOV     #MES2$,R4      ;POINT R4 TO BUFFER
843      MOV     #16,,R0        ;NUMBER OF LOOPS TO LOOP COUNTER R1
844      MOV     R1,STORE$      ;SAVE ORIGINAL BIT PATTERN
845      CLR     NO$CHAR        ;CLEAR CHARACTER MOVED FLAG
846      10$:    CLRB      NO$CHAR+1    ;CLEAR BIT DEFINED AND MOVED FLAG
847      ROL     R1             ;CHECK BIT
848      BCS     20$           ;IF BITS SET, GO MOVE BIT NAME
849      ;BIT WAS NOT SET, FIND NEXT COMMA
850      5$:      CMPB      #0,(R3)      ;END OF DEFINITION?
851      BEQ     40$           ;IF END, GO PRINT MESSAGE
852      CMPB      #'',(R3)          ;IS IT A COMMA?
853      BEQ     35$           ;IF COMMA, CONTINUE TO PROCESS
854      INC     R3             ;POINT TO NEXT CHARACTER IN DEFINITION
855      BR      5$            ;CHECK NEXT CHARACTER IF NOT A COMMA
856      ;BIT WAS SET, MOVE BIT NAME IF THERE
857      20$:    CMPB      #0,(R3)      ;IS IT END OF FORMAT?
858      BEQ     45$           ;IF END, GO PRINT MESSAGE
859      CMPB      #'',(R3)          ;IS IT A COMMA?
860      BEQ     30$           ;IF IT WAS A COMMA, GO
861      ;SHOULD BE A VALID CHARACTER, MOVE TO PRINT BUFFER
862      MOVB      (R3)+,(R4)+      ;MOVE CHARACTER
863      INCB      NO$CHAR        ;INCREMENT CHARACTER FLAG
864      INCB      NO$CHAR+1      ;INCREMENT BIT DEFINED AND MOVED FLAG
865      BR      20$           ;GO PROCESS NEXT CHAR
866      ;END OF CHARACTER MOVE, ADVANCE POINTER TO PUT COMMA AFTER NAME
867      30$:    TSTB      NO$CHAR+1    ;WAS THE LAST BIT DEFINED
868      BEQ     35$           ;IF LAST BIT NOT DEFINED, BRANCH
869      MOVB      #'',(R4)+      ;PUT COMMA AFTER BIT NAME
870      MOVB      #40,(R4)+      ;PUT SPACE AFTER COMMA
871      INC     R3             ;POINT TO NEXT CHARACTER IN DEFINITION
872      DEC     R0             ;DECREMENT LOOP COUNTER
873      BNE     10$           ;IF NOT DONE, GO PROCESS NEXT BIT
874      ;NOW SET UP TO PRINT THE MESSAGE
875      40$:    DEC     R4             ;POINT BACK TO LAST COMMA AND SPACE
876      DEC     R4
877      MOVB      #0,(R4)        ;PUT ZERO AS TERMINATOR IN BUFFER
878      CLR     R0             ;DEFAULT RETURN CODE = NOTHING PRINTED
879      TST      STATU$        ;SEE WHICH MACRO CALLED ROUTINE
880      BNE     50$           ;BRANCH IF S$BITS MACRO CALLED
881      TST      NO$CHAR        ;SEE IF NOTHING PUT IN PRINT BUFFER
882      BEQ     50$           ;IF NOTHING THERE, DON'T PRINT IT
883      PRINTB   #MES1$,R2,STORE$,#MES2$      ;PRINT LINE
884      MOV     #MES2$,-(SP)
885
886      012530      012704      013015
887      012530      012700      000204
888
889      012540      105024
890      012542      005300
891      012544      001375
892      012546      012704      013015
893      012552      012700      000020
894      012556      010137      013222
895      012562      005037      013224
896      012566      105037      013225
897      012572      006101
898      012574      103410
899
900      012576      122713      000000
901      012602      001433
902      012604      122713      000054
903      012610      001425
904      012612      005203
905      012614      000770
906
907      012616      122713      000000
908      012622      001425
909      012624      122713      000054
910      012630      001406
911
912      012632      112324
913      012634      105237      013224
914      012640      105237      013225
915      012644      000764
916
917      012646
918      012646      105737      013225
919      012652      001404
920      012654      112724      000054
921      012660      112724      000040
922      012664      005203
923      012666      005300
924      012670      001336
925
926      012672
927      012672      005304
928      012674      005304
929      012676      112714      000000
930      012702      005000
931      012704      005737      013226
932      012710      001020
933      012712      005737      013224
934      012716      001415
935      012720
936      012720      012746      013015

```


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
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 = #06\$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

GLOBAL SUBROUTINES SECTION

```

903 013230          STARS
904                ;*****
905                ;**
906                ; SUBROUTINE DUMP -
907                ; FUNCTIONAL DESCRIPTION:
908                ; SUBROUTINE TO PRINT THE CONTENTS OF ALL THE REGISTERS
909                ; IT USES THE PRINTX PRINT EXTENDED FORMAT AND CAN BE DISABLED
910                ; BY THE OPERATOR BY /FLAG:IXE, INHIBIT EXTENDED ERROR REPORTS
911                ; INPUTS:
912                ; NONE
913                ; OUTPUT:
914                ; DEVICE REGISTER DUMP PRINTED IF EXTENDED ERROR REPORTS ENABLED
915                ; CALLING SEQUENCE:
916                ; JSR      PC,DUMP
917          STARS
918          ;*****
919          DUMP::
920          ;FIRST CHECK FOR INHIBIT EXTENDED ERROR REPORTS FLAG SET
921          RFLAGS FLAG
922          TRAP   C0RFLA
923          MOV    R0,FLAG
924          BIT    @BIT11,FLAG
925          BNE    EDUMP
926          MOV    R1,-(SP)
927          MOV    R2,-(SP)
928          MOV    R3,-(SP)
929          MOV    R4,-(SP)
930          MOV    R5,-(SP)
931          MOV    @QCTL,-(SP)
932          ;FIRST READ THE REGISTERS
933          BIS    @QVREN,@QCTL
934          MOV    @12,R1
935          MOV    @CMD,R2
936          MOV    @VALUES,R3
937          CMP    @7,R1
938          BNE    110
939          JMP    P1
940          CMP    @2,R1
941          BNE    120
942          JMP    P2
943          MOV    (R2)+,(R3)+
944          DEC    R1
945          BNE    E100
946          ;END READ, NOW PRINT THE RESULTS
947          PRINTX @RP1
948          MOV    @RP1,-(SP)
949          MOV    @1,-(SP)
950          MOV    SP,R0
951          TRAP   C0PNTX
952          ADD    @4,SP
953          MOV    @NAMES-5,R4
954          CLR    R2
955          MOV    @12,R3
956          ADD    @5,R4
957          MOV    ADDRESSES(R2),ADD
958          MOV    VALUES(R2),VAL
959          ;PRINT FIRST LINE OF MESSAGE
960          ;POINT R4 TO REGISTER NAMES TABLE
961          ;CLEAR POINTER REGISTER
962          ;LOAD R3 AS LOOP COUNTER
963          ;POINT R4 TO NEXT REGISTER NAME
964          ;GET ADDRESS OF REGISTER READY TO PRINT
965          ;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      X1:      ADD      @2,R2          ;POINT R2 TO NEXT REGISTER DATA
966 013514 005303          DEC      R3          ;DECREMENT LOOP COUNTER
967 013516 001325          BNE      E64          ;GO DO NEXT REGISTER IF NOT DONE
968 013520          PRINTX  @RP0
      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
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

```


GLOBAL SUBROUTINES SECTION

```

995                                     ;PRINT BIT DEFINITION LINE
996 013666 PRINTIT:
997 013666 010005 MOV R0,R5 ;MOVE ADDRESS TO R5
998 013670 105715 TSTB (R5) ;IS IT A BLANK
999 013672 001706 BEQ X8 ;BRANCH IF NOTHING TO PRINT
1000 013674 ;PRINT LINE
      013674 010546 PRINTB @RP4,R5
      013676 012746 014165 MOV R5,-(SP)
      013702 012746 000002 MOV @RP4,-(SP)
      013706 010600 MOV @2,-(SP)
      013710 104414 MOV SP,R0
      013712 062706 000006 TRAP C:PNTB
      013716 000674 ADD @6,SP
                                     ;RETURN TO MAINLINE
1001 BR X8
1002
1003
1004 ;PROCESS QUBA REGISTER (NOT NORMAL REGISTER READ
1005 013720 017746 166372 P1: MOV @QMT1,-(SP) ;SAVE CURRENT QMT1
1006 013724 052777 000200 166364 BIS @MIODIS,@QMT1 ;ENABLE INTERNAL LOOPBACK
1007 013732 052777 000020 166356 BIS @BIT4,@QMT1 ;SELECT UBAR FUNCTION
1008 013740 017723 166350 MOV @QMT0,(R3)+ ;READ QUBA
1009 013744 005722 TST (R2)+ ;CORRECT R2
1010 013746 042777 000200 166342 BIC @MIODIS,@QMT1 ;DISABLE INTERNAL LOOPBACK
1011 013754 042777 000020 166334 BIC @BIT4,@QMT1 ;DESELECT UBAR FUNCTION
1012 013762 012677 166330 MOV (SP)+,@QMT1 ;RESTORE QMT1
1013 013766 000137 013334 JMP E5 ;BRANCH BACK
1014
1015 ;PROCESS QVAD REGISTER READ (NOT NORMAL REGISTER READ)
1016 013772 017746 166320 P2: MOV @QMT1,-(SP) ;SAVE CURRENT QMT1
1017 013776 052777 000200 166312 BIS @MIODIS,@QMT1 ;ENABLE INTERNAL LOOPBACK
1018 014004 052777 000002 166304 BIS @MVADR,@QMT1 ;ENABLE ADDRESS REGISTER READ
1019 014012 017723 166302 MOV @QVAD,(R3)+ ;READ ADDRESS REGISTER
1020 014016 005722 TST (R2)+ ;CORRECT R2
1021 014020 042777 000200 166270 BIC @MIODIS,@QMT1 ;DISABLE INTERNAL LOOPBACK
1022 014026 042777 000002 166262 BIC @MVADR,@QMT1 ;DISABLE ADDRESS REGISTER READ
1023 014034 012677 166256 MOV (SP)+,@QMT1 ;RESTORE QMT1
1024 014040 000137 013334 JMP E5 ;BRANCH BACK
1025
1026 ;FORMATS USED IN REGISTER DUMP PRINTING
1027
1028 014044 045 116 061 RP0:: .ASCIZ /#N1/ ;LINE FEED CARRIAGE RETURN ONLY
1029 014050 045 116 061 RP1:: .ASCIZ /#N1#S7#AREGISTER DUMP#N1#ANAME#S4#ADDRESS#S4#ACONTENTS/
1030 .EVEN
1031 014140 045 116 061 RP2:: .ASCIZ /#N1#T#S5#06#S5#06#S3/
1032 014165 045 124 000 RP4:: .ASCIZ /#T/
1033 .EVEN
1034 014170 000000 FLAG: .WORD 0 ;STORES RETURNED FLAG SETTINGS

```

GLOBAL SUBROUTINES SECTION

1036 014172

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051 014172

1052

1053 014172

1054 014172 010537 014366

1055 014176

014176 010500

014200 104434

1056

1057 014202 012702 000015

1058 014206 004537 014270

1059 014212 012702 000012

1060 014216 004537 014270

1061 014222 005001

1062 014224

014224 104426

014226 110002

1063 014230

014230 103011

1064 014232 022702 000000

1065 014236 001406

1066 014240 022702 000011

1067 014244 001437

1068 014246 004537 014270

1069 014252 000764

1070

1071 014254

1072 014254

014254 104435

1073 014256 013705 014366

1074 014262 062705 000012

1075 014266 000205

1076

1077 014270 010237 014364

1078 014274

014274 012746 014362

014300 012746 000001

014304 010600

014306 104414

014310 062706 000004

1079 014314 005201

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

;*****

;HLP::

MOV R5,\$STORE

;SAVE RETURN ADDRESS

OPEN R5

;OPEN FILE NAME

MOV R5,R0

TRAP C\$OPEN

;FIRST PRINT CR/LF

MOV #15,R2

;CARRIAGE RETURN TO R2

JSR R5,60\$

;GO PRINT IT

MOV #12,R2

;LINE FEED TO R2

JSR R5,60\$

;GO PRINT IT

CLR R1

;CLEAR POSITION COUNTER

10\$: GETBYTE R2

;GET BYTE COUNT OF RECORD

TRAP C\$GETB

MOVB R0,R2

BNCOMPLETE 50\$

;BRANCH IF END OF FILE

BCC 50\$

CMP #0,R2

;END OF RECORDS?

BEQ 50\$

;BRANCH IF END

30\$: CMP #11,R2

;IS IT TAB CHARACTER?

BEQ 80\$

;IF YES HANDLE TAB

40\$: JSR R5,60\$

;GO PRINT IT

BR 10\$

;GO PROCESS NEXT CHARACTER

;END OF FILE OR LAST RECORD

50\$:

CLOSE

;CLOSE FILE

TRAP C\$CLOS

MOV \$STORE,R5

;RESTORE RETURN ADDRESS

ADD #10.,R5

;SKIP PAST FILE NAME

RTS R5

;PRINT AND COUNT ROUTINE

60\$: MOV R2,\$HESLNE

;GET CHARACTER TO PRINT IN R2

PRINTB #HLPFOR

;GO PRINT LINE

MOV #HLPFOR,-(SP)

MOV #1,-(SP)

MOV SP,R0

TRAP C\$PNTB

ADD #4,SP

INC R1

;INCREMENT POSITION COUNTER

GLOBAL SUBROUTINES SECTION

Address	Hex	Dec	Label	Op	Opnd	Comment
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		CLR	R1	;CLEAR POSITION COUNTER
1087	014342	000205		RTS	R5	;FROM PRINT AND COUNT ROUTINE
1088						
1089	014344					
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						
1097						
1098	014362					
1099	014362	045	101			
1100	014364					
1101	014364	000				
1102	014365	000				
1103						
1104	014366	000000				

GLOBAL SUBROUTINES SECTION

1106 014370

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119 014370

1120

1121 014370

1122 014370 005001

1123 014372 012702 172300

1124 014376 012703 000007

1125 014402 012712 077406

1126 014406 010162 000040

1127 014412 005722

1128 014414 062701 000200

1129 014420 005303

1130 014422 001367

1131 014424 012712 077406

1132 014430 012762 177600 000040

1133 014436 000207

STARS

;*****

; **

; MMINIT

; FUNCTIONAL DESCRIPTION-

; SUBROUTINE TO INITIALIZE MEMORY MANAGEMENT REGISTERS

; INPUTS-

; NONE

; OUTPUT-

; NONE

; CALLING SEQUENCE-

; JSR PC,MMINIT

; --

STARS

;*****

MMINIT::

CLR R1

MOV #KISDR0,R2

MOV #7,R3

1\$: MOV #77406,(R2)

MOV R1,40(R2)

TST (R2)+

ADD #200,R1

DEC R3

BNE 1\$

MOV #77406,(R2)

MOV #177600,40(R2)

RTS PC

;GET DESCRIPTOR REGISTER ADDRESS

;DO 7 REGISTERS

;SET 4K, R/W UP

;SET ADDRESS REGISTER

;SET TO NEXT REGISTER ADDRESS

;DONE?

;IF NOT DONE

;SET I/O PAGE

GLOBAL SUBROUTINES SECTION

```

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

```

GLOBAL SUBROUTINES SECTION

```
11
12      .ENABL  AMA
40      .SBTTL  REPORT CODING SECTION
41 014446      BGNMOD
42
43      ;**
44      ; THE REPORT CODING SECTION CONTAINS THE
45      ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
46      ;--
47
48 014446      BGNRPT
49 014446      L$RPT::
50 014446 004737 013230      CALL  DUMP                      ;GO PRINT THE DUMP ROUTINE
61
62 014452      EXIT  RPT
63 014452 000167      .WORD  J$JMP
64 014454 000000      .WORD  L10007-2-.
74
75      .EVEN
76
77 014456      ENDRPT
78 014456      L10007:
79 014456 104425      TRAP  C$RPT
```


PROTECTION TABLE

79
80
81
82
83
84
85
86
87
88
89
90
91
92
93014460
014460
014460 177777
014462 177777
014464 177777

014466

.SBTTL PROTECTION TABLE

; **
; THIS TABLE IS USED BY THE RUNTIME SERVICES
; TO PROTECT THE LOAD MEDIA.
; --BGNPROT
L\$PROT::-1
-1
-1; OFFSET INTO P-TABLE FOR CSR ADDRESS
; OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
; OFFSET INTO P-TABLE FOR DRIVE NUMBER

ENDPROT

INITIALIZE SECTION

```

108      .SBTTL  INITIALIZE SECTION
109
110      ;**
111      ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
112      ; AT THE BEGINNING OF EACH PASS.
113      ;--
114
115      014466      BGNINIT
116      014466      L$INIT::
117      014466      SETPRI  #PRI00      ;INITIALIZE WITH CPU ZERO PRIORITY
118      014466      MOV      #PRI00,R0
119      014472      TRAP     C$SPRI
120
121      014474      INIT0:  READEF  #EF.CONTINUE      ; WAS THERE A CONTINUE COMMAND
122      014474      MOV      #EF.CONTINUE,R0
123      014500      TRAP     C$REFG
124      014502      BNCOMPLETE INIT1      ;IF NO, CONTINUE
125      014502      BCC      INIT1
126      014504      JMP      ENDI      ;YES, SKIP INITIALIZATION
127
128      014510      INIT1:  READEF  #EF.NEW
129      014510      MOV      #EF.NEW,R0      ;NEW PASS OR SUB PASS?
130      014514      TRAP     C$REFG
131      014516      BNCOMPLETE NEXT
132      014516      BCC      NEXT      ;IF NOT NEW PASS, SKIP SETUP
133
134      014520      INIT2:  TST      BRDGLG      ;HAS BRIDGE TYPE BEEN DETERMINED?
135      014524      BNE      INIT3      ;IF NE YES
136      014526      INC      BRDGLG      ;SET FLAG
137      014532      CLR      BRIDGETYPE      ;CLEAR BRIDGE TYPE FLAG (Q-BRIDGE)
138      014536      MOV      #QBUS,R2      ;LOAD BRIDGE TYPE PTR (Q-BRIDGE)
139      014542      READBUS
140      014542      TRAP     C$RDBU      ;DETERMINE BRIDGE TYPE
141      014544      BCOMPLETE 10$
142      014544      BCS      10$      ;IF COMPLETE THEN Q-BRIDGE
143      014546      MOV      #-1,BRIDGETYPE      ;SET BRIDGE TYPE FLAG (U-BRIDGE)
144      014554      MOV      #UBUS,R2      ;LOAD BRIDGE TYPE PTR (U-BRIDGE)
145      014560      CLR      MMR3      ;DISABLE UNIBUS MAPPING      ;V C.O
146      014564      10$:  PRINTB  #CONFIG,R2      ;IDENTIFY BRIDGE TYPE
147      014564      MOV      R2,-(SP)
148      014566      MOV      #CONFIG,-(SP)
149      014572      MOV      #2,-(SP)
150      014576      MOV      SP,R0
151      014600      TRAP     C$PNTB
152      014602      ADD      #6,SP
153
154      ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
155      ; HELP FILE SECTION REMOVED.  TO REINSTALL, REMOVED COMMENTS
156      ; !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
157
158      014606      INIT3:
159      014606      TST      HP1      ;WAS HELP FILE PRINTED
160      014606      BNE      1$      ;BRANCH IF ALREADY PRINTED
161      014606      INC      HP1      ;SET HELP FILE QUESTION FLAG
162      014606      GMANIL  HELP1,VAL,1,YES      ;ASK IF HELP FILE WANTED
163      014606      CMP      #NO,VAL      ;DO WE WANT TO PRINT HELP FILE?
164      014606      BEQ      1$      ;IF NO, BRANCH

```

INITIALIZE SECTION

148	:	HLPMAC	DRS.HLP	:	PRINT OUT SUPERVISOR HELP FILE
149	:1:	TST	HP2	:	WAS HELP FILE PRINTED
150	:	BNE	SETUP	:	BRANCH IF ALREADY PRINTED
151	:	INC	HP2	:	SET HELP FILE QUESTION FLAG
152	:2:	GMANIL	HELP2,VAL,1,YES	:	ASK IF DIAGNOSTIC HELP FILE WANTED
153	:	CMP	#NO,VAL	:	DO WE WANT TO PRINT HELP FILE
154	:	BEQ	SETUP	:	IF NO, BRANCH
155	:	HLPMAC	Z0791.HLP	:	PRINT OUT DIAGNOSTIC HELP FILE
156					
157	014606 012737 177777 002504	SETUP:	MOV	#-1,LOGUNIT	:INITIALIZE LOGICAL UNIT COUNTER
158					

INITIALIZE SECTION

```

160
161 014614 005237 002504      NEXT:  INC      LOGUNIT      ;POINT TO NEXT LOGICAL UNIT
162 014620 023737 002504 002012  CMP      LOGUNIT,L#UNIT ;HAVE WE PASSED MAXIMUM
163 014626 001500          BEQ      ABORT      ;GO IF MAXIMUM PASSED
164 014630          GPHARD  LOGUNIT,PLOC    ;GET P-TABLE ADDRESS
      014630 013700 002504      MOV      LOGUNIT,R0
      014634 104442          TRAP      C#GPHRD
      014636 010037 002506      MOV      R0,PLOC
165 014642          BNCOMPLETE NEXT      ;IF NOT AVAILABLE, GET NEXT UNIT
      014642 103364          BCC      NEXT
166 014644 013737 002504 002074  MOV      LOGUNIT,L#LUN    ;STORE CURRENT LOGICAL UNIT IN HEADER
167          ;GET ALL HARDWARE TABLE ENTRIES IN STORAGE
168 014652 013700 002506      MOV      PLOC,R0      ;PLACE HEADER POINTER IN R0
169 014656 016037 000000 002510  MOV      0(R0),BASEADD ;BASE ADDRESS OF BOARD
170 014664 016037 000002 002512  MOV      2(R0),VECTOR  ;VECTOR ADDRESS OF BOARD
171 014672 016037 000004 002514  MOV      4(R0),LOOPBK   ;LOOPBACK OPTION INSTALLED?
172 014700 016037 000006 002516  MOV      6(R0),BRLEVEL ;BR INTERRUPT LEVEL ON THE BOARD
173          ;FILL IN BRIDGE INTERFACE REGISTER TABLE
174 014706 013701 002510      MOV      BASEADD,R1      ;GET BASE ADDRESS IN R1
175 014712 012702 002274      MOV      #QCMD,R2      ;POINT TO REGISTER TABLE ADDRESS
176 014716 012703 000014      MOV      #12,,R3      ;# OF REGISTERS IN TABLE
177 014722 010122          10$: MOV      R1,(R2)+      ;PLACE REGISTER ADDRESS IN TABLE
178 014724 062701 000002      ADD      #2,R1      ;ADJUST R1 TO NEXT REGISTER ADDRESS
179 014730 005303          DEC      R3      ;ARE WE DONE?
180 014732 001373          BNE      10$      ;IF NOT DONE, GO AGAIN
181          ;TABLE BUILT, NOW ESTABLISH INITIALIZE WORD FOR TEST (LOOPBACK YES OR NO INSTALLED)
182 014734 005037 002520      CLR      INITWORD      ;CLEAR STORAGE
183 014740 005737 002514      TST      LOOPBK      ;IS LOOPBACK CABLE INSTALLED
184 014744 001003          BNE      20$      ;IF LOOPBACK CABLE INSTALLED, BRANCH
185 014746 052737 000200 002520  BIS      #MIODIS,INITWORD ;SET LOOPBACK ENABLED (NO CABLE INSTALLED)
186 014754          20$:
187          ;GET FREE MEMORY ADDRESS AND SIZE
188 014754          MEMORY  FREMEM      ;GET MEMORY ADDRESS
      014754 104431          TRAP      C#MEM
      014756 010037 002552      MOV      R0,FREMEM
189 014762 017737 165564 002550  MOV      #FREMEM,MEMSIZ ;GET FREE MEMORY SIZE
190          ;CHECK MEMORY SIZE
191 014770 022737 002000 002120  CMP      #2000,L#HIME ;> 32K WORD MEMORY? ;V C.0
192 014776 003004          BGT      22$      ;IF NOT ;V C.0
193 015000 005037 177572          CLR      MMR0      ;DISABLE MEM. MGT. ;V C.0
194 015004 005037 172516          CLR      MMR3      ;DISABLE UMR ;V C.0
195 015010 022737 007577 002120 22$: CMP      #7577,L#HIME ;COMPARE 128K TO MEMORY SIZE
196 015016 100003          BPL      25$      ;OK IF SIZE <= 128K
197 015020 012737 007577 002120  MOV      #7577,L#HIME ;RESET MEMORY SIZE TO 128K
198 015026 000401          BR      ENDI      ;SKIP ABORT INSTRUCTION
199 015030          ABORT: DOCLN      ;DO CLEANUP AND ABORT THE PASS
      015030 104444          TRAP      C#DOCLN
200 015032          ENDI:  EXIT      INIT
      015032 104432          TRAP      C#EXIT
      015034 000002          .WORD    L10011-.
201
225
226          ;      EXIT      INIT
227
239          .EVEN
240
241

```

N4

CZFPA-A0 CTRL BRIDGE INTFC DIAG MACRO M1200 09-JAN-84 12:33 PAGE 34-1

SEQ 0052

INITIALIZE SECTION

242 015036
015036
015036 104411

ENDINIT
L10011: TRAP C\$INIT

AUTODROP SECTION

244		.SBTTL AUTODROP SECTION
245		
246		;
247		; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
248		; THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
249		; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
250		; DROPPED FROM TESTING.
251		;
252		
253	015040	BGNAUTO
	015040	L\$AUTO::
254		
255	015040 000240	NOP ; NO ON-LINE SIGNAL PROVIDED TO USER FOR AUTODROP
256		
263		
264	015042	ENDAUTO
	015042	L10012:
	015042 104461	TRAP C\$AUTO

CLEANUP CODING SECTION

```

266
267
268
269
270
271
272
273
274
275
276 015044
    015044
277
278 015044
    015044 012746 000340
    015050 012746 015112
    015054 012746 000004
    015060 012746 000003
    015064 104437
    015066 062706 000010
279 015072 042777 000200 165216
280 015100
    015100 012700 000004
    015104 104436
281 015106
    015106 104432
    015110 000010
282
283 015112 000002
284
293
294 015114
    015114 104432
    015116 000002
295
307
308
309
310 015120
    015120
    015120 104412

.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.
;--

        BGNCLN
L$CLEAN::

        SETVEC  #4,#BAD,#PRI07          ; SET FOR BAD ADDRESS TRAP
        MOV     #PRI07,-(SP)
        MOV     #BAD,-(SP)
        MOV     #4,-(SP)
        MOV     #3,-(SP)
        TRAP    C$SVEC
        ADD     #10,SP
        BIC     #MIODIS,#QMT1          ; CLEAR MAINTENANCE LOOP-BACK BIT
        CLRVEC  #4                      ; RESET VECTOR
        MOV     #4,R0
        TRAP    C$CVEC
        EXIT    CLN
        TRAP    C$EXIT
        .WORD   L10013-

BAD:     RTI

        EXIT    CLN
        TRAP    C$EXIT
        .WORD   L10013-

        .EVEN

        ENDCLN
L10013:  TRAP    C$CLEAN

```

DROP UNIT SECTION

```

312
313
314
315
316
317
318
319 015122
    015122
320
321 015122 000240
322
331
332 015124
    015124 000167
    015126 000000
333
345
346
347
348 015130
    015130
    015130 104453

.SBTTL DROP UNIT SECTION
;
; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO NO LONGER BE TESTED.
;
L#DU:: BGNDU
      NOP          ;NO SPECIAL DROP UNIT CODING NEEDED

      EXIT        DU
      .WORD       J#JMP
      .WORD       L10014-2-.

      .EVEN

      ENDDU
L10014: TRAP      C#DU

```

ADD UNIT SECTION

```

350          .SBTTL  ADD UNIT SECTION
351
352          ;**
353          ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
354          ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
355          ; TO THE TEST CYCLE.
356          ;--
357
358 015132      BGNAU
359 015132
360 015132 000240      NOP          ;NO SPECIAL ADD UNIT CODING NEEDED
361
362
363
364
365
366
367 015134      EXIT      AU
368 015134 000167      .WORD  J&JMP
369 015136 000000      .WORD  L10015-2-.
370
371
372
373
374
375
376
377          .EVEN
378
379
380
381
382
383
384
385
386
387 015140      ENDAU
388 015140
389 015140 104452      L10015:
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```


F5

CZFPA-AO CTRL BRIDGE INTFC DIAG MACRO M1200 09-JAN-84 12:33 PAGE 40

SEQ 0057

ADD UNIT SECTION

11
13
15
44

.ENABL AMA
.DSABL GBL

ADD UNIT SECTION

```

46 015142
47
48 015142      040      122      105
015142
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63 015152      BGNTST
015152
64 015152      T1::
015152      CKPNT
015154      104421      001000      TRAP      C:RFLA      ; PRINT OUT TEST TITLE IF PNT SET
015160      001012      CMP      @PNT,R0      ; IS PNT FLAG SET?
015162      012746      015142      BNE      64:      ; BRANCH IF NOT SET
015166      012746      002270      MOV      @T1MSG,-(SP)
015172      012746      000002      MOV      @PNTMSG,-(SP)
015176      010600      MOV      @2,-(SP)
015200      104417      MOV      SP,R0
015202      062706      000006      TRAP      C:PNTF
015206      005037      002532      ADD      @6,SP
65 015212      64:      CLR      ERROR1      ; CLEAR ERROR COUNTER
015212      012746      000340      SETVEC      @4,@TRAP4,@PRI07      ; SET FOR BAD ADDRESS TRAP
015216      012746      014440      MOV      @PRI07,-(SP)
015222      012746      000004      MOV      @TRAP4,-(SP)
015226      012746      000003      MOV      @4,-(SP)
015232      104437      MOV      @3,-(SP)
015234      062706      000010      TRAP      C:SVEC
66 015240      5:      ADD      @10,SP
015240      104422      5:      BREAK
67 015242      005301      TRAP      C:BRK
68 015244      001375      DEC      R1      ; DECREMENT DELAY COUNTER
69 015246      004537      016002      BNE      5:      ; LOOP IF NOT DONE
70      6:      JSR      R5,BRESET      ; GO DO BUS RESET
71      ;
72      ; CHECK REGISTER ADDRESSING
73 015252      005037      002522      ;
74 015256      012701      177777      CLR      TRPFLG      ; CLEAR TRAP FLAG
75 015262      013702      002274      MOV      @-1,R1      ; COUNTER IN R1
76 015266      005201      MOV      QCMD,R2      ; REGISTER TABLE BASE TO R2
77 015270      022701      000014      10:      INC      R1      ; INCREMENT COUNTER
78 015274      001423      CMP      @12,R1      ; DONE YES?
79 015276      012203      BEQ      CONT      ; BRANCH IF DONE WITH CHECK
80 015300      005737      002522      MOV      (R2)+,R3      ; READ REGISTER
81 015304      001770      TST      TRPFLG      ; ANY TRAPS?
82      BEQ      10:      ; BRANCH IF NO TRAPS
83      ;
      ; MUST HAVE BEEN BAD ADDRESS

```

TEST1: RESET

```

84
85 015306 005037 002522      ; CLR TRPFLG      ; CLEAR TRAP INDICATOR
86 015312 010237 002524      ; MOV R2,ADD     ; PUT OFFENDING ADDRESS IN MEM
87 015316 162737 000002 002524 ; SUB #2,ADD     ; CORRECT FOR AUTO INCREMENT
88 015324      104455      ; ERRDF 1,MSG2,ERR2 ; DEVICE FATAL ERROR MESSAGE
      015324 000001      ; TRAP C#ERDF
      015326 004571      ; .WORD 1
      015330 011702      ; .WORD MSG2
      015332      000004      ; .WORD ERR2
89 015334      012700      ; CLRVEC #4      ; RESET VECTOR
      015334 104436      ; MOV #4,R0
      015340      000000      ; TRAP C#CVEC
90 015342      104444      ; DOCLN C#DCLN   ; EXIT PASS IF BAD ADDRESS
      015342      000000      ; TRAP C#DCLN
91
92
93      ; NOW TEST ALL THE REGISTERS
94      ;
95 015344      000000      ; CONT:
96 015344      000000      ; BGNSUB
      015344 104402      ; T1.1:
      015346      000000      ; TRAP C#BSUB
      015346 104420      ; INLOOP
      015346 104420      ; TRAP C#INLP
97 015346      000000      ; BNCOMPLETE 11#
98 015350      000000      ; BCC 11#
      015350 103002      ; JSR R5,BRESET
99 015352 004537 016002      ;
100 015356      000000      ; 11#
101
102      ; TEST QCMD FOR EXPECTED RESULTS
103      ;
104 015356 017737 164712 002526 ; MOV BQCMD,VAL
105 015364 042737 177476 002526 ; BIC #177476,VAL
106 015372 012737 000000 002530 ; MOV #0,EXP
107 015400 023737 002530 002526 ; CMP EXP,VAL
108 015406 001406      000000 ; BEQ 30#
109
110      ; ERROR CONDITION 1
111      ;
112 015410 012701 000000      ; MOV #0,R1
113 015414      000000      ; ERRHRD 1,MSG3,ERR4
      015414 104456      ; TRAP C#ERRHRD
      015416 000001      ; .WORD 1
      015420 004632      ; .WORD MSG3
      015422 011776      ; .WORD ERR4
114 015424      000000      ; 30#
115 015424      000000      ; ENDSUB
      015424 104403      ; L10017:
      015426      000000      ; TRAP C#ESUB
      015426      000000      ; BGNSUB
116 015426      000000      ; T1.2:
      015426 104402      ; TRAP C#BSUB
      015430      000000      ; INLOOP
      015430 104420      ; TRAP C#INLP
117 015430      000000      ; BNCOMPLETE 31#
118 015432      000000      ; BCC 31#
      015432 103002      ; JSR R5,BRESET
119 015434 004537 016002      ; IF ERROR LOOP, BRANCH

```


TEST1: RESET

120	015440				314:		
121					:		
122					:	TEST QVNT REGISTER	
123					:		
124	015440	017737	164632	002526		MOV	QVNT,VAL
125	015446	042737	177476	002526		BIC	#177476,VAL
126	015454	012737	000000	002530		MOV	#0,EXP
127	015462	023737	002530	002526		CMP	EXP,VAL
128	015470	001406				BEQ	404
129							
130					:	ERROR CONDITION 2	
131					:		
132	015472	012701	000001			MOV	#1,R1
133	015476					ERRHRD	2,MSG3,ERR4
	015476	104456				TRAP	C#ERRHRD
	015500	000002				.WORD	2
	015502	004632				.WORD	MSG3
	015504	011776				.WORD	ERR4
134	015506				404:		
135	015506					ENDSUB	
	015506	104403			L10020:		
	015506					TRAP	C#ESUB
136	015510					BGNSUB	
	015510				T1.3:		
	015510	104402				TRAP	C#BSUB
137	015512					INLOOP	
	015512	104420				TRAP	C#INLP
138	015514					BNCOMPLETE	414
	015514	103002				BCC	414
139	015516	004537	016002			JSR	R5,BRESET
140	015522				414:		
141					:		
142					:	TEST QINT REGISTER	
143					:		
144	015522	017737	164552	002526		MOV	QINT,VAL
145	015530	012737	001000	002530		MOV	#1000,EXP
146	015536	023737	002530	002526		CMP	EXP,VAL
147	015544	001406				BEQ	504
148							
149					:	ERROR CONDITION 3	
150					:		
151	015546	012701	000002			MOV	#2,R1
152	015552					ERRHRD	3,MSG3,ERR4
	015552	104456				TRAP	C#ERRHRD
	015554	000003				.WORD	3
	015556	004632				.WORD	MSG3
	015560	011776				.WORD	ERR4
153	015562				504:		
154	015562					ENDSUB	
	015562	104403			L10021:		
	015562					TRAP	C#ESUB
155	015564					BGNSUB	
	015564				T1.4:		
	015564	104402				TRAP	C#BSUB
156	015566					INLOOP	
	015566	104420				TRAP	C#INLP
157	015570					BNCOMPLETE	514

: READ QVNT REGISTER
: MASK OUT UNDETERMINED BITS
: ALL ZEROS TO EXPECTED
: COMPARE EXPECTED WITH VALUE
: BRANCH IF OK

: LOAD R1 WITH REGISTER NUMBER
: UNEXPECTED RESULT, TELL OPERATOR

: CHECK FOR ERROR LOOP
: BRANCH IF NOT ERROR LOOP
: IF ERROR LOOP, BRANCH

: READ QINT REGISTER
: EXPECTED VALUE TO EXP
: COMPARE EXPECTED WITH VALUE
: BRANCH IF OK

: LOAD R1 WITH REGISTER NUMBER
: UNEXPECTED RESULT, TELL OPERATOR

: CHECK FOR ERROR LOOP
: BRANCH IF NOT ERROR LOOP

Address	Hex	Hex	Hex	Hex	Assembly	Comment
158	015570	103002			BCC	51\$
159	015572	004537	016002		JSR	R5,BRESET
160	015576				51\$:	; IF ERROR LOOP, BRANCH
161					:	
162					; TEST QCTL REGISTER	
163	015576	017737	164500	002526	MOV	QCTL,VAL
164	015604	012737	000000	002530	MOV	#0,EXP
165	015612	023737	002530	002526	CMP	EXP,VAL
166	015620	001406			BEQ	60\$
167					:	
168					; ERROR CONDITION 4	
169					:	
170	015622	012701	000003		MOV	#3,R1
171	015626				ERRHRD	4,MSG3,ERR4
	015626	104456			TRAP	C\$ERRHRD
	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	
	015642	104420			TRAP	C\$INLP
176	015644				BNCOMPLETE	61\$
	015644	103002			BCC	61\$
177	015646	004537	016002		JSR	R5,BRESET
178	015652				61\$:	
179					:	
180					; TEST QDMA REGISTER	
181					:	
182	015652	017737	164426	002526	MOV	QDMA,VAL
183	015660	012737	000200	002530	MOV	#200,EXP
184	015666	023737	002530	002526	CMP	EXP,VAL
185	015674	001406			BEQ	70\$
186					:	
187					; ERROR CONDITION 5	
188					:	
189	015676	012701	000004		MOV	#4,R1
190	015702				ERRHRD	5,MSG3,ERR4
	015702	104456			TRAP	C\$ERRHRD
	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					; QWC REGISTER NOT CHECKED	
197					:	

TEST1: RESET

```

198
199
200      ; QMT0 REGISTER NOT CHECKED (TRI-STATE)
201      ;
202
203      015714      BGNSUB
204      015714      104402      T1.6:      TRAP      C#BSUB
205      015716      104420      INLOOP      ; CHECK FOR ERROR LOOP
206      015720      103002      TRAP      C#INLP      ; BRANCH IF NOT ERROR LOOP
207      015722      004537      016002      BNCOMPLETE      71#      ; IF ERROR LOOP, BRANCH
208      015726      BCC      71#
209      015726      JSR      R5,BRESET
210      71# :
211      ; TEST QMT1 REGISTER
212      ;
213      015726      017737      164364      002526      MOV      @QMT1,VAL      ; READ QMT1 REGISTER
214      015734      012737      000000      002530      MOV      #0,EXP      ; ALL ZEROS TO EXPECTED
215      015742      023737      002530      002526      CMP      EXP,VAL      ; COMPARE EXPECTED WITH VALUE
216      015750      001406      BEQ      80#      ; BRANCH IF OK
217      ;
218      ; ERROR CONDITION 6
219      ;
220      015752      012701      000011      MOV      #9,R1      ; LOAD R1 WITH REGISTER NUMBER
221      015756      104456      ERRHRD      6,MSG3,ERR4      ; UNEXPECTED RESULT, TELL OPERATOR
222      015760      000006      TRAP      C#ERRHRD
223      015762      004632      .WORD      6
224      015764      011776      .WORD      MSG3
225      015766      .WORD      ERR4
226      015766      104436      012700      000004      CLRVEC      #4      ; CLEAR BAD ADDRESS TRAP
227      015772      104432      MOV      #4,R0
228      015774      104403      TRAP      C#CVEC
229      015774      104403      ENDSUB
230      L10024:      TRAP      C#ESUB
231      ;
232      ; QVAD AND QVDA REGISTERS ARE NOT CHECKED
233      ;
234      015776      104432      EXIT TST
235      016000      000020      TRAP      C#EXIT
236      016000      000020      .WORD      L10016-.
237      ;
238      ; SUBROUTINE TO DO A BUS RESET
239      ; CALLING ROUTINE-
240      ; JSR R5,BRESET
241      ;
242      ; BRESET:
243      016002      012701      003000      10# :      MOV      #3000,R1      ; GET READY FOR DELAY LOOP
244      016006      104422      BREAK      ; ALLOW CONTROL C
245      016010      005301      TRAP      C#BRK
246      016012      001375      DEC      R1      ; DECREMENT DELAY COUNTER
247      016014      000005      BNE      10#      ; LOOP IF NOT DONE
248      016016      000205      RESET      ; DO BUS RESET
249      016016      000205      RTS      R5      ; RETURN FROM SUBROUTINE

```


L5

SEQ 0063

TEST1: RESET

240
241 016020
016020
016020 104401

ENDTST
L10016: TRAP C#ETST

TEST2: CHECK QMT1 REGISTER

```

243 .SBTTL TEST2: CHECK QMT1 REGISTER
244 016022 STATST <CHECK QMT1 REGISTER>
016022 040 103 110 T2MSG:: .ASCIZ / CHECK QMT1 REGISTER/

245 ;*****
246 ;**
247 ;TEST DESCRIPTION-
248 ; CHECKS R/W BITS
249 ;TEST STEPS-
250 ; LOAD 1'S IN QMT1
251 ; READ AND CHECK
252 ; LOAD 0'S
253 ; READ AND CHECK
254 ;
255 ;
256 ;--
257 ;*****
258
259 016050 BGNTST
016050 T2::
260 016050 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
016050 104421 TRAP C#RFLA
016052 022700 001000 CMP #PNT,RO ; 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,RO
016076 104417 TRAP C#PNTF
016100 062706 000006 ADD #6,SP
016104 005037 002532 64$: CLR ERROR1 ; CLEAR ERROR COUNTER
261 016110 BGNSUB
016110 T2.1:
016110 104402 TRAP C#BSUB
262 ;
263 ; FIRST CHECK WITH ONE'S
264 ;
265 016112 012737 000236 002530 MOV #236,EXP ; MOVE ALL 1'S TO EXPECTED
266 016120 013777 002530 164170 MOV EXP,BQMT1 ; MOVE EXPECTED TO QMT1
267 016126 017737 164164 002526 MOV BQMT1,VAL ; MOVE RESULT TO VALUE STORAGE
268 016134 042737 177541 002526 BIC #C236,VAL ; CLEAR UNDETERMINED BITS
269 016142 023737 002526 002530 CMP VAL,EXP ; IS THAT WHAT IS REALLY THERE?
270 016150 001406 BEQ 1$ ; BRANCH IF OK
271 ;
272 ; ERROR CONDITION 1
273 ;
274 016152 012701 000011 MOV #9,R1 ; REGISTER NUMBER FOR MESSAGE
275 016156 ERRHRD 1,MSG4,ERR4 ; GO REPORT ERROR TO OPERATOR
016156 104456 TRAP C#ERRRD
016160 000001 .WORD 1
016162 004672 .WORD MSG4
016164 011776 .WORD ERR4
276 016166 1$:
277 016166 ENDSUB
016166 L10026:
016166 104403 TRAP C#ESUB
278 016170 BGNSUB
016170 T2.2:

```

TEST2: CHECK QMT1 REGISTER

```

016170 104402
279
280
281
282 016172 012737 000000 002530
283 016200 013777 002530 164110
284 016206 017737 164104 002526
285 016214 042737 177541 002526
286 016222 023737 002526 002530
287 016230 001411
288
289
290
291 016232 012701 000011
292 016236 012737 000000 002530
293 016244
    016244 104456
    016246 000002
    016250 004731
    016252 011776
294 016254
295 016254
    016254
    016254 104403
296 016256
    016256
    016256 104401
297

TRAP C#BSUB
;
; NOW CHECK WITH ZERO'S
;
MOV #0,EXP
MOV EXP,QMT1
MOV QMT1,VAL
BIC #C236,VAL
CMP VAL,EXP
BEQ 2#

; ERROR CONDITION 2
;
MOV #9,R1
MOV #0,EXP
ERRHRD 2,MSG5,ERR4
TRAP C#ERRRD
.WORD 2
.WORD MSG5
.WORD ERR4
2#:
ENDSUB
L10027: TRAP C#ESUB
ENDTST
L10025: TRAP C#ETST

; MOVE ALL 0'S TO EXPECTED
; MOVE EXPECTED TO QMT1
; MOVE RESULT TO VALUE STORAGE
; CLEAR UNDETERMINED BITS
; IS THAT WHAT IS REALLY THERE?
; BRANCH IF OK

; REGISTER # FOR MESSAGE
; EXPECTED RESULT FOR MESSAGE
; GO REPORT ERROR TO OPERATOR

```


TEST3: CHECK QCLR RESET

```

299 .SBTTL TEST3: CHECK QCLR RESET
300 016260 STATST <CHECK QCLR RESET>
016260 040 103 110 T3MSG:: .ASCIZ / CHECK QCLR RESET/

301 ;*****
302 ;**
303 ;TEST DESCRIPTION-
304 ; CHECKS QCLR REGISTER BIT RESET
305 ; USES QHALT - QCLR LOOP-BACK
306 ;TEST STEPS-
307 ; LOAD ALL R/W BITS THAT ARE CLEARABLE (EXCEPT THREN8)
308 ; SET QMT1 QMHLT BIT
309 ; READ ALL REGISTERS THAT WERE LOADED
310 ; CHECK FOR CLEAR (SEE EXPECTED VALUES IN TABLE EXP3)
311 ;
312 ;--
313 ;*****
314
315
316 016302 BGNTST
016302 T3::
317 016302 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
016302 104421 TRAP C1RFLA
016304 022700 001000 CMP @PNT,RO ; IS PNT FLAG SET?
016310 001012 BNE 641 ; 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,RO
016330 104417 TRAP C1PNTF
016332 062706 000006 ADD @6,SP
016336 005037 002532 641: CLR ERROR1 ; CLEAR ERROR COUNTER
016342 005077 163732 CLR @QINT ; CLEAR UPWFL BIT FROM PREVIOUS ; V C.0
; TEST TO PREVENT UNEXPECTED ; V C.0
; INTERRUPT. ; V C.0

318
319
320 ;
321 ; LOAD ALL R/W BITS THAT ARE CLEARABLE
322 ;
323
324 016346 012777 177401 163720 MOV @177401,@QCMD ; SET ALL R/W BITS IN QCMD(NO IE);V C.0
325 016354 012777 177401 163714 MOV @177401,@QVNT ; SET ALL R/W BITS IN QVNT(NO IE);V C.0
326 016362 012777 000014 163712 MOV @000014,@QCTL ; SET ALL R/W BITS IN QCTL
327 016370 012777 037562 163706 MOV @037562,@QDMA ; SET ALL R/W BITS IN QDMA
328 016376 012777 000100 163674 MOV @000100,@QINT ; SET ALL R/W BITS IN QINT
329 016404 052777 000100 163662 BIS @PRIE,@QCMD ; NOW ENABLE INTERRUPTS ; V C.0
330 016412 052777 000100 163656 BIS @ACKIE,@QVNT ; V C.0
331 016420 012777 000236 163670 MOV @000236,@QMT1 ; V C.0
332 016426 052777 000100 163662 BIS @QHALT,@QMT1 ; RESET REGISTERS

333 ;
334 ; TEST QCMD REGISTER
335 ;
336 016434 017737 163634 002526 MOV @QCMD,VAL ; GET VALUE IN VAL
337 016442 042737 177477 002526 BIC @C300,VAL ; CLEAR UNTESTED BITS
338 016450 012737 000000 002530 MOV @0,EXP ; EXPECTED IN EXP
339 016456 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
340 016464 001406 BEQ 10 ; BRANCH IF OK

341 ;
342 ; ERROR CONDITION 1
343 ;

```

TEST3: CHECK QCLR RESET

344	016466	012701	000000	MOV	#0,R1	; LOAD R1 WITH REGISTER NUMBER
345	016472			ERRHRD	1,MSG6,ERR4	; REPORT ERROR TO OPERATOR
	016472	104456		TRAP	C1ERRHRD	
	016474	000001		.WORD	1	
	016476	004771		.WORD	MSG6	
	016500	011776		.WORD	ERR4	
346	016502			18:		
347				; TEST QVNT REGISTER		
348						
349						
350	016502	017737	163570	MOV	QVNT,VAL	; GET VALUE IN VAL
351	016510	042737	177477	BIC	#1C300,VAL	; CLEAR UNTESTED BITS
352	016516	012737	000000	MOV	#0,EXP	; EXPECTED IN EXP
353	016524	023737	002530	CMF	EXP,VAL	; GET EXPECTED IN VAL
354	016532	001406	002526	BEQ	28	; IF OK, BRANCH
355						
356				; ERROR CONDITION 2		
357						
358	016534	012701	000001	MOV	#1,R1	; LOAD R1 WITH REGISTER NUMBER
359	016540			ERRHRD	2,MSG6,ERR4	; REPORT ERROR TO OPERATOR
	016540	104456		TRAP	C1ERRHRD	
	016542	000002		.WORD	2	
	016544	004771		.WORD	MSG6	
	016546	011776		.WORD	ERR4	
360	016550			28:		
361				; TEST QINT REGISTER		
362						
363						
364	016550	017737	163524	MOV	QINT,VAL	; GET VALUE IN VAL
365	016556	012737	001000	MOV	#1000,EXP	; EXPECTED VALUE TO EXP
366	016564	023737	002530	CMF	EXP,VAL	; COMPARE EXPECTED WITH VALUE
367	016572	001406	002526	BEQ	38	; IF OK, BRANCH
368						
369				; ERROR CONDITION 3		
370						
371	016574	012701	000002	MOV	#2,R1	; LOAD R1 WITH REGISTER NUMBER
372	016600			ERRHRD	3,MSG6,ERR4	; REPORT ERROR TO OPERATOR
	016600	104456		TRAP	C1ERRHRD	
	016602	000003		.WORD	3	
	016604	004771		.WORD	MSG6	
	016606	011776		.WORD	ERR4	
373	016610			38:		
374				; TEST QCTL REGISTER		
375						
376						
377	016610	017737	163466	MOV	QCTL,VAL	; GET VALUE IN VAL
378	016616	012737	000000	MOV	#0,EXP	; ALL ZEROS TO EXPECTED
379	016624	023737	002530	CMF	EXP,VAL	; COMPARE EXPECTED WITH VALUE
380	016632	001406	002526	BEQ	48	; IF OK, BRANCH
381						
382				; ERROR CONDITION 4		
383						
384	016634	012701	000003	MOV	#3,R1	; LOAD R1 WITH REGISTER NUMBER
385	016640			ERRHRD	4,MSG6,ERR4	; REPORT ERROR TO OPERATOR
	016640	104456		TRAP	C1ERRHRD	
	016642	000004		.WORD	4	
	016644	004771		.WORD	MSG6	

TEST3: CHECK QCLR RESET

```

016646 011776
386 016650
387
388
389
390 016650 017737 163430 002526
391 016656 012737 000200 002530
392 016664 023737 002530 002526
393 016672 001406
394
395
396
397 016674 012701 000004
398 016700
016700 104456
016702 000005
016704 004771
016706 011776
399 016710
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415 016710 017737 163402 002526
416 016716 012737 000000 002530
417 016724 023737 002530 002526
418 016732 001406
419
420
421
422 016734 012701 000011
423 016740
016740 104456
016742 000006
016744 004771
016746 011776
424 016750
425
426
427
428
429
430 016750
016750
016750 104401

.WORD ERR4
44:
;
; TEST QDMA REGISTER
;
MOV BQDMA,VAL ; GET VALUE IN VAL
MOV #200,EXP ; EXPECTED VALUE TO EXP
CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
BEQ 54 ; IF OK, BRANCH
;
; ERROR CONDITION 5
;
MOV #4,R1 ; LOAD R1 WITH REGISTER NUMBER
ERRHRD 5,MSG6,ERR4 ; REPORT ERROR TO OPERATOR
TRAP C4ERRHD
.WORD 5
.WORD MSG6
.WORD ERR4
54:
;
; QUBA REGISTER NOT TESTED
;
;QBA REGISTER NOT TESTED
;
; QWC REGISTER NOT TESTED
;
;QMT0 REGISTER NOT TESTED
;
; TEST QMT1 REGISTER
;
MOV BQMT1,VAL ; GET VALUE IN VAL
MOV #0,EXP ; ALL ZEROS TO EXPECTED
CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
BEQ 64 ; IF OK, BRANCH
;
; ERROR CONDITION 6
;
MOV #9,R1 ; LOAD R1 WITH REGISTER NUMBER
ERRHRD 6,MSG6,ERR4 ; REPORT ERROR TO OPERATOR
TRAP C4ERRHD
.WORD 6
.WORD MSG6
.WORD ERR4
64:
;
; QVAD AND QVDA REGISTERS ARE NOT CHECKED
;
ENDTST
L10030:
TRAP C4ETST

```


TEST4: CHECK QCMD LO BITS (MOVING 1)

```

432 .SBTTL TEST4: CHECK QCMD LO BITS (MOVING 1)
433 016752 STATST <CHECK QCMD LO BITS (MOVING 1)>
016752 040 103 110 T4MSG:: .ASCIZ / CHECK QCMD LO BITS (MOVING 1)/
434
435 ;*****
436 ;**
437 ;TEST DESCRIPTION-
438 ; CHECKS THAT BITS 1-5 CAN BE LOADED
439 ; USES LDU CMD - LDQEVENT LOOP-BACK
440 ;TEST STEPS-
441 ; SET QMT1 FUNC = 2 (SELQCMD)
442 ; LOAD MOVING BIT WORD IN QMT0
443 ; READ QCMD, CHECK BITS 1-5
444 ; REPEAT FOR ALL 5 BITS
445 ;
446 ;--
447 ;*****
448
449 017012 BGNTST
017012
450 017012 T4::
017012 104421 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
017014 022700 001000 TRAP C0RFLA
017020 001012 CMP #PNT,R0 ; IS PNT FLAG SET?
017022 012746 016752 BNE 644 ; BRANCH IF NOT SET
017026 012746 002270 MOV #T4MSG,-(SP)
017032 012746 000002 MOV #PNTMSG,-(SP)
017036 010600 MOV #2,-(SP)
017040 104417 MOV SP,R0
017042 062706 000006 TRAP C0PNTF
017046 005037 002532 ADD #6,SP
451 017032 644: CLR ERROR1 ; CLEAR ERROR COUNTER
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
452 017076 005037 002532 CLR ERROR1
453 017102 012701 000000 MOV #0,R1 ; REGISTER NUMBER TO R1
454 017106 012702 000005 MOV #5,R2 ; NUMBER OF BITS TO CHECK
455 017112 012737 000002 002530 MOV #BIT1,EXP ; FIRST DATA PATTERN TO CHECK
456 017120 052777 000010 163170 BIS #SELQCMD,QMT1 ; SELECT QCMD MODE IN QMT1
457
458 ; TEST BITS 1-5 WITH MOVING ONE
459 ;
460 017126 BGNSUB
017126
461 017126 104402 T4.1:
017130 104404 104:
017132 013777 002530 163154 TRAP C0BSUB
017140 017737 163130 002526 BGNSEG
017146 042737 177701 002526 TRAP C0BSEG
462 017132 013777 002530 163154 MOV EXP,QMT0 ; MOVE EXPECTED TO REGISTER QMT0
463 017140 017737 163130 002526 MOV QCMD,VAL ; READ QCMD AND MOVE TO VALUE
464 017146 042737 177701 002526 BIC #177701,VAL ; MASK OUT UNWANTED BITS
465 017154 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
466 017162 001414 BEQ 204 ; IF OK, BRANCH
467
468 ; 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:ESUB				
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			BEG	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)

```

490 .SBTTL TEST5: CHECK QCMD LOW BITS (MOVING 0)
491 017260 STATST <CHECK QCMD LOW BITS (MOVING 0)>
    017260 040 103 110 T5MSG:: .ASCIZ / CHECK QCMD LOW BITS (MOVING 0)/
492 ;*****
493 ;**
494 ;TEST DESCRIPTION-
495 ; CHECKS THAT BITS 1-5 CAN BE LOADED
496 ; USES LDUCMD - LDQEVENT LOOP-BACK
497 ;TEST STEPS-
498 ; SET QMT1 FUNC = 2 (SELQCMD)
499 ; LOAD MOVING BIT WORD IN QMT0
500 ; READ QCMD, CHECK BITS 1-5
501 ; REPEAT FOR ALL 5 BITS
502 ;
503 ;
504 ;--
505 ;*****
506
507 017320 BGNTST
    017320 T5::
508 017320 CKPNT
    017320 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
    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
509 017360 INIT
    017360 012777 000100 162730 ; INITIALIZE DEVICE
    017366 013777 002520 162722 MOV #MHALT,BQMT1
    017374 005077 162700 MOV INITWORD,BQMT1
    017400 005037 002532 CLR BQINT ; INITIALIZE Q-BRIDGE
    017404 012701 000000 CLR ERROR1 ; SET UP LOOPBACK OPTION
    017410 012702 000005 MOV #0,R1 ; CLEAR QINT AFTER LOOPBACK ENABLED
    017414 012703 000002 MOV #5,R2 ; CLEAR ERROR FLAG
    017420 010337 002530 MOV #BIT1,R3 ; REGISTER NUMBER TO R1
    017424 005137 002530 MOV R3,EXP ; NUMBER OF BITS TO CHECK
    017430 052777 000010 162660 COM EXP ; INITIAL DATA PATTERN TO R3
    017436 017436 BGNSUB ; MOVE BIT PATTERN TO EXPECTED
    017436 104402 T5.1: TRAP C#BSUB ; COMPLEMENT FOR MOVING 0
    017440 104404 10# : TRAP C#BSEG ; SELECT QCMD MODE IN QMT1
    017442 042737 177701 002530 BIC #177701,EXP ; INITIALIZE Q-BRIDGE
    017450 013777 002530 162636 MOV EXP,BQMT0 ; SET UP LOOPBACK OPTION
    017456 017737 162612 002526 MOV BQCMD,VAL ; CLEAR QINT AFTER LOOPBACK ENABLED
    017464 042737 177701 002526 BIC #177701,VAL ; CLEAR ERROR FLAG
    017472 023737 002530 002526 CMP EXP,VAL ; REGISTER NUMBER TO R1
    017500 001414 BEQ 20# ; NUMBER OF BITS TO CHECK
    ; INITIAL DATA PATTERN TO R3
    ; MOVE BIT PATTERN TO EXPECTED
    ; COMPLEMENT FOR MOVING 0
    ; SELECT QCMD MODE IN QMT1
    ; TEST BITS 1-5 WITH MOVING ZERO
    ;
    ; BGNSUB
    T5.1:
    10# :
    TRAP C#BSUB
    BGNSEG
    TRAP C#BSEG
    BIC #177701,EXP
    MOV EXP,BQMT0
    MOV BQCMD,VAL
    BIC #177701,VAL
    CMP EXP,VAL
    BEQ 20#
    ; CLEAR UNTESTED BITS
    ; MOVE EXPECTED TO REGISTER QMT0
    ; READ QCMD AND MOVE TO VALUE
    ; MASK OUT UNDETERMINED BITS
    ; COMPARE EXPECTED WITH VALUE
    ; IF OK, BRANCH

```


TEST5: CHECK QCMD LOW BITS (MOVING 0)

```

527
528
529
530 017502 005737 002532
531 017506 001005
532 017510
    017510 104456
    017512 000001
    017514 005054
    017516 012246
533 017520 000402
534 017522 004737 012366
535 017526 005237 002532
536
537
538
539 017532
    017532
    017532 104405
540 017534 006303
541 017536 010337 002530
542 017542 005137 002530
543 017546 005302
544 017550 001333
545 017552
    017552
    017552 104403
546 017554 005737 002532
547 017560 001410
548 017562
    017562 012746 003333
    017566 012746 000001
    017572 010600
    017574 104415
    017576 062706 000004
549 017602
550 017602
    017602
    017602 104401

; ERROR CONDITION
;
; TST ERROR1
; BNE 150
; ERRMRD 1,MSG8,ERR7
; TRAP C#ERRMRD
; .WORD 1
; .WORD MSG8
; .WORD ERR7
; BR 180
150: CALL DETAIL
180: INC ERROR1
;
; END ERROR CONDITION
;
200: ENDSEG
100000:
; TRAP C#ESEG
; ASL R3
; MOV R3,EXP
; COM EXP
; DEC R2
; BNE 100
; ENDSUB
L10034:
; TRAP C#ESUB
; TST ERROR1
; BEQ 5000
; PRINTX #FORO
; MOV #FORO,-(SP)
; MOV #1,-(SP)
; MOV SP,R0
; TRAP C#PNTX
; ADD #4,SP
;
; ANY ERRORS
; BRANCH IF NO ERRORS
; LINE FEED
5000:
L10033:
; TRAP C#ETST

```

TEST6: CHECK QCMD HI BITS (MOVING 1)

```

552 .SBTTL TEST6: CHECK QCMD HI BITS (MOVING 1)
553 017604 STATST <CHECK QCMD HI BITS (MOVING 1)>
    017604 040 103 110 T6MSG:: .ASCIZ / CHECK QCMD HI BITS (MOVING 1)/
554
555 ;*****
556 ;**
557 ;TEST DESCRIPTION-
558 ; CHECKS THAT BITS 8-15 CAN BE LOADED
559 ;TEST STEPS-
560 ; LOAD MOVING BIT WORD IN QCMD
561 ; READ QCMD, CHECK BITS 8-15
562 ; REPEAT FOR ALL 8 BITS
563 ;
564 ;-
565 ;*****
566
567 017644 BGNTST
    017644
568 017644 T6::
    017644 CKPNT
    017644 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
    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
569 017704 INIT
    017704 ;INITIALIZE DEVICE
    017712 012777 000100 162404 MOV #MHALT,BQMT1 ; INITIALIZE Q-BRIDGE
    017712 013777 002520 162376 MOV INITWORD,BQMT1 ; SET UP LOOPBACK OPTION
    017720 005077 162354 CLR BQINT ; CLEAR QINT AFTER LOOPBACK ENABLED
    017724 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
570 017730 012701 000000 MOV #0,R1 ; REGISTER NUMBER TO R1
571 017734 012702 000010 MOV #8,R2 ; NUMBER OF BITS TO CHECK
572 017740 012737 000400 002530 MOV #BIT1,EXP ; FIRST DATA PATTERN TO CHECK
573
574 ; TEST BITS 8-15 WITH MOVING ONE
575 ;
576 017746 BGNSUB
    017746
    017746 104402 T6.1:
    017750 104404 10# TRAP C#BSUB
    017752 013777 002530 162314 BGNSUB
    017752 017737 162310 002526 TRAP C#BSEG
    017760 042737 000377 002526 MOV EXP,BQCMD ; MOVE EXPECTED TO REGISTER QCMD
    017766 023737 002530 002526 MOV BQCMD,VAL ; READ QCMD AND MOVE TO VALUE
    017774 023737 002530 002526 BIC #377,VAL ; MASK OUT UNTESTED BITS
    020002 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
    020002 001414 BEQ 20# ; IF OK, BRANCH
583
584 ; ERROR CONDITION
585 ;
586 020004 005737 002532 TST ERROR1 ; SEE IF FIRST ERROR
587 020010 001005 BNE 15# ; SKIP HEADER IF ALREADY PRINTED
588 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	: PRINT DETAIL LINE
591 020030	005237	002532	18#: INC	ERROR1	: INCREMENT ERROR COUNT
592			:		
593			: END ERROR CONDITION		
594			:		
595 020034			20#: ENDSEG		
020034			10000#: TRAP	C#ESEG	
020034	104405		ASL	EXP	: SHIFT BIT FOR NEXT TEST
596 020036	006337	002530	DEC	R2	: DECREMENT LOOP COUNTER
597 020042	005302		BNE	10#	: IF NOT DONE, BRANCH BACK
598 020044	001341		ENDSUB		
599 020046			L10036: TRAP	C#ESUB	
020046			TST	ERROR1	: ANY ERRORS
600 020050	005737	002532	BEQ	500#	: BRANCH IF NO ERRORS
601 020054	001410		PRINTX	#FORO	: LINE FEED
602 020056			MOV	#FORO, -(SP)	
020056	012746	003333	MOV	#1, -(SP)	
020062	012746	000001	MOV	SP, R0	
020066	010600		TRAP	C#PNTX	
020070	104415		ADD	#4, SP	
020072	062706	000004	500#: ENDTST		
603 020076			L10035: TRAP	C#ETST	
604 020076					
020076	104401				

TEST7: CHECK QCMD HI BITS (MOVING 0)

```

606 .SBTTL TEST7: CHECK QCMD HI BITS (MOVING 0)
607 020100 STATST <CHECK QCMD HI BITS (MOVING 0)>
      020100 040 103 110 T7MSG:: .ASCIZ / CHECK QCMD HI BITS (MOVING 0)/
608
609 ;*****
610 ;**
611 ;TEST DESCRIPTION-
612 ; CHECKS THAT BITS 8-15 CAN BE LOADED
613 ;TEST STEPS-
614 ; LOAD MOVING BIT WORD IN QCMD
615 ; READ QCMD, CHECK BITS 8-15
616 ; REPEAT FOR ALL 8 BITS
617 ;
618 ;--
619 ;*****
620
621 020140 BGNTST
      020140
622 020140 T7::
      020140 CKPNT
      020140 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
      020142 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
      020146 001012 BNE 644 ; 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 644: CLR ERROR1 ; CLEAR ERROR COUNTER
623 020200 INIT
      020200 012777 000100 162110 ;INITIALIZE DEVICE
      020206 013777 002520 162102 MOV #MHALT,BQMT1
      020214 005077 162060 MOV INITWORD,BQMT1
      020220 005037 002532 CLR BQINT ; INITIALIZE Q-BRIDGE
      020224 012701 000000 CLR ERROR1 ; SET UP LOOPBACK OPTION
      020230 012702 000010 MOV #0,R1 ; CLEAR QINT AFTER LOOPBACK ENABLED
      020234 012703 000400 MOV #8,R2 ; CLEAR ERROR FLAG
      020240 010337 002530 MOV #BIT8,R3 ; REGISTER NUMBER TO R1
      020244 005137 002530 MOV R3,EXP ; NUMBER OF BITS TO CHECK
      020244 005137 002530 COM EXP ; INITIAL DATA PATTERN TO R3
      020244 005137 002530 ; MOVE BIT PATTERN TO EXPECTED
      020244 005137 002530 ; COMPLEMENT FOR MOVING 0
629
630 ; TEST BITS 8-15 WITH MOVING ZERO
631 ;
632 020250 BGNSUB
      020250
      020250 104402 T7.1:
633 020252 104404 104:
      020252 104404 TRAP C#BSUB
      020254 042737 000377 002530 BGNSEG
      020262 013777 002530 162004 TRAP C#BSEG
      020270 017737 162000 002526 BIC #377,EXP
      020276 042737 000377 002526 MOV EXP,BQCMD ; MASK OUT UNTESTED BITS
      020304 023737 002530 002526 MOV BQCMD,VAL ; MOVE EXPECTED TO REGISTER QCMD
      020312 001414 BIC #377,VAL ; READ QCMD AND MOVE TO VALUE
      020312 001414 CMP EXP,VAL ; MASK OUT UNTESTED BITS
      020312 001414 BEQ 204 ; COMPARE EXPECTED WITH VALUE
      020312 001414 ; IF OK, BRANCH
640
641 ; ERROR CONDITION
642 ;

```

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)

```

665 .SBTTL TEST8: CHECK QVNT LO BITS (MOVING 1)
666 020416 STATST <CHECK QVNT LO BITS (MOVING 1)>
667 020416 040 103 110 T8MSG:: .ASCIZ / CHECK QVNT LO BITS (MOVING 1)/
668 ;*****
669 ;**
670 ;TEST DESCRIPTION-
671 ; CHECKS THAT BITS 1-5 CAN BE LOADED
672 ; USES LDUCMD - LDQEVENT LOOP-BACK
673 ;TEST STEPS-
674 ; SET QMT1 FUNC = 1 (SELQVNT)
675 ; LOAD MOVING BIT WORD IN QMT0
676 ; READ QVNT, CHECK BITS 1-5
677 ; REPEAT FOR ALL 5 BITS
678 ;
679 ;--
680 ;*****
681
682 020456 BGNTST
683 020456 T8::
684 020456 CKPNT
685 020456 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
686 020460 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
687 020464 001012 BNE 644 ; BRANCH IF NOT SET
688 020466 012746 020416 MOV #T8MSG,-(SP)
689 020472 012746 002270 MOV #PNTMSG,-(SP)
690 020476 012746 000002 MOV #2,-(SP)
691 020502 010600 MOV SP,R0
692 020504 104417 TRAP C#PNTF
693 020506 062706 000006 ADD #6,SP
694 020512 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
695 020516 INIT
696 020516 ;INITIALIZE DEVICE
697 020516 MOV #MQHALT,QMT1 ;INITIALIZE Q-BRIDGE
698 020524 013777 002520 161572 MOV INITWORD,QMT1 ;SET UP LOOPBACK OPTION
699 020532 005077 161542 CLR QINT ;CLEAR QINT AFTER LOOPBACK ENABLED
700 020536 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
701 020542 012701 000001 MOV #1,R1 ; REGISTER NUMBER TO R1
702 020546 012702 000005 MOV #5,R2 ; NUMBER OF BITS TO CHECK
703 020552 012737 000002 002530 MOV #BIT1,EXP ; FIRST DATA PATTERN TO CHECK
704 020560 052777 000004 161530 BIS #SELQEVENT,QMT1 ; SELECT QVNT MODE IN QMT1
705 ;
706 ; TEST BITS 1-5 WITH MOVING ONE
707 ;
708 BGNSUB
709 T8.1:
710 104: TRAP C#BSUB
711 104: BGNSEG
712 104: TRAP C#BSEG
713 104: MOV EXP,QMT0 ; MOVE EXPECTED TO REGISTER QMT0
714 104: MOV QVNT,VAL ; READ QVNT AND MOVE TO VALUE
715 104: BIC #177701,VAL ; MASK OUT UNTESTED BITS
716 104: CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
717 104: BEQ 204 ; IF OK, BRANCH
718 ;
719 ; ERROR CONDITION
720 ;
721 ;

```


TEST8: CHECK QVNT LO BITS (MOVING 1)

702	020624	005737	002532	TST	ERROR1		
703	020630	001005		BNE	15:		: SEE IF FIRST ERROR
704	020632			ERRHRD	1,MSG11,ERR7		: SKIP HEADER IF ALREADY PRINTED
	020632	104456		TRAP	C#ERRHRD		: TELL OPERATOR ERROR
	020634	000001		.WORD	1		
	020636	005224		.WORD	MSG11		
	020640	012246		.WORD	ERR7		
705	020642	000402		BR	18:		: SKIP DETAIL LINE IF HEADER PRINTED
706	020644	004737	012366	CALL	DETAIL		: PRINT DETAIL LINE
707	020650	005237	002532	INC	ERROR1		: INCREMENT ERROR COUNT
708							
709							
710							
711	020654			20:	ENDSEG		
	020654			100004:			
	020654	104405		TRAP	C#ESEG		
712	020656	006337	002530	ASL	EXP		: SHIFT BIT FOR NEXT TEST
713	020662	005302		DEC	R2		: DECREMENT LOOP COUNTER
714	020664	001341		BNE	10:		: IF NOT DONE, BRANCH BACK
715	020666			ENDSUB			
	020666	104403		L10042:			
716	020670	005737	002532	TRAP	C#ESUB		
717	020674	001410		TST	ERROR1		: ANY ERRORS
718	020676			BEG	500:		: BRANCH IF NO ERRORS
	020676	012746	003333	PRINTX	#FORO		: LINE FEED
	020702	012746	000001	MOV	#FORO, -(SP)		
	020706	010600		MOV	#1, -(SP)		
	020710	104415		MOV	SP, R0		
	020712	062706	000004	TRAP	C#PNTX		
719	020716			ADD	#4, SP		
720	020716			500:			
	020716	104401		ENDTST			
				L10041:			
				TRAP	C#ETST		

TEST9: CHECK QVNT LOW BITS (MOVING 0)

```

722 .SBTTL TEST9: CHECK QVNT LOW BITS (MOVING 0)
723 020720 STATST <CHECK QVNT LOW BITS (MOVING 0)>
724 020720 040 103 110 T9MSG:: .ASCIZ / CHECK QVNT LOW BITS (MOVING 0)/
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
740 020760 T9::
741 020760 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
742 020760 TRAP C1RFLA
743 020762 104421 001000 CMP @PNT,R0 ; IS PNT FLAG SET?
744 020766 001012 001000 BNE 641 ; BRANCH IF NOT SET
745 020770 012746 020720 MOV @T9MSG,-(SP)
746 020774 012746 002270 MOV @PNTMSG,-(SP)
747 021000 012746 000002 MOV @2,-(SP)
748 021004 010600 MOV SP,R0
749 021006 104417 TRAP C1PNTF
750 021010 062706 000006 ADD @6,SP
751 021014 005037 002532 641: CLR ERROR1 ; CLEAR ERROR COUNTER
752 021020 INIT
753 ;INITIALIZE DEVICE
754 021020 012777 000100 161270 MOV @MOMALT,BQMT1 ; INITIALIZE Q-BRIDGE
755 021026 013777 002520 161262 MOV INITWORD,BQMT1 ; SET UP LOOPBACK OPTION
756 021034 005077 161240 CLR BQINT ; CLEAR QINT AFTER LOOPBACK ENABLED
757 021040 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
758
759 021044 012701 000001 MOV @1,R1 ; REGISTER NUMBER TO R1
760
761 021050 012702 000005 MOV @5,R2 ; NUMBER OF BITS TO CHECK
762 021054 012703 000002 MOV @BIT1,R3 ; INITIAL DATA PATTERN TO R3
763 021060 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
764 021064 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
765 021070 052777 000004 161220 BIS @SELQEVENT,BQMT1 ; SELECT QVNT MODE IN QMT1
766
767 ; TEST BITS 1-5 WITH MOVING ZERO
768
769 021076 BGNSUB
770 021076 T9.1:
771 021076 104402 TRAP C1BSUB
772 021100 104404 104404 104: BGNSEG
773 021102 042737 177701 002530 TRAP C1BSEG
774 021110 013777 002530 161176 BIC @177701,EXP ; MASK OUT UNTESTED BITS
775 021116 017737 161154 002526 MOV EXP,BQMT0 ; MOVE EXPECTED TO REGISTER QMT0
776 021124 042737 177701 002526 MOV BQVNT,VAL ; READ QVNT AND MOVE TO VALUE
777 BIC @177701,VAL ; MASK OUT UNTESTED BITS

```

C7

SEQ 0080

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					:		
763					: ERROR CONDITION		
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:ERRHRD	
	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		CALL	DETAIL	: PRINT DETAIL LINE
769	021166	005237	002532		INC	ERROR1	: INCREMENT ERROR COUNT
770							
771					:		
772					: END ERROR CONDITION		
773	021172				20:	ENDSEG	
	021172				10000:		
	021172	104405			TRAP	C:SESEG	
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	#FOR0	: LINE FEED
	021222	012746	003333		MOV	#FOR0,-(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)

```

786 .SBTTL TEST10: CHECK QVNT HI BITS (MOVING 1)
787 021244 STATST <CHECK QVNT HI BITS (MOVING 1)>
021244 040 103 110 T10MSG:: .ASCIZ / CHECK QVNT HI BITS (MOVING 1)/
788
789 ;*****
790 ;**
791 ;TEST DESCRIPTION-
792 ; CHECKS THAT BITS 8-15 CAN BE LOADED
793 ;TEST STEPS-
794 ; LOAD MOVING BIT WORD IN QVNT
795 ; READ QVNT, CHECK BITS 8-15
796 ; REPEAT FOR ALL 8 BITS
797 ;
798 ;--
799 ;*****
800
801 021304 BGNTST
021304 T10::
802 021304 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
021304 104421 TRAP C#FLA
021306 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
021312 001012 BNE 644 ; 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 644: CLR ERROR1 ; CLEAR ERROR COUNTER
803 021344 INIT
;INITIALIZE DEVICE
021344 012777 000100 160744 MOV #MHALT,BQMT1 ;INITIALIZE Q-BRIDGE
021352 013777 002520 160736 MOV INITWORD,BQMT1 ;SET UP LOOPBACK OPTION
021360 005077 160714 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
021364 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
804 021370 012701 000001 MOV #1,R1 ; REGISTER NUMBER TO R1
805 021374 012702 000010 MOV #8,R2 ; NUMBER OF BITS TO CHECK
806 021400 012737 000400 002530 MOV #BIT8,EXP ; FIRST DATA PATTERN TO CHECK
807
808 ; TEST BITS 8-15 WITH MOVING ONE
809 ;
810 021406 BGNSUB
021406 T10.1:
021406 104402 104404 104404 TRAP C#BSUB
811 021410 104404 104404 BGNSEG
021410 104404 TRAP C#BSEG
812 021412 013777 002530 160656 MOV EXP,BQVNT ; MOVE EXPECTED TO REGISTER QVNT
813 021420 017737 160652 002526 MOV BQVNT,VAL ; READ QVNT AND MOVE TO VALUE
814 021426 042737 000377 002526 BIC #377,VAL ; MASK OUT UNTESTED BITS
815 021434 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
816 021442 001414 BEQ 204 ; IF OK, BRANCH
817
818 ; ERROR CONDITION
819 ;
820 021444 005737 002532 TST ERROR1 ; SEE IF FIRST ERROR
821 021450 001005 BNE 154 ; SKIP HEADER IF ALREADY PRINTED
822 021452 ERRHRD 1,MSG13,ERR7 ; TELL OPERATOR ERROR

```

TEST10: CHECK QVNT HI BITS (MOVING 1)

Address	Hex	Dec	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op41
---------	-----	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

TEST11: CHECK QVNT HI BITS (MOVING 0)

```

840 .SBTTL TEST11: CHECK QVNT HI BITS (MOVING 0)
841 021540 STATST <CHECK QVNT HI BITS (MOVING 0)>
      021540      040      103      110 T11MSG:: .ASCIZ / CHECK QVNT HI BITS (MOVING 0)/
842
843 ;*****
844 ;**
845 ;TEST DESCRIPTION-
846 ; CHECKS THAT BITS 8-15 CAN BE LOADED
847 ;TEST STEPS-
848 ; LOAD MOVING BIT WORD IN QVNT
849 ; READ QVNT, CHECK BITS 8-15
850 ; REPEAT FOR ALL 8 BITS
851 ;
852 ;--
853 ;*****
854
855 021600      BGNTST
      021600 T11::
856 021600      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
857 021640      INIT
      021640      012777      000100  160450 ;INITIALIZE DEVICE
      021646      013777      002520  160442  MOV      #MCMHALT,BQMT1
      021654      005077      160420      MOV      INITWORD,BQMT1
      021660      005037      002532      CLR      BQINT
      021664      012701      000001      CLR      ERROR1
858 021664      012702      000010      MOV      #1,R1
859 021670      012703      000400      MOV      #8,R2
860 021674      010337      002530      MOV      #BIT8,R3
861 021700      005137      002530      MOV      R3,EXP
862 021704      005137      002530      COM      EXP
863
864 ;
865 ; TEST BITS 8-15 WITH MOVING ZERO
866 ;
866 021710      BGNSUB
      021710 T11.1:
      021710      104402      TRAP      C#BSUB
867 021712      104404      10#:: BGNSEG
      021712      042737      000377  002530  TRAP      C#BSEG
868 021722      013777      002530  160346  BIC      #377,EXP
869 021730      017737      160342      002526  MOV      EXP,BQVNT
870 021736      042737      000377      002526  MOV      BQVNT,VAL
871 021744      023737      002530      002526  BIC      #377,VAL
872 021752      001414      002530      002526  CMP      EXP,VAL
873
874      BEQ      20#
875
876 ; ERROR CONDITION

```

```

; MASK OUT UNTESTED BITS
; MOVE EXPECTED TO REGISTER QVNT
; READ QVNT AND MOVE TO VALUE
; MASK OUT UNTESTED BITS
; COMPARE EXPECTED WITH VALUE
; IF OK, BRANCH

```


TEST11: CHECK QVNT MI BITS (MOVING 0)

877	021754	005737	002532	TST	ERROR1		: SEE IF FIRST ERROR
878	021760	001005		BNE	154		: SKIP HEADER IF ALREADY PRINTED
879	021762			ERRHRD	1,MSG14,ERR7		: TELL OPERATOR ERROR
	021762	104456		TRAP	C#ERRHRD		
	021764	000001		.WORD	1		
	021766	005373		.WORD	MSG14		
	021770	012246		.WORD	ERR7		
880	021772	000402		BR	184		: SKIP DETAIL LINE IF HEADER PRINTED
881	021774	004737	012366	154:	CALL	DETAIL	: PRINT DETAIL LINE
882	022000	005237	002532	184:	INC	ERROR1	: INCREMENT ERROR COUNT
883				:			
884				:	END ERROR CONDITION		
885				:			
886	022004			204:	ENDSEG		
	022004			100004:			
	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	104		: 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	5004		: BRANCH IF NO ERRORS
895	022034			PRINTX	#FOR0		: LINE FEED
	022034	012746	003333	MOV	#FOR0,-(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			5004:			
897	022054			ENDTST			
	022054			L10047:			
	022054	104401		TRAP	C#ETST		

TEST12: CHECK ACK - PRS LINK

```

899 .SBTTL TEST12: CHECK ACK - PRS LINK
900 022056 STATST <CHECK ACK - PRS LINK>
    022056      040      103      110 T12MSG:: .ASCIZ / CHECK ACK - PRS LINK/
901
902 ;*****
903 ;**
904 ;TEST DESCRIPTION
905 ; CHECKS THAT CMDACK SETS VNTACK AND CLEARS CMDPRS
906 ; CHECKS THAT VNTPRS SETS CMDPRS AND CLEARS VNTACK
907 ; USES QCMDACK - UEVENTACK LOOP-BACK
908 ; USES QEVNTPRSNT - UCMDPRSNT LOOP-LOOPBACK
909 ;TEST STEPS-
910 ; SET QCMD CMDACK
911 ; CHECK FOR CMDPRS CLEAR
912 ; READ QVNT VNTACK, CHECK FOR SET
913 ; SET VNTPRS
914 ; READ VNTACK, CHECK FOR CLEAR
915 ; READ QCMD CMDPRS, CHECK FOR SET
916 ; LOOPS 100 TIMES OR COMPLETES CURRENT PASS WITH ERROR AND EXITS
917 ;
918 ;--
919 ;*****
920
921 022104      BGNTST
    022104
922 022104      T12::
    022104      CKPNT
    022106      TRAP      C1RFLA      ; PRINT OUT TEST TITLE IF PNT SET
    022112      CMP      @PNT,R0      ; IS PNT FLAG SET?
    022114      BNE      644          ; BRANCH IF NOT SET
    022120      MOV      @T12MSG,-(SP)
    022124      MOV      @PNTMSG,-(SP)
    022130      MOV      @2,-(SP)
    022132      MOV      SP,R0
    022134      TRAP      C1PNTF
    022140      ADD      @6,SP
    022144      644:      CLR      ERROR1      ; CLEAR ERROR COUNTER
    022146      INIT
    022148      ;INITIALIZE DEVICE
    022152      MOV      @MCHALT,BQMT1      ;INITIALIZE Q-BRIDGE
    022154      MOV      INITWORD,BQMT1      ;SET UP LOOPBACK OPTION
    022156      CLR      BQINT              ;CLEAR QINT AFTER LOOPBACK ENABLED
    022158      CLR      ERROR1              ;CLEAR ERROR FLAG
924
925 ;
926 ; LOOP THROUGH TEST 100 TIMES
927 ;
928      MOV      @100.,R2      ; # OF LOOPS TO R2
929 14:
    022174      BGNSEG
    022176      TRAP      C1BSEG
    022178      MOV      @CMDACK,BQCMD      ; SET BIT 0 (CMDACK) IN QCMD
    022180      MOV      BQINT,VAL          ; READ AND STORE QINT REGISTER
    022182      BIT      @CMDPRS,VAL        ; IS CMDPRS CLEAR
    022184      BEQ      24                ; BRANCH IF OK
930
931 ;
932 ; ERROR CONDITION 1
933 ;
934      INC      ERROR1      ; SET ERROR FLAG
935
936
937 022222      005237      002532

```

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#ERRHRD
      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#ERRHRD
      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#ERRHRD
      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	02,R1	: REGISTER # FOR MESSAGE
982	022420	012737	004000 002530	MOV	0CMDPRS,EXP	: VALUE OF EXPECTED FOR MESSAGE
983	022426			ERRMRD	4,MSG18	: GO REPORT ERROR TO OPERATOR
	022426	104456		TRAP	C#ERRMRD	
	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)

```

993 .SBTTL TEST13: CHECK QBA (MOVING 1)
994 022454 STATST <CHECK QBA (MOVING 1)>
995 022454 040 103 110 T13MSG:: .ASCIZ / CHECK QBA (MOVING 1)/
996 ;*****
997 ;**
998 ;TEST DESCRIPTION
999 ; CHECKS ALL REGISTER BITS
1000 ;TEST STEPS-
1001 ; LOAD WORD IN QBA
1002 ; READ QBA, CHECK
1003 ; MOVE BIT
1004 ; REPEAT FOR ALL BITS
1005 ;
1006 ;--
1007 ;*****
1008
1009 022502 BGNTST
1010 022502 T13::
1011 022502 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
1012 022502 104421 TRAP C#RFLA
1013 022504 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1014 022510 001012 BNE 64# ; BRANCH IF NOT SET
1015 022512 012746 022454 MOV #T13MSG,-(SP)
1016 022516 012746 002270 MOV #PNTMSG,-(SP)
1017 022522 012746 000002 MOV #2,-(SP)
1018 022526 010600 MOV SP,R0
1019 022530 104417 TRAP C#PNTF
1020 022532 062706 000006 ADD #6,SP
1021 022536 005037 002532 64#: CLR ERROR1 ; CLEAR ERROR COUNTER
1022 022542 INIT
1023 022542 012777 000100 157546 ;INITIALIZE DEVICE
1024 022550 013777 002520 157540 MOV #MHALT,BQMT1
1025 022556 005077 157516 MOV INITWORD,BQMT1
1026 022562 005037 002532 CLR BQINT
1027 022566 012701 000006 CLR ERROR1
1028 022572 012702 000020 MOV #6,R1 ; INITIALIZE Q-BRIDGE
1029 022576 012737 000001 002530 MOV #16,R2 ; SET UP LOOPBACK OPTION
1030 ; ; CLEAR QINT AFTER LOOPBACK ENABLED
1031 ; ; REGISTER NUMBER TO R1
1032 ; ; NUMBER OF BITS TO CHECK
1033 ; ; FIRST DATA PATTERN TO CHECK
1034 ;
1035 ; TEST BITS 0-15 WITH MOVING ONE
1036 ;
1037 022604 BGNSUB
1038 022604 T13.1:
1039 022604 104402 TRAP C#BSUB
1040 022606 104404 104404 BGNSEG
1041 022610 013777 002530 157472 TRAP C#BSEG
1042 022616 017737 157466 002526 MOV EXP,BQBA ; MOVE EXPECTED TO REGISTER QBA
1043 022624 023737 002530 002526 MOV BQBA,VAL ; READ QBA AND MOVE TO VALUE
1044 022632 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1045 022634 005737 002532 BEQ 20# ; IF OK, BRANCH
1046 022636 001005
1047 022642 ; ERROR CONDITION
1048 ;
1049 022642 TST ERROR1 ; SEE IF FIRST ERROR
1050 022642 BNE 15# ; SKIP HEADER IF ALREADY PRINTED
1051 022642 ERRHRD 1,MSG19,ERR7 ; TELL OPERATOR ERROR

```

TEST13: CHECK QBA (MOVING 1)

022642	104456		TRAP	C#ERHRD	
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)

```

1047 .SBTTL TEST14: CHECK QBA (MOVING 0)
1048 022730 STATST <CHECK QBA (MOVING 0)>
1049 022730 040 103 110 T14MSG:: .ASCIZ / CHECK QBA (MOVING 0)/
1050 ;*****
1051 ;**
1052 ;TEST DESCRIPTION
1053 ; CHECKS ALL REGISTER BITS
1054 ;TEST STEPS-
1055 ; LOAD WORD IN QBA
1056 ; READ QBA, CHECK
1057 ; MOVE BIT
1058 ; REPEAT FOR ALL BITS
1059 ;
1060 ;--
1061 ;*****
1062
1063 022756 BGNTST
1064 022756 T14::
1065 022756 CKPNT
1066 022756 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
1067 022760 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1068 022764 001012 BNE 644 ; BRANCH IF NOT SET
1069 022766 012746 022730 MOV #T14MSG,-(SP)
1070 022772 012746 002270 MOV #PNTMSG,-(SP)
1071 022776 012746 000002 MOV #2,-(SP)
1072 023002 010600 MOV SP,R0
1073 023004 104417 TRAP C#PNTF
1074 023006 062706 000006 ADD #6,SP
1075 023012 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
1076 023016 INIT
1077 023016 ;INITIALIZE DEVICE
1078 023016 MOV #MHALT,BQMT1 ;INITIALIZE Q-BRIDGE
1079 023024 013777 002520 157272 MOV INITWORD,BQMT1 ;SET UP LOOPBACK OPTION
1080 023032 005077 157242 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
1081 023036 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
1082 023042 012701 000006 MOV #6,R1 ; REGISTER NUMBER TO R1
1083 023046 012702 000020 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1084 023052 012703 000001 MOV #BIT0,R3 ; INITIAL DATA PATTERN TO R3
1085 023056 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
1086 023062 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
1087 023066 ;
1088 023066 ; TEST BITS 0-15 WITH MOVING ZERO
1089 023066 ;
1090 023066 BGNSUB
1091 023066 104402 T14.1:
1092 023070 104404 104: TRAP C#BSUB
1093 023072 013777 002530 157210 TRAP C#BSEG
1094 023100 017737 157204 002526 MOV EXP,BQBA ; MOVE EXPECTED TO REGISTER QBA
1095 023106 023737 002530 002526 MOV BQBA,VAL ; READ QBA AND MOVE TO VALUE
1096 023114 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1097 023116 005737 002532 BEQ 204 ; IF OK, BRANCH
1098 023116 ;
1099 023116 ; ERROR CONDITION
1100 023116 ;
1101 023116 TST ERROR1 ; SEE IF FIRST ERROR

```

TEST14: CHECK QBA (MOVING 0)

Line	Address	Hex	Label	Instruction	Comment
1084	023122	001005	BNE	15:	; SKIP HEADER IF ALREADY PRINTED
1085	023124		ERRHRD	1,MSG20,ERR7	; TELL OPERATOR ERROR
	023124	104456	TRAP	C#ERRHRD	
	023126	000001	.WORD	1	
	023130	006033	.WORD	MSG20	
	023132	012246	.WORD	ERR7	
1086	023134	000402	BR	18:	; SKIP DETAIL LINE IF HEADER PRINTED
1087	023136	004737	CALL	DETAIL	; PRINT DETAIL LINE
1088	023142	005237	INC	ERROR1	; INCREMENT ERROR COUNT
1089					
1090					
1091					
1092	023146				
	023146				
	023146	104405			
1093	023150	006303	TRAP	C#ESEG	
1094	023152	010337	ASL	R3	; SHIFT BIT FOR NEXT TEST
1095	023156	005137	MOV	R3,EXP	; PLACE PATTERN IN EXPECTED
1096	023162	005302	COM	EXP	; COMPLIMENT FOR MOVING 0
1097	023164	001341	DEC	R2	; DECREMENT LOOP COUNTER
1098	023166		BNE	10:	; IF NOT DONE, BRANCH BACK
	023166		ENDSUB		
	023166	104403			
1099	023170	005737	L10055:	TRAP	C#ESUB
1100	023174	001410	TST	ERROR1	; ANY ERRORS
1101	023176		BEQ	500:	; BRANCH IF NO ERRORS
	023176	012746	PRINTX	#FOR0	; LINE FEED
	023202	012746	MOV	#FOR0, -(SP)	
	023206	010600	MOV	#1, -(SP)	
	023210	104415	MOV	SP,R0	
	023212	062706	TRAP	C#PNTX	
1102	023216		ADD	#4, SP	
1103	023216				
	023216	104401	500:	ENDTST	
	023216		L10054:	TRAP	C#ETST

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

Line	Address	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op46
------	---------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

Line	Address	Op-Code	Register	Value	Instruction	Comment
1142	023444	023737	002530	002526	120: CMP EXP,VAL	; COMPARE EXPECTED WITH VALUE
1143	023452	001414			BEQ 200	; IF OK, BRANCH
1144						
1145					; ERROR CONDITION	
1146						
1147	023454	005737	002532		TST ERROR1	; SEE IF FIRST ERROR
1148	023460	001005			BNE 150	; SKIP HEADER IF ALREADY PRINTED
1149	023462				ERRHRD 1,MSG21,ERR7	; TELL OPERATOR ERROR
	023462	104456			TRAP C0ERRHRD	
	023464	000001			.WORD 1	
	023466	006060			.WORD MSG21	
	023470	012246			.WORD ERR7	
1150	023472	000402			BR 180	; SKIP DETAIL LINE IF HEADER PRINTED
1151	023474	004737	012366		150: CALL DETAIL	; PRINT DETAIL LINE
1152	023500	005237	002532		180: INC ERROR1	; INCREMENT ERROR COUNT
1153						
1154					; END ERROR CONDITION	
1155						
1156	023504				200: ENDSEG	
	023504				100000: TRAP C0ESEG	
	023504	104405			ASL EXP	; SHIFT BIT FOR NEXT TEST
1157	023506	006337	002530		DEC R2	; DECREMENT LOOP COUNTER
1158	023512	005302			BNE 100	; IF NOT DONE, BRANCH BACK
1159	023514	001333			ENDSUB	
1160	023516				L10057: TRAP C0ESUB	
	023516	104403			TST ERROR1	; ANY ERRORS
1161	023520	005737	002532		BEQ 5000	; BRANCH IF NO ERRORS
1162	023524	001410			PRINTX 0FOR0	; LINE FEED
1163	023526				MOV 0FOR0, -(SP)	
	023526	012746	003333		MOV 01, -(SP)	
	023532	012746	000001		MOV SP, R0	
	023536	010600			TRAP C0PNTX	
	023540	104415			ADD 04, SP	
	023542	062706	000004		5000: ENDTST	
1164	023546				L10056: TRAP C0ETST	
1165	023546					
	023546					
	023546	104401				

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

```

1167 .SBTTL TEST16: CHECK QDMA HI BITS (8-13) (MOVING 0)
1168 023550 STATST <CHECK QDMA HI BITS (8-13) (MOVING 0)>
1169 023550 040 103 110 T16MSG: .ASCIZ / CHECK QDMA HI BITS (8-13) (MOVING 0)/
1170 ;*****
1171 ;**
1172 ;TEST DESCRIPTION-
1173 ; CHECKS HI Q-BUS ADDRESS BITS
1174 ;TEST STEPS-
1175 ; LOAD QDMA BITS 8-13
1176 ; IF U-BRIDGE THEN LOAD BITS 8,9
1177 ; READ QDMA, CHECK BITS
1178 ; MOVE BIT
1179 ; REPEAT FOR ALL BITS
1180 ;
1181 ;--
1182 ;*****
1183
1184 023616 BGNTST
1185 023616 T16:
1186 023616 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
1187 023616 TRAP C1RFLA ; IS PNT FLAG SET?
1188 023620 022700 001000 CMP #PNT,R0 ; BRANCH IF NOT SET
1189 023624 001012 BNE 641
1190 023626 012746 023550 MOV #T16MSG,-(SP)
1191 023632 012746 002270 MOV #PNTMSG,-(SP)
1192 023636 012746 000002 MOV #2,-(SP)
1193 023642 010600 MOV SP,R0
1194 023644 104417 TRAP C1PNTF
1195 023646 062706 000006 ADD #6,SP
1196 023652 005037 002532 641: CLR ERROR1 ; CLEAR ERROR COUNTER
1197 023656 INIT
1198 023656 ;INITIALIZE DEVICE
1199 023656 MOV #QHALT,BQMT1 ;INITIALIZE Q-BRIDGE
1200 023664 013777 002520 156432 MOV INITWORD,BQMT1 ;SET UP LOOPBACK OPTION
1201 023672 005077 156402 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
1202 023676 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
1203 023702 012701 000004 MOV #4,R1 ; REGISTER NUMBER TO R1
1204 023706 012702 000006 MOV #6,R2 ; NUMBER OF BITS TO CHECK
1205 023712 005737 002500 TST BRIDGETYPE ; TEST BRIDGE TYPE
1206 023716 001402 BEQ 11 ; IF EQ THEN Q-BRIDGE
1207 023720 012702 000002 MOV #2,R2 ; NUMBER OF BITS (U-BRIDGE)
1208 023724 012703 000400 11: MOV #BIT8,R3 ; INITIAL DATA PATTERN TO R3
1209 023730 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
1210 023734 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
1211 023740 ;
1212 023740 ; TEST BITS 8-13 WITH MOVING ZERO
1213 023740 ;
1214 023740 BGNSUB
1215 023740 T16.1:
1216 023740 104402 TRAP C1BSUB
1217 023742 104404 BGNSEG
1218 023744 005737 002500 TRAP C1BSEG
1219 023750 001015 TST BRIDGETYPE ; TEST BRIDGE TYPE
1220 023752 042737 140377 002530 BNE 111 ; IF NE THEN U-BRIDGE
1221 023760 013777 002530 156316 BIC #140377,EXP ; CLEAR UNTESTED BITS
1222 023760 013777 002530 156316 MOV EXP,BQDMA ; MOVE EXPECTED TO REGISTER QDMA

```


TEST17: CHECK QDMA UXAD BITS

```

1238 .SBTTL TEST17: CHECK QDMA UXAD BITS
1239 STATST <CHECK QDMA UXAD BITS>
1240 024146 040 103 110 T17MSG:: .ASCIZ / CHECK QDMA UXAD BITS/
1241
1242 ;*****
1243 ;**
1244 ;TEST DESCRIPTION-
1245 ; CHECKS UNIBUS EXTENDED ADDRESS BITS
1246 ; LDUXAD DRIVER AND I21 ARE NOT CHECKED
1247 ;TEST STEPS-
1248 ; LOAD QDMA BITS 4,5
1249 ; READ QDMA, CHECK BITS
1250 ; INCREMENT PATTERN
1251 ; REPEAT FOR ALL PATTERNS
1252 ;
1253 ;--
1254 ;*****
1255 024174 BGNTST
1256 024174 T17::
1257 024174 CKPNT
1258 024174 TRAP C0RFLA ; PRINT OUT TEST TITLE IF PNT SET
1259 024174 CMP #PNT,R0 ; IS PNT FLAG SET?
1260 024174 BNE 640 ; BRANCH IF NOT SET
1261 024174 MOV #T17MSG,-(SP)
1262 024174 MOV #PNTMSG,-(SP)
1263 024174 MOV #2,-(SP)
1264 024174 MOV SP,R0
1265 024174 TRAP C0PNTF
1266 024174 ADD #6,SP
1267 024174 CLR ERROR1 ; CLEAR ERROR COUNTER
1268 024174 INIT
1269 024174 ;INITIALIZE DEVICE
1270 024174 MOV #M0HALT,BQMT1 ;INITIALIZE Q-BRIDGE
1271 024174 MOV INITWORD,BQMT1 ;SET UP LOOPBACK OPTION
1272 024174 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
1273 024174 CLR ERROR1 ;CLEAR ERROR FLAG
1274 024174 MOV #4,R1 ; REGISTER NUMBER TO R1
1275 024174 BGNSUB
1276 024174 T17.1: TRAP C0BSUB
1277 024174 ;
1278 024174 ; TEST WITH BOTH BITS HIGH
1279 024174 ;
1280 024174 MOV #60,EXP ; BIT PATTERN IN EXPECTED
1281 024174 MOV EXP,BQDMA ; MOVE PATTERN TO QDMA REGISTER
1282 024174 MOV BQDMA,VAL ; GET VALUE OF REGISTER
1283 024174 BIC #177717,VAL ; MASK OUT UNDETERMINED BITS
1284 024174 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1285 024174 BEQ 10 ; IF OK, BRANCH
1286 024174 ;
1287 024174 ; ERROR CONDITION 1
1288 024174 ;
1289 024174 ERRHRD 1,MSG23,ERR4 ; TELL OPERATOR ERROR
1290 024174 TRAP C0ERRRD
1291 024174 .WORD 1
1292 024174 .WORD MSG23

```

TEST17: CHECK QDMA UXAD BITS

1273	024334	011776			16:	.WORD	ERR4	
1274	024336					ENDSUB		
	024336				L10063:	TRAP	C#ESUB	
1275	024340	104403				BGNSUB		
	024340				T17.2:	TRAP	C#BSUB	
1276	024340	104402						
1277								
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	24	: IF OK, BRANCH
1285								
1286								
1287								
1288	024402							
	024402	104456				ERRHRD	2,MSG23,ERR4	: TELL OPERATOR ERROR
	024404	000002				TRAP	C#ERHRD	
	024406	006172				.WORD	2	
	024410	011776				.WORD	MSG23	
						.WORD	ERR4	
1289	024412				24:			
1290	024412					ENDSUB		
	024412				L10064:	TRAP	C#ESUB	
1291	024414	104403				BGNSUB		
	024414				T17.3:	TRAP	C#BSUB	
	024414	104402						
1292								
1293								
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	34	: IF OK, BRANCH
1301								
1302								
1303								
1304	024456							
	024456	104456				ERRHRD	3,MSG23,ERR4	: TELL OPERATOR ERROR
	024460	000003				TRAP	C#ERHRD	
	024462	006172				.WORD	3	
	024464	011776				.WORD	MSG23	
						.WORD	ERR4	
1305	024466				34:			
1306	024466					ENDSUB		
	024466				L10065:	TRAP	C#ESUB	
1307	024470	104403				BGNSUB		
	024470				T17.4:	TRAP	C#BSUB	
1308	024470	104402						

TEST17: CHECK QDMA UXAD BITS

1309					; TEST WITH BOTH BITS LOW	
1310						
1311	024472	012737	000000	002530	MOV	#00,EXP
1312	024500	013777	002530	155576	MOV	EXP,QDMA
1313	024506	017737	155572	002526	MOV	QDMA,VAL
1314	024514	042737	177717	002526	BIC	#177717,VAL
1315	024522	023737	002530	002526	CMP	EXP,VAL
1316	024530	001404			BEQ	46
1317						
1318					; ERROR CONDITION 4	
1319						
1320	024532				ERRHRD	4,MSG23,ERR4
	024532	104456			TRAP	C#ERRHRD
	024534	000004			.WORD	4
	024536	006172			.WORD	MSG23
	024540	011776			.WORD	ERR4
1321	024542				46:	
1322	024542				ENDSUB	
	024542				L10066:	
	024542	104403			TRAP	C#ESUB
1323	024544				ENDTST	
	024544				L10062:	
	024544	104401			TRAP	C#ETST

; BIT PATTERN IN EXPECTED
 ; MOVE PATTERN TO QDMA REGISTER
 ; GET VALUE OF REGISTER
 ; MASK OUT UNDETERMINED BITS
 ; COMPARE EXPECTED WITH VALUE
 ; IF OK, BRANCH

; TELL OPERATOR ERROR

TEST18: CHECK QUBA (MOVING 1)

```

1325 .SBTTL TEST18: CHECK QUBA (MOVING 1)
1326 024546 STATST <CHECK QUBA (MOVING 1)>
1327 024546 T18MSG:: .ASCIZ / CHECK QUBA (MOVING 1)/
1328
1329 ;*****
1330 ;**
1331 ;TEST DESCRIPTION
1332 ; CHECKS ALL REGISTER BITS
1333 ; USES QTSOUT - QTSIN LOOP-BACK
1334 ;TEST STEPS-
1335 ; SET QMT1 FUNC = 4 (SELUBAR)
1336 ; LOAD WORD IN QUBA
1337 ; READ QMT0, CHECK
1338 ; MOVE BIT
1339 ; REPEAT FOR ALL BITS
1340 ;
1341 ;--
1342 ;*****
1343 024576 BGNTST
1344 024576 T18::
1345 024576 CKPNT
1346 024576 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
1347 024600 104421 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1348 024604 001012 000000 BNE 644 ; BRANCH IF NOT SET
1349 024606 012746 024546 MOV #T18MSG,-(SP)
1350 024612 012746 002270 MOV #PNTMSG,-(SP)
1351 024616 012746 000002 MOV #2,-(SP)
1352 024622 010600 000000 MOV SP,R0
1353 024624 104417 000006 TRAP C#PNTF
1354 024626 062706 000006 ADD #6,SP
1355 024632 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
1356 024636 INIT
1357 024636 ;INITIALIZE DEVICE
1358 024636 MOV #QHALT,QMT1 ;INITIALIZE Q-BRIDGE
1359 024644 013777 002520 155452 MOV INITWORD,QMT1 ;SET UP LOOPBACK OPTION
1360 024652 005077 155422 155444 CLR QINT ;CLEAR QINT AFTER LOOPBACK ENABLED
1361 024656 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
1362 024662 012701 000005 MOV #5,R1 ; REGISTER NUMBER TO R1
1363 024666 012702 000020 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1364 024672 012737 000001 002530 MOV #BIT0,EXP ; FIRST DATA PATTERN TO CHECK
1365 024700 052777 000020 155410 BIS #SELUBAR,QMT1 ; SELECT QMT1 FUNC = 4
1366
1367 ; TEST BITS 0-15 WITH MOVING ONE
1368 ;
1369 BGNSUB
1370 T18.1:
1371 104: TRAP C#BSUB
1372 BGNSUB
1373 TRAP C#BSEG
1374 MOV EXP,QUBA ; MOVE EXPECTED TO REGISTER QUBA
1375 MOV QMT0,VAL ; READ QMT0 AND MOVE TO VALUE
1376 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1377 BEQ 204 ; IF OK, BRANCH
1378
1379 ; ERROR CONDITION
1380 ;

```

TEST18: CHECK QUBA (MOVING 1)

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

TEST19: CHECK QUBA (MOVING 0)

```

1382 .SBTTL TEST19: CHECK QUBA (MOVING 0)
1383 STATST <CHECK QUBA (MOVING 0)>
1384 025032 040 103 110 T19MSG:: .ASCIZ / CHECK QUBA (MOVING 0)/
1385 ;*****
1386 ;**
1387 ;TEST DESCRIPTION
1388 ; CHECKS ALL REGISTER BITS
1389 ; USES QTSOUT - QTSIN LOOP-BACK
1390 ;TEST STEPS-
1391 ; SET QMT1 FUNC = 4 (SELUBAR)
1392 ; LOAD WORD IN QUBA
1393 ; READ QMT0, CHECK
1394 ; MOVE BIT
1395 ; REPEAT FOR ALL BITS
1396 ;
1397 ;--
1398 ;*****
1399
1400 025062 BGNTST
1401 025062 T19::
1402 025062 CKPNT
1403 025062 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
1404 025064 104421 CMP #PNT,R0 ; IS PNT FLAG SET?
1405 025064 022700 001000 BNE 64$ ; BRANCH IF NOT SET
1406 025070 001012 MOV #T19MSG,-(SP)
1407 025072 012746 025032 MOV #PNTMSG,-(SP)
1408 025076 012746 002270 MOV #2,-(SP)
1409 025102 012746 000002 MOV SP,R0
1410 025106 010600 TRAP C#PNTF
1411 025110 104417 ADD #6,SP
1412 025112 062706 000006 64$: CLR ERROR1 ; CLEAR ERROR COUNTER
1413 025116 005037 002532 INIT
1414 025122 ;INITIALIZE DEVICE
1415 025122 012777 000100 155166 MOV #QHALT,QMT1 ;INITIALIZE Q-BRIDGE
1416 025130 013777 002520 155160 MOV INITWORD,QMT1 ;SET UP LOOPBACK OPTION
1417 025136 005077 155136 CLR QINT ;CLEAR QINT AFTER LOOPBACK ENABLED
1418 025142 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
1419 025146 012701 000005 MOV #5,R1 ; REGISTER NUMBER TO R1
1420 025152 012702 000020 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1421 025156 012703 000001 MOV #BIT0,R3 ; INITIAL DATA PATTERN TO R3
1422 025162 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
1423 025166 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
1424 025172 052777 000020 155116 BIS #SELUBAR,QMT1 ; SELECT QMT1 FUNC = 4
1425
1426 ; TEST BITS 0-15 WITH MOVING ZERO
1427 ;
1428 BGNSUB
1429 T19.1:
1430 025200 TRAP C#BSUB
1431 025200 104402 BGNSEG
1432 025202 104404 TRAP C#BSEG
1433 025202 104404 MOV EXP,QUBA ; MOVE EXPECTED TO REGISTER QUBA
1434 025204 013777 002530 155074 MOV QMT0,VAL ; READ QMT0 AND MOVE TO VALUE
1435 025212 017737 155076 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1436 025220 023737 002530 002526 BEQ 20$ ; IF OK, BRANCH
1437 025226 001414
1438

```


TEST19: CHECK QUBA (MOVING 0)

```

1419      ; ERROR CONDITION
1420      ;
1421 025230 005737 002532      TST      ERROR1      ; SEE IF FIRST ERROR
1422 025234 001005      BNE      154      ; 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      184      ; SKIP DETAIL LINE IF HEADER PRINTED
1425 025250 004737 012366      154:      CALL      DETAIL      ; PRINT DETAIL LINE
1426 025254 005237 002532      184:      INC      ERROR1      ; INCREMENT ERROR COUNT
1427
1428      ; END ERROR CONDITION
1429      ;
1430 025260      204:      ENDSEG
      025260      100004:
      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      104      ; IF NOT DONE, BRANCH BACK
1436 025300      ENDSUB
      025300
      025300 104403      L10072:      TRAP      C:ESUB
1437 025302 005737 002532      TST      ERROR1      ; ANY ERRORS
1438 025306 001410      BEQ      5004      ; BRANCH IF NO ERRORS
1439 025310      PRINTX 4,0      ; LINE FEED
      025310 012746 003333      MOV      4,0,--(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      5004:
1441 025330      L10071:      ENDTST
      025330      TRAP      C:ETST
      025330 104401

```

TEST20: CHECK QWC (MOVING 1)

```

1443 .SBTTL TEST20: CHECK QWC (MOVING 1)
1444 025332 STATST <CHECK QWC (MOVING 1)>
1445 025332 040 103 110 T20MSG:: .ASCIZ / CHECK QWC (MOVING 1)/
1446 ;*****
1447 ;**
1448 ;TEST DESCRIPTION
1449 ; CHECKS ALL REGISTER BITS
1450 ;TEST STEPS-
1451 ; LOAD WORD IN QWC
1452 ; READ QWC, CHECK
1453 ; MOVE BIT
1454 ; REPEAT FOR ALL BITS
1455 ;
1456 ;--
1457 ;*****
1458
1459 025360 BGNTST
1460 025360 T20::
1461 025360 CKPNT
1462 025360 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
1463 025362 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1464 025366 001012 BNE 64# ; BRANCH IF NOT SET
1465 025370 012746 025332 MOV #T20MSG,-(SP)
1466 025374 012746 002270 MOV #PNTMSG,-(SP)
1467 025400 012746 000002 MOV #2,-(SP)
1468 025404 010600 MOV SP,R0
1469 025406 104417 TRAP C#PNTF
1470 025410 062706 000006 ADD #6,SP
1471 025414 005037 002532 64#: CLR ERROR1 ; CLEAR ERROR COUNTER
1472 025420 INIT
1473 025420 012777 000100 154670 ; INITIALIZE DEVICE
1474 025426 013777 002520 154662 MOV #MQHALT,@QMT1
1475 025434 005077 154640 MOV INITWORD,@QMT1
1476 025440 005037 002532 CLR @QINT ; INITIALIZE Q-BRIDGE
1477 025444 012701 000007 CLR ERROR1 ; SET UP LOOPBACK OPTION
1478 025450 012702 000020 MOV #7,R1 ; CLEAR QINT AFTER LOOPBACK ENABLED
1479 025454 012737 000001 002530 MOV #16,R2 ; CLEAR ERROR FLAG
1480 ; ; REGISTER NUMBER TO R1
1481 ; ; NUMBER OF BITS TO CHECK
1482 ; ; FIRST DATA PATTERN TO CHECK
1483 ;
1484 ; TEST BITS 0-15 WITH MOVING ONE
1485 ;
1486 025462 BGNSUB
1487 025462 T20.1:
1488 025462 104402 TRAP C#BSUB
1489 025464 104404 10# BGNSEG
1490 025466 013777 002530 154616 TRAP C#BSEG
1491 025474 017737 154612 002526 MOV EXP,@QWC ; MOVE EXPECTED TO REGISTER QWC
1492 025502 023737 002530 002526 MOV @QWC,VAL ; READ QWC AND MOVE TO VALUE
1493 025510 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1494 ; BEQ 20# ; IF OK, BRANCH
1495 ;
1496 ; ERROR CONDITION
1497 ;
1498 025512 005737 002532 TST ERROR1 ; SEE IF FIRST ERROR
1499 025516 001005 BNE 15# ; SKIP HEADER IF ALREADY PRINTED
1500 025520 ERRHRD 1,MSG26,ERR7 ; TELL OPERATOR ERROR

```

TEST20: CHECK QWC (MOVING 1)

Line	Address	Hex	Label	Instruction	Comment
	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)

```

1497 .SBTTL TEST21: CHECK QWC (MOVING 0)
1498 STATST <CHECK QWC (MOVING 0)>
1499 025606 040 103 110 T21MSG:: .ASCIZ / CHECK QWC (MOVING 0)/
1500 ;.....
1501 ;..
1502 ;TEST DESCRIPTION
1503 ; CHECKS ALL REGISTER BITS
1504 ;TEST STEPS-
1505 ; LOAD WORD IN QWC
1506 ; READ QWC, CHECK
1507 ; MOVE BIT
1508 ; REPEAT FOR ALL BITS
1509 ;
1510 ;..
1511 ;.....
1512
1513 025634 BGNTST
1514 025634 T21::
1515 025634 CKPNT
1516 025634 TRAP C0RFLA ; PRINT OUT TEST TITLE IF PNT SET
1517 025636 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1518 025642 001012 BNE 640 ; BRANCH IF NOT SET
1519 025644 012746 025606 MOV #T21MSG,-(SP)
1520 025650 012746 002270 MOV #PNTMSG,-(SP)
1521 025654 012746 000002 MOV #2,-(SP)
1522 025660 010600 MOV SP,R0
1523 025662 104417 TRAP C0PNTF
1524 025664 062706 000006 ADD #6,SP
1525 025670 005037 002532 640: CLR ERROR1 ; CLEAR ERROR COUNTER
1526 1515 025674 INIT
1527 ;INITIALIZE DEVICE
1528 025674 012777 000100 154414 MOV #QHALT,Q0MT1 ; INITIALIZE Q-BRIDGE
1529 025702 013777 002520 154406 MOV INITWORD,Q0MT1 ; SET UP LOOPBACK OPTION
1530 025710 005077 154364 CLR Q0INT ; CLEAR QINT AFTER LOOPBACK ENABLED
1531 025714 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
1532 1516 025720 012701 000007 MOV #7,R1 ; REGISTER NUMBER TO R1
1533 1517 025724 012702 000020 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1534 1518 025730 012703 000001 MOV #BIT0,R3 ; INITIAL DATA PATTERN TO R3
1535 1519 025734 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
1536 1520 025740 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
1537 1521 ;
1538 1522 ; TEST BITS 0-15 WITH MOVING ZERO
1539 1523 ;
1540 1524 025744 BGNSUB
1541 025744 T21.1:
1542 025744 104402 TRAP C0BSUB
1543 1525 025746 104404 BGNSEG
1544 1526 025750 013777 002530 154334 TRAP C0BSEG
1545 1527 025756 017737 154330 002526 MOV EXP,Q0WC ; MOVE EXPECTED TO REGISTER QWC
1546 1528 025764 023737 002530 002526 MOV Q0WC,VAL ; READ QWC AND MOVE TO VALUE
1547 1529 025772 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1548 1530 BEQ 200 ; IF OK, BRANCH
1549 1531 ;
1550 1532 ; ERROR CONDITION
1551 1533 025774 005737 002532 TST ERROR1 ; SEE IF FIRST ERROR

```

TEST21: CHECK QMC (MOVING 0)

1534	026000	001005		BNE	150		: SKIP HEADER IF ALREADY PRINTED
1535	026002			ERRHRD	1,MSG27,ERR7		: TELL OPERATOR ERROR
	026002	104456		TRAP	C1ERRHRD		
	026004	000001		.WORD	1		
	026006	006342		.WORD	MSG27		
	026010	012246		.WORD	ERR7		
1536	026012	000402		BR	180		: SKIP DETAIL LINE IF HEADER PRINTED
1537	026014	004737	012366	CALL	DETAIL		: PRINT DETAIL LINE
1538	026020	005237	002532	INC	ERROR1		: INCREMENT ERROR COUNT
1539							
1540							
1541							
1542	026024						
	026024						
	026024	104405					
1543	026026	006303		TRAP	C1ESEG		: SHIFT BIT FOR NEXT TEST
1544	026030	010337	002530	ASL	R3		: PLACE PATTERN IN EXPECTED
1545	026034	005137	002530	MOV	R3,EXP		: COMPLIMENT FOR MOVING 0
1546	026040	005302		COM	EXP		: DECREMENT LOOP COUNTER
1547	026042	001341		DEC	R2		: IF NOT DONE, BRANCH BACK
1548	026044			BNE	100		
	026044			ENDSUB			
	026044	104403					
1549	026046	005737	002532	L10076: TRAP	C1ESUB		: ANY ERRORS
1550	026052	001410		TST	ERROR1		: BRANCH IF NO ERRORS
1551	026054			BEQ	5000		: LINE FEED
	026054	012746	003333	PRINTX	#FORO		
	026060	012746	000001	MOV	#FORO,-(SP)		
	026064	010600		MOV	#1,-(SP)		
	026066	104415		MOV	SP,R0		
	026070	062706	000004	TRAP	C1PNTX		
1552	026074			ADD	#4,SP		
1553	026074			5000: ENDTST			
	026074			L10075: TRAP	C1ETST		
	026074	104401					

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

```

1555 .SBTTL TEST22: CHECK QVAD (Q-REGISTER) (MOVING 1)
1556 026076 STATST <CHECK QVAD (Q-REGISTER) (MOVING 1)>
1557 026076 T22MSG:: .ASCIZ / CHECK QVAD (Q-REGISTER) (MOVING 1)/
1558 ;*****
1559 ;**
1560 ;TEST DESCRIPTION
1561 ; CHECKS THAT QVAD (A-COUNTER) CAN BE LOADED AND READ
1562 ;TEST STEPS-
1563 ; LOAD WORD IN QVAD
1564 ; SET QMT1 MVADR BIT
1565 ; READ QVAD, CHECK
1566 ; CLEAR QMT1 MVADR BIT
1567 ; MOVE BIT
1568 ; REPEAT FOR ALL BITS
1569 ;
1570 ;--
1571 ;*****
1572
1573 026142 BGNTST
1574 026142 T22::
1575 026142 CKPNT
1576 026142 TRAP C0FLA ; PRINT OUT TEST TITLE IF PNT SET
1577 026142 CMP #PNT,R0 ; IS PNT FLAG SET?
1578 026144 104421 001000 BNE 640 ; BRANCH IF NOT SET
1579 026150 001012 026076 MOV #T22MSG,-(SP)
1580 026152 012746 002270 MOV #PNTMSG,-(SP)
1581 026156 012746 000002 MOV #2,-(SP)
1582 026162 010600 SP,R0
1583 026166 104417 TRAP C0PNTF
1584 026172 062706 000006 ADD #6,SP
1585 026176 005037 002532 640: CLR ERROR1 ; CLEAR ERROR COUNTER
1586 ;
1587 ; INITIALIZE DEVICE
1588 ;
1589 026202 012777 000100 154106 MOV #MOMHALT,BQMT1 ; INITIALIZE Q-BRIDGE
1590 026210 005037 002532 CLR ERROR1 ; CLEAR ERROR COUNTER
1591 026214 012701 000012 MOV #10.,R1 ; REGISTER NUMBER TO R1
1592 026220 012702 000020 MOV #16.,R2 ; NUMBER OF BITS TO CHECK
1593 026224 012737 000001 002530 MOV #BIT0,EXP ; FIRST DATA PATTERN TO CHECK
1594 ;
1595 ; TEST BITS 0-15 WITH MOVING ONE
1596 ;
1597 026232 BGNSUB
1598 026232 T22.1:
1599 026232 104402 TRAP C0BSUB
1600 026234 104404 100: BGNSEG
1601 026236 052777 000020 154036 TRAP C0BSEG
1602 026244 013777 002530 154046 BIS #QVREN,BQCTL ; READ ENABLE
1603 026252 052777 000002 154036 MOV EXP,BQVAD ; MOVE EXPECTED TO REGISTER QVAD
1604 026260 017737 154034 002526 BIS #MVADR,BQMT1 ; SET MVADR BIT IN QMT1
1605 026266 042777 000002 154022 MOV BQVAD,VAL ; READ QVAD AND MOVE TO VALUE
1606 026274 023737 002530 002526 BIC #MVADR,BQMT1 ; CLEAR MVADR BIT IN QMT1
1607 026302 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1608 1595 BEQ 200 ; IF OK, BRANCH
1609 1596 ; 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$ERRHRD
      026314 000001              .WORD    1
      026316 006367              .WORD    MSG28
      026320 012246              .WORD    ERR7
1601 026322 000402              BR       18$          ; 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$ESEG
1608 026336 006337 002530              ASL      EXP
1609 026342 005302              DEC      R2
1610 026344 001333              BNE      10$
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
      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              L10077:
      026376 104401              TRAP     C$ETST

```

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

```

1618 .SBTTL TEST23: CHECK QVAD (Q-REGISTER) (MOVING 0)
1619 026400 STATST <CHECK QVAD (Q-REGISTER) (MOVING 0)>
1620 026400 T23MSG:: .ASCIZ / CHECK QVAD (Q-REGISTER) (MOVING 0)/
1621 ;*****
1622 ;**
1623 ;TEST DESCRIPTION
1624 ; CHECKS THAT QVAD (A-COUNTER) CAN BE LOADED AND READ
1625 ;TEST STEPS-
1626 ; LOAD WORD IN QVAD
1627 ; SET QMT1 MVADR BIT
1628 ; READ QVAD, CHECK
1629 ; CLEAR QMT1 MVADR BIT
1630 ; MOVE BIT
1631 ; REPEAT FOR ALL BITS
1632 ;
1633 ;--
1634 ;*****
1635
1636 026444 BGNTST
1637 026444 T23::
1638 026444 CKPNT
1639 026444 TRAP C0RFLA ; PRINT OUT TEST TITLE IF PNT SET
1640 026446 104421 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1641 026452 001012 BNE 640 ; BRANCH IF NOT SET
1642 026454 012746 026400 MOV #T23MSG,-(SP)
1643 026460 012746 002270 MOV #PNTMSG,-(SP)
1644 026464 012746 000002 MOV #2,-(SP)
1645 026470 010600 MOV SP,R0
1646 026472 104417 TRAP C0PNTF
1647 026474 062706 000006 ADD #6,SP
1648 026500 005037 002532 640: CLR ERROR1 ; CLEAR ERROR COUNTER
1649 ;
1650 ; INITIALIZE DEVICE
1651 ;
1652 026504 012777 000100 153604 MOV #QHALT,QMT1 ; INITIALIZE Q-BRIDGE
1653 026512 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
1654 026516 012701 000012 MOV #10,R1 ; REGISTER NUMBER TO R1
1655 026522 012702 000020 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1656 026526 012703 000001 MOV #BIT0,R3 ; INITIAL DATA PATTERN TO R3
1657 026532 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
1658 026536 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
1659 ;
1660 ; TEST BITS 0-15 WITH MOVING ZERO
1661 ;
1662 026542 BGNSUB
1663 026542 T23.1:
1664 026542 104402 TRAP C0BSUB
1665 026544 104404 104: BGNSEG
1666 026546 052777 000020 153526 TRAP C0BSEG
1667 026554 013777 002530 153536 BIS #QVREN,QCTL ; READ ENABLE
1668 026562 052777 000002 153526 MOV EXP,QVAD ; MOVE EXPECTED TO REGISTER QVAD
1669 026570 017737 153524 002526 BIS #MVADR,QMT1 ; SET MVADR BIT IN QMT1
1670 026576 042777 000002 153512 MOV QVAD,VAL ; READ QVAD AND MOVE TO VALUE
1671 026604 023737 002530 002526 BIC #MVADR,QMT1 ; CLEAR MVADR BIT IN QMT1
1672 026612 001414 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1673 BEQ 200 ; IF OK, BRANCH

```

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

```

1660
1661
1662
1663 026614 005737 002532
1664 026620 001005
1665 026622 104456
      026622 000001
      026626 006430
      026630 012246
1666 026632 000402
1667 026634 004737 012366
1668 026640 005237 002532
1669
1670
1671
1672 026644
      026644 104405
1673 026646 006303
1674 026650 010337 002530
1675 026654 005137 002530
1676 026660 005302
1677 026662 001330
1678 026664
      026664 104403
1679 026666 005737 002532
1680 026672 001410
1681 026674
      026674 012746 003333
      026700 012746 000001
      026704 010600
      026706 104415
      026710 062706 000004
1682 026714
1683 026714
      026714 104401
1684 026716
      026716 012777 000100 153372
      026724 013777 002520 153364
      026732 005077 153342
      026736 005037 002532

; ERROR CONDITION
;
      TST      ERROR1
      BNE      154
      ERRHRD   1,MSG29,ERR7
      TRAP     C1ERRHRD
      .WORD    1
      .WORD    MSG29
      .WORD    ERR7
      BR       184
154:   CALL    DETAIL
184:   INC     ERROR1
; SEE IF FIRST ERROR
; SKIP HEADER IF ALREADY PRINTED
; TELL OPERATOR ERROR
;
; SKIP DETAIL LINE IF HEADER PRINTED
; PRINT DETAIL LINE
; INCREMENT ERROR COUNT
;
; END ERROR CONDITION
;
204:   ENDSEG
100004:
      TRAP     C1ESEG
      ASL      R3
      MOV      R3,EXP
      COM      EXP
      DEC      R2
      BNE      104
      ENDSUB
L10102:
      TRAP     C1ESUB
      TST      ERROR1
      BEQ      5004
      PRINTX   #FORO
      MOV      #FORO,-(SP)
      MOV      #1,-(SP)
      MOV      SP,R0
      TRAP     C1PNTX
      ADD      #4,SP
; SHIFT BIT FOR NEXT TEST
; PLACE PATTERN IN EXPECTED
; COMPLIMENT FOR MOVING 0
; DECREMENT LOOP COUNTER
; IF NOT DONE, BRANCH BACK
;
; ANY ERRORS
; BRANCH IF NO ERRORS
; LINE FEED
;
5004:   ENDTST
L10101:
      TRAP     C1ETST
      INIT
; INITIALIZE DEVICE
      MOV      #MQHALT,BQMT1
      MOV      INITWORD,BQMT1
      CLR      BQINT
      CLR      ERROR1
; INITIALIZE Q-BRIDGE
; SET UP LOOPBACK OPTION
; CLEAR QINT AFTER LOOPBACK ENABLED
; CLEAR ERROR FLAG

```


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

```

1686 .SBTTL TEST24: CHECK QVAD (U-REGISTER) (MOVING 1)
1687 026742 STATST <CHECK QVAD (U-REGISTER) (MOVING 1)>
1688 026742 040 103 110 T24MSG: .ASCIZ / CHECK QVAD (U-REGISTER) (MOVING 1)/
1689
1690 ;*****
1691 ;**
1692 ;TEST DESCRIPTION
1693 ; CHECKS THAT QVAD (C-COUNTER) CAN BE LOADED AND READ
1694 ; USES LDQVADR - LDUVADR LOOP-BACK
1695 ; USES QTSOUT - QTSIN LOOP-BACK
1696 ;TEST STEPS-
1697 ; SET QMT1 FUNC = 3 (SELUVADR)
1698 ; LOAD WORD IN QMT0
1699 ; SET QMT1 MVADR BIT
1700 ; READ QMT0, CHECK
1701 ; CLEAR QMT1 MVADR BIT
1702 ; MOVE BIT
1703 ; REPEAT FOR ALL BITS
1704 ;
1705 ;--
1706 ;*****
1707 027006 BGNTST
1708 027006 T24::
1709 027006 CKPNT
1710 027010 104421 001000 TRAP C0RFLA ; PRINT OUT TEST TITLE IF PNT SET
1711 027014 001012 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1712 027016 012746 026742 BNE 644 ; BRANCH IF NOT SET
1713 027022 012746 002270 MOV #T24MSG,-(SP)
1714 027026 012746 000002 MOV #PNTMSG,-(SP)
1715 027032 010600 MOV #2,-(SP)
1716 027034 104417 MOV SP,R0
1717 027036 062706 000006 TRAP C0PNTF
1718 027042 005037 002532 ADD #6,SP
1719 027046 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
1720 027046 012777 000100 153242 INIT
1721 027054 013777 002520 153234 ;INITIALIZE DEVICE
1722 027062 005077 153212 MOV #QHALT,QMT1 ;INITIALIZE Q-BRIDGE
1723 027066 005037 002532 MOV INITWORD,QMT1 ;SET UP LOOPBACK OPTION
1724 027072 012701 000012 CLR QINT ;CLEAR QINT AFTER LOOPBACK ENABLED
1725 027076 052777 000014 153212 CLR ERROR1 ;CLEAR ERROR FLAG
1726 027104 012702 000020 000001 MOV #10,R1 ; REGISTER NUMBER TO R1
1727 027110 012737 000001 002530 BIS #SELUVADR,QMT1 ; SELECT QMT1 FUNCTION = 3
1728 027116 012737 000001 002530 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1729 027122 013777 002530 153164 MOV #BIT0,EXP ; FIRST DATA PATTERN TO CHECK
1730 027130 052777 000002 153160 ;
1731 027136 017737 153152 002526 ; TEST BITS 0-15 WITH MOVING ONE
1732 027144 042777 000002 153144 ;
1733 027116 027116 BGNSUB
1734 027116 104402 T24.1:
1735 027120 104404 104: TRAP C0BSUB
1736 027122 013777 002530 153164 TRAP C0BSEG
1737 027130 052777 000002 153160 MOV EXP,QMT0 ; MOVE EXPECTED TO REGISTER QVAD
1738 027136 017737 153152 002526 BIS #MVADR,QMT1 ; SET MVADR BIT IN QMT1
1739 027144 042777 000002 153144 MOV QMT0,VAL ; READ QVAD AND MOVE TO VALUE
1740 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        204          ; IF OK, BRANCH
1725                                     ;
1726                                     ; ERROR CONDITION
1727                                     ;
1728 027162 005737 002532      TST      ERROR1      ; SEE IF FIRST ERROR
1729 027166 001005                      BNE        154          ; SKIP HEADER IF ALREADY PRINTED
1730 027170                      ERRMRD    1,MSG30,ERR7 ; TELL OPERATOR ERROR
      027170 104456      TRAP      C#ERRMRD
      027172 000001      .WORD     1
      027174 006471      .WORD     MSG30
      027176 012246      .WORD     ERR7
1731 027200 000402                      BR         184          ; SKIP DETAIL LINE IF HEADER PRINTED
1732 027202 004737 012366      154:      CALL     DETAIL ; PRINT DETAIL LINE
1733 027206 005237 002532      184:      INC      ERROR1 ; INCREMENT ERROR COUNT
1734                                     ;
1735                                     ; END ERROR CONDITION
1736                                     ;
1737 027212                      204:      ENDSEG
      027212                      100004:
      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      104      ; 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      5004      ; 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                      5004:
1746 027254                      L10103:
      027254 104401      TRAP      C#ETST

```

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

```

1748 .SBTTL TEST25: CHECK QVAD (U-REGISTER) (MOVING 0)
1749 027256 STATST <CHECK QVAD (U-REGISTER) (MOVING 0)>
1750 027256 040 103 110 T25MSG:: .ASCIZ / CHECK QVAD (U-REGISTER) (MOVING 0)/
1751 ;*****
1752 ;**
1753 ;TEST DESCRIPTION
1754 ; CHECKS THAT QVAD (C-COUNTER) CAN BE LOADED AND READ
1755 ; USES LDQVADR - LDUVADR LOOP-BACK
1756 ; USES QTSOUT - QTSIN LOOP-BACK
1757 ;TEST STEPS-
1758 ; SET QMT1 FUNC = 3 (SELUVADR)
1759 ; LOAD WORD IN QMT0
1760 ; SET QMT1 MVADR BIT
1761 ; READ QMT0, CHECK
1762 ; CLEAR QMT1 MVADR BIT
1763 ; MOVE BIT
1764 ; REPEAT FOR ALL BITS
1765 ;
1766 ;--
1767 ;*****
1768
1769 027322 BGNTST
1770 027322 T25::
1771 027322 CKPNT
1772 027322 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
1773 027324 104421 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1774 027330 001012 BNE 64# ; BRANCH IF NOT SET
1775 027332 012746 027256 MOV #T25MSG,-(SP)
1776 027336 012746 002270 MOV #PNTMSG,-(SP)
1777 027342 012746 000002 MOV #2,-(SP)
1778 027346 010600 MOV SP,R0
1779 027350 104417 TRAP C#PNTF
1780 027352 062706 000006 ADD #6,SP
1781 027356 005037 002532 64# CLR ERROR1 ; CLEAR ERROR COUNTER
1782 027362 INIT
1783 ;INITIALIZE DEVICE
1784 027362 012777 000100 152726 MOV #QHALT,QMT1 ; INITIALIZE Q-BRIDGE
1785 027370 013777 002520 152720 MOV INITWORD,QMT1 ; SET UP LOOPBACK OPTION
1786 027376 005077 152676 CLR #QINT ; CLEAR QINT AFTER LOOPBACK ENABLED
1787 027402 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
1788 1772 027406 012701 000012 MOV #10,R1 ; REGISTER NUMBER TO R1
1789 1773 027412 012702 000020 MOV #16,R2 ; NUMBER OF BITS TO CHECK
1790 1774 027416 012703 000001 MOV #BIT0,R3 ; INITIAL DATA PATTERN TO R3
1791 1775 027422 010337 002530 MOV R3,EXP ; MOVE BIT PATTERN TO EXPECTED
1792 1776 027426 005137 002530 COM EXP ; COMPLEMENT FOR MOVING 0
1793 1777 027432 052777 000014 152656 BIS #SELUVADR,QMT1 ; SELECT FUNCTION 3 IN QMT1
1794 1778
1795 1779 ; TEST BITS 0-15 WITH MOVING ZERO
1796 1780
1797 1781 027440 BGNSUB
1798 027440 T25.1:
1799 027440 104402 TRAP C#BSUB
1800 1782 027442 104404 BGNSEG
1801 027442 104404 TRAP C#BSEG
1802 1783 027444 013777 002530 152642 MOV EXP,QMT0 ; MOVE EXPECTED TO REGISTER QVAD
1803 1784 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	QVAD,VAL	; MOVE CONTENTS OF QVAD TO VALUE
1786	027466	042777	000002	152622	BIC	QVADR,QVAD1	; CLEAR QVADR BIT IN QVAD1
1787	027474	023737	002530	002526	CMP	EXP,VAL	; COMPARE EXPECTED WITH VALUE
1788	027502	001414			BEQ	204	; IF OK, BRANCH
1789							
1790							
1791							
1792	027504	005737	002532		TST	ERROR1	; SEE IF FIRST ERROR
1793	027510	001005			BNE	154	; SKIP HEADER IF ALREADY PRINTED
1794	027512				ERRHRD	1,MSG31,ERR7	; TELL OPERATOR ERROR
	027512	104456			TRAP	C#ERRHRD	
	027514	000001			.WORD	1	
	027516	006532			.WORD	MSG31	
	027520	012246			.WORD	ERR7	
1795	027522	000402			BR	184	; SKIP DETAIL LINE IF HEADER PRINTED
1796	027524	004737	012366		CALL	DETAIL	; PRINT DETAIL LINE
1797	027530	005237	002532		INC	ERROR1	; INCREMENT ERROR COUNT
1798							
1799							
1800							
1801	027534						
	027534				204:	ENDSEG	
	027534	104405			100004:		
1802	027536	006303			TRAP	C#ESEG	
1803	027540	010337	002530		ASL	R3	; SHIFT BIT FOR NEXT TEST
1804	027544	005137	002530		MOV	R3,EXP	; PLACE PATTERN IN EXPECTED
1805	027550	005302			COM	EXP	; COMPLIMENT FOR MOVING 0
1806	027552	001333			DEC	R2	; DECREMENT LOOP COUNTER
1807	027554				BNE	104	; IF NOT DONE, BRANCH BACK
	027554				ENDSUB		
	027554	104403			L10106:		
1808	027556	005737	002532		TRAP	C#ESUB	
1809	027562	001410			TST	ERROR1	; ANY ERRORS
1810	027564				BEQ	5004	; BRANCH IF NO ERRORS
	027564	012746	003333		PRINTX	#FOR0	; LINE FEED
	027570	012746	000001		MOV	#FOR0,-(SP)	
	027574	010600			MOV	#1,-(SP)	
	027576	104415			MOV	SP,R0	
	027600	062706	000004		TRAP	C#PNTX	
1811	027604				ADD	#4,SP	
1812	027604				5004:		
	027604				L10105:	ENDTST	
	027604	104401			TRAP	C#ETST	

TEST26: CHECK QVAD INCREMENT (Q-REGISTER)

```

1814 .SBTTL TEST26: CHECK QVAD INCREMENT (Q-REGISTER)
1815 027606 STATST <CHECK QVAD INCREMENT (Q-REGISTER)>
1816 027606 040 103 110 T26MSG:: .ASCIZ / CHECK QVAD INCREMENT (Q-REGISTER)/
1817 ;*****
1818 ;**
1819 ;TEST DESCRIPTION
1820 ; CHECKS THAT QVAD (A-COUNTER) INCREMENT ON READ QVDA
1821 ;TEST STEPS-
1822 ; LOAD QVAD WITH 1'S
1823 ; READ QVDA
1824 ; READ QVAD, CHECK FOR 0
1825 ;
1826 ;--
1827 ;*****
1828
1829 027652 BGNTST
1830 027652 T26::
1831 027652 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
1832 027652 TRAP C:RFLA
1833 027654 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
1834 027660 001012 BNE 64# ; BRANCH IF NOT SET
1835 027662 012746 027606 MOV #T26MSG,-(SP)
1836 027666 012746 002270 MOV #PNTMSG,-(SP)
1837 027672 012746 000002 MOV #2,-(SP)
1838 027676 010600 MOV SP,R0
1839 027700 104417 TRAP C:PNTF
1840 027702 062706 000006 ADD #6,SP
1841 027706 005037 002532 64#: CLR ERROR1 ; CLEAR ERROR COUNTER
1842 ;
1843 ; INITIALIZE DEVICE
1844 ;
1845 027712 012777 000100 152376 MOV #MQHALT,BQMT1 ; INITIALIZE Q-BRIDGE
1846 027720 012701 000012 MOV #10.,R1 ; REGISTER NUMBER TO R1
1847 ;
1848 ; TEST FOR INCREMENT
1849 ;
1850 027724 BGNSUB
1851 027724 T26.1:
1852 027724 104402 TRAP C:BSUB
1853 027726 052777 000020 152346 BIS #QVREN,BQCTL ; READ ENABLE
1854 027734 012777 177777 152356 MOV #1,BQVAD ; MOVE ALL ONE'S TO QVAD
1855 027742 017702 152354 MOV BQVDA,R2 ; DUMMY READ TO QVDA
1856 027746 012737 000000 002530 MOV #0,EXP ; CLEAR EXPECTED FIELD
1857 027754 052777 000002 152334 BIS #MVADR,BQMT1 ; ENABLE TO READ QVAD
1858 027762 017737 152332 002526 MOV BQVAD,VAL ; MOVE CURRENT VALUE OF QVAD TO VAL
1859 027770 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
1860 027776 001406 BEQ 1# ; IF OK, BRANCH
1861 ;
1862 ; ERROR CONDITION
1863 ;
1864 030000 ERRHRD 1,MSG32,ERR4 ; TELL OPERATOR ERROR
1865 030000 104456 TRAP C:ERHRD
1866 030002 000001 .WORD 1
1867 030004 006573 .WORD MSG32
1868 030006 011776 .WORD ERR4
1869 030010 004737 013230 JSR PC,DUMP ; DO OPTIONAL REGISTER DUMP

```

TEST26: CHECK QVAD INCREMENT (Q-REGISTER)

1853	030014		18:		
1854	030014			ENDSUB	
	030014		L10110:		
	030014	104403		TRAP	C#ESUB
1855	030016			ENDTST	
	030016		L10107:		
	030016	104401		TRAP	C#ETST

TEST27: CHECK QVAD INCREMENT (U-REGISTER)

```

1857 .SBTTL TEST27: CHECK QVAD INCREMENT (U-REGISTER)
1858 030020 STATST <CHECK QVAD INCREMENT (U-REGISTER)>
      030020 040 103 110 T27MSG:: .ASCIZ / CHECK QVAD INCREMENT (U-REGISTER)/
1859
1860 ;*****
1861 ;**
1862 ;TEST DESCRIPTION-
1863 ; CHECKS THAT QVAD (C-COUNTER) INCREMENT ON LDUVDAT
1864 ; USES LDQVADR - LDUVADR LOOP-BACK
1865 ; USES QTSOUT - QTSIN LOOP-BACK
1866 ;TEST STEPS-
1867 ; SET QMT1 FUNC = 3 (SELUVADR)
1868 ; LOAD QMT0 WITH 1'S
1869 ; SET QMT1 FUNC = 6 (SEL UV DAT)
1870 ; LOAD QMT0 (ANY VALUE)
1871 ; SET QMT1 MVADR BIT
1872 ; SET QMT1 FUNC = 3 (SELUVADR)
1873 ; READ QMT0, CHECK FOR 0
1874 ;
1875 ;--
1876 ;*****
1877
1878 030064 BGNTST
      030064
1879 030064 T27:: CKPNT
      030064 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
      030066 022700 001000 CMP #PNT,RO ; 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,RO
      030112 104417 TRAP C#PNTF
      030114 062706 000006 ADD #6,SP
      030120 005037 002532 64#: CLR ERROR1 ; CLEAR ERROR COUNTER
1880 030124 INIT
      030124 ;INITIALIZE DEVICE
      030124 MOV #MHALT, BQMT1 ;INITIALIZE Q-BRIDGE
      030132 013777 002520 152164 MOV INITWORD, BQMT1 ;SET UP LOOPBACK OPTION
      030140 005077 152134 152156 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
      030144 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
1881 030150 012701 000012 MOV #10., R1 ; REGISTER NUMBER TO R1
1882
1883 ;
1884 ; CHECK QVAD INCREMENT
1885 ;
      030154 BGNSUB
      030154
1886 030154 T27.1: TRAP C#BSUB
      030154 104402 BIS #SELUVADR, BQMT1 ; SELECT QMT1 FUNC = 3 (SELUVADR)
1886 030156 052777 000014 152132 MOV #-1, BQMT0 ; LOAD QMT0 WITH ALL ZERO'S
1887 030164 012777 177777 152122 BIC #SELUVADR, BQMT1 ; DESELECT QMT1 FUNC = 3 (SELUVADR)
1888 030172 042777 000014 152116 BIS #SELUVADR, BQMT1 ; SELECT QMT1 FUNC = 3 (SELUVADR)
1889 030200 052777 000030 152110 BIS #SELUVADR, BQMT1 ; SELECT QMT1 FUNC = 6 (SELUVADR)
1890 030206 012777 177777 152100 MOV #-1, BQMT0 ; LOAD QMT0 (ANY VALUE)
1891 030214 042777 000030 152074 BIC #SELUVADR, BQMT1 ; DESELECT QMT1 FUNC = 6 (SELUVADR)
1892 030222 052777 000016 152066 BIS #SELUVADR, BQMT1 ; SET QMT1 = FUNC 3 (SELUVADR) & MVADR BIT
1893 030230 012737 000000 002530 MOV #0, EXP ; LOAD EXP WITH EXPECTED VALUE
1894 030236 017737 152052 002526 MOV BQMT0, VAL ; READ QMT0, 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	18	: IF OK, BRANCH
1898							
1899							
1900							
1901	030262						
	030262	104456			ERRMRD	1,MSG33,ERR4	: TELL OPERATOR ERROR
	030264	000001			TRAP	C#ERRMRD	
	030266	006633			.WORD	1	
	030270	011776			.WORD	MSG33	
1902	030272	004737	013230		.WORD	ERR4	
1903	030276				JSR	PC,DUMP	: DO OPTIONAL REGISTER DUMP
1904	030276						
	030276				18:		
	030276	104403			ENDSUB		
1905	030300				L10112:		
	030300				TRAP	C#ESUB	
	030300	104401			ENDTST		
					L10111:		
					TRAP	C#ETST	

TEST28: CHECK VIEW REGISTER RAM

```

1907 .SBTTL TEST28: CHECK VIEW REGISTER RAM
1908 030302 STATST <CHECK VIEW REGISTER RAM>
      030302 040 103 110 T28MSG:: .ASCIZ / CHECK VIEW REGISTER RAM/

1909 ;*****
1910 ;**
1911 ;TEST DESCRIPTION-
1912 ; CHECKS EACH VIEW DATA RAM LOCATION WITH 1'S AND 0'S
1913 ; CHECKS ALL LOCATIONS FOR NO CHANGE
1914 ; USES QVAD (Q-REGISTER) FOR READ ADDRESS
1915 ; USED QVAD (U-REGISTER) FOR WRITE ADDRESS
1916 ; USES LUQVAD - LDUVAD LOOP-BACK
1917 ; USES QTSOUT - QTSIN LOOP-BACK
1918 ;TEST STEPS-
1919
1920 ; SET MIODIS (LOOPBACK ENABLE BIT)
1921 ; SELECT FUNCTION = 3 UVADR
1922 ; LOAD QVAD (UVADR) WITH ADDRESS (START AT 0)
1923 ; DESELECT FUNCTION 3
1924 ; SELECT FUNCTION = 6 UVDAT
1925 ; LOAD QMT0 WITH 1'S (QVDA WRITE)
1926 ; CLEAR MIODIS (LOOPBACK ENABLE BIT) AND FUNCTION 6
1927 ; LOAD QVAD (QVADR) WITH ADDRESS
1928 ; READ QVDA, CHECK FOR ALL 1'S
1929
1930 ; SET MIODIS (LOOPBACK ENABLE BIT)
1931 ; SELECT FUNCTION = 3 UVADR
1932 ; LOAD QVAD (UVADR) WITH ADDRESS
1933 ; DESELECT FUNCTION 3
1934 ; SELECT FUNCTION = 6 UVDAT
1935 ; LOAD QMT0 WITH 0'S (QVDA WRITE)
1936 ; SELECT FUNCTION = 3 UVADR
1937 ; SET MVADR BIT (QVAD DIRECTION BIT) IN QMT1
1938 ; READ QMT0 (UVADR) ADDRESS, CHECK FOR INCREMENT
1939 ; CLEAR MVADR BIT IN QMT1
1940 ; CLEAR MIODIS (LOOPBACK ENABLE BIT) AND FUNCTION 6
1941 ; LOAD QVAD (QVADR) WITH ADDRESS
1942 ; READ QVDA, CHECK FOR ALL 0'S
1943 ; SET MVADR BIT (QVAD DIRECTION BIT) IN QMT1
1944 ; READ QVAD, CHECK FOR ADDRESS INCREMENTS
1945 ; CLEAR QMT1 MVADR BIT
1946 ; REPEAT FOR 2K ADDRESSES (RAM ADDRESS SHOULD BE 0 AT END)
1947
1948 ;
1949 ;--
1950 ;*****
1951
1952 030334 BGNTST
      030334 T28::
1953 030334 CKPNT
      030334 TRAP C0RFLA ; PRINT OUT TEST TITLE IF PNT SET
      030336 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
      030342 001012 BNE 641 ; 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 C0PNTF

```


TEST28: CHECK VIEW REGISTER RAM

1954	030364	062706	000006		641:	ADD	#6,SP		
1955	030370	005037	002532			CLR	ERROR1		; CLEAR ERROR COUNTER
1956									
1957	030374	012777	000100	151714		MOV	#MOMALT,8QMT1		; 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									
1964									
1965	030424								
	030424								
	030424	104402			T28.1:	TRAP	C1BSUB		
1966	030426				L28:				; START LOOPS HERE
1967	030426					BGNSEG			
	030426	104404				TRAP	C1BSEG		
1968									
1969									
1970									
1971									
1972	030430	013777	002520	151660		MOV	INITWORD,8QMT1		; SET UP LOOPBACK OPTION
1973	030436	005077	151636			CLR	8QINT		; CLEAR QINT AFTER LOOPBACK ENABLED
1974	030442	042777	000020	151632		BIC	#QVREN,8QCTL		; CLEAR READ ENABLE
1975	030450	052777	000014	151640		BIS	#SELUVADR,8QMT1		; SELECT UVADR
1976	030456	013777	002524	151630		MOV	ADD,8QMT0		; LOAD U SIDE OF QVAD WITH RAM ADDRESS
1977									
1978									
1979									
1980	030464	042777	000014	151624		BIC	#SELUVADR,8QMT1		; DESELECT UVADR
1981	030472	052777	000030	151616		BIS	#SELUVDAT,8QMT1		; SELECT UVDAT TO WRITE TO RAM
1982									
1983									
1984									
1985	030500	012737	177777	002530		MOV	#-1,EXP		; EXPECTED PATTEN TO EXP
1986	030506	013777	002530	151600		MOV	EXP,8QMT0		; MOVE ALL 1'S TO QVDA THROUGH QMT0
1987									
1988									
1989									
1990	030514	052777	000020	151560		BIS	#QVREN,8QCTL		; READ ENABLE
1991	030522	012777	000000	151566		MOV	#0,8QMT1		; CLEAR MAINTENANCE LOOP-BACK BIT
1992	030530	013777	002524	151562		MOV	ADD,8QVAD		; LOAD QVAD WITH RAM ADDRESS
1993									
1994									
1995									
1996	030536	017737	151560	002526		MOV	8QVDA,VAL		; READ QVDA
1997	030544	023737	002530	002526		CMP	EXP,VAL		; COMPARE EXPECTED WITH VALUE
1998	030552	001433				BEQ	204		; IF OK, BRANCH
1999									
2000									
2001									
2002	030554	005737	002532			TST	ERROR1		; CHECK FOR PREVIOUS ERRORS
2003	030560	001004				BNE	104		; BRANCH IF PREVIOUS ERRORS
2004	030562					ERRHRD	1,MSG34,ERR5		; PRINT ERROR HEADER
	030562	104456				TRAP	C1ERRHRD		

TEST28: CHECK VIEW REGISTER RAM

	030564	000001			.WORD	1		
	030566	006673			.WORD	MSG34		
	030570	012136			.WORD	ERR5		
2005	030572	022737	000023	002532	104:	CMP	#19.,ERROR1	; SEE IF TOO MANY TO PRINT
2006	030600	100416				BMI	154	; 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,RO	
	030630	104414				TRAP	C#PNTB	
	030632	062706	000012			ADD	#12,SP	
2008	030636	005237	002532		154:	INC	ERROR1	; INCREMENT ERROR OCCURRED COUNTER
2009	030642				204:			
2010								
2011								
2012								
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,@QMT0	; LOAD U SIDE OF QVAD WITH RAM ADDRESS
2019								
2020								
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								
2026								
2027	030712	012737	000000	002530		MOV	#0,EXP	; LOAD ALL 0'S TO EXP
2028	030720	013777	002530	151366		MOV	EXP,@QMT0	; MOVE ALL 0'S TO QVDA THROUGH QMT0
2029								
2030								
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	@QMT0,VAL	; READ U ADDRESS IN QVAD FROM QMT0
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	1704	; IF OK, BRANCH
2042								
2043								
2044								
2045	031014	005737	002532			TST	ERROR1	; CHECK FOR PREVIOUS ERROR FROM ABOVE
2046	031020	001007				BNE	1504	; IF ERROR FROM ABOVE, BRANCH
2047	031022	005737	002534			TST	ERROR2	; CHECK FOR PREVIOUS ERROR
2048	031026	001004				BNE	1504	; BRANCH IF PREVIOUS ERRORS
2049	031030					ERRHRD	2,MSG35	
	031030	104456				TRAP	C#ERRHRD	

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,BQCTL ; READ ENABLE
2060 031112 012777 000000 151176 MOV #0,BQMT1 ; CLEAR MAINTENANCE LOOP-BACK BIT
2061 031120 013777 002524 151172 MOV ADD,BQVAD ; 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 BQVDA,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#ERRHRD
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,BQMT1 ; 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      614      ; 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      504      ; IF ERROR FROM ABOVE, BRANCH
2096 031312 005737 002534      TST      ERROR2      ; CHECK FOR PREVIOUS ERROR
2097 031316 001004      BNE      504      ; BRANCH IF PREVIOUS ERRORS
2098 031320      ERRHRD 4,MSG35
      031320 104456      TRAP      C@ERRHRD
      031322 000004      .WORD      4
      031324 006736      .WORD      MSG35
      031326 000000      .WORD      0
2099 031330 022737 000011 002534 504:  CMP      #9,ERROR2      ; SEE IF TOO MANY TO PRINT
2100 031336 100414      RMI      604      ; 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      604:  INC      ERROR2      ; INCREMENT ERROR COUNTER
2103
2104 031374      614:
2105 031374      ENDSEG
      031374      100004:
      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      754      ; 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      754:
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      804      ; BRANCH IF NO ERRORS
2121
2122      ; PRINT FINAL ERROR TOTALS
2123
2124 031426      PRINTB  #FOR6,ERROR1 ; ERROR ONE 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, RO	
	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, RO	
	031510	104414		TRAP	C#PNTB	
	031512	062706	000004	ADD	#4, SP	
2127	031516			ENDSUB		
	031516					
	031516	104403		TRAP	C#ESUB	
2128	031520			ENDTST		
	031520					
	031520	104401		TRAP	C#ETST	

804:

L10114:

L10113:

TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```

2130 .SBTTL TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)
2131 031522 STATST <CHECK RAM ADDRESSING (ADDRESS VALUE)>
      031522 040 103 110 T29MSG: .ASCIZ / CHECK RAM ADDRESSING (ADDRESS VALUE)/
2132 ;*****
2133 ;**
2134 ;TEST DESCRIPTION-
2135 ; CHECKS VIEW DATA RAM ADDRESSING
2136 ; LOADS ALL RAM LOCATIONS WITH ADDRESS
2137 ; CHECKS ALL RAM LOCATIONS
2138 ; USES QVAD (Q-REGISTER) FOR ADDRESS
2139 ; USES LDQVAD - LDQVAD LOOP-BACK
2140 ; USES QTSOUT - QTSIN LOOP-BACK
2141 ;TEST STEPS-
2142 ; SELECT FUNCTION UVADR
2143 ; LOAD QVAD WITH ZERO'S THROUGH QMTO
2144 ; DESELECT FUNCTION UVADR
2145 ; SELECT FUNCTION UVDAT
2146 ; LOAD QVDA WITH ADDRESS VALUE THROUGH QMTO
2147 ; REPEAT LOAD QVDA THROUGH QMTO FOR 2K ADDRESSES
2148 ; CLEAR MIODIS MAINTENANCE LOOP-BACK BIT
2149 ; LOAD QVAD WITH ZERO
2150 ; READ QVDA, CHECK FOR ZERO OR 2000 (CONFIRMS 1K OR 2K)
2151 ; TELL OPERATOR 1K RAM, 2K RAM OR UNKNOWN SIZE RAM
2152 ; READ QVDA, CHECK DATA
2153 ; REPEAT READ QVDA FOR ALL RAM
2154 ;
2155 ;
2156 ;--
2157 ;*****
2158
2159 031570 BGNTST
2160 031570 T29:
      031570 CKPNT
      031572 104421 TRAP C:RFLA ; PRINT OUT TEST TITLE IF PNT SET
      031576 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
      031600 012746 031522 BNE 64: ; BRANCH IF NOT SET
      031604 012746 002270 MOV #T29MSG,-(SP)
      031610 012746 000002 MOV #PNTMSG,-(SP)
      031614 010600 MOV #2,-(SP)
      031616 104417 MOV SP,R0
      031620 062706 000006 TRAP C:PNTF
      031624 005037 002532 ADD #6,SP
      2161 031630 64: CLR ERROR1 ; CLEAR ERROR COUNTER
      ;INITIALIZE DEVICE
      031630 012777 000100 150460 MOV #QHALT,QMTO1 ;INITIALIZE Q-BRIDGE
      031636 013777 002520 150452 MOV INITWORD,QMTO1 ;SET UP LOOPBACK OPTION
      031644 005077 150430 CLR QINT ;CLEAR QINT AFTER LOOPBACK ENABLED
      031650 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
      2162 031654 012701 000000 MOV #0,R1 ; CLEAR R1 AS COUNTER
2163 ;
2164 ; SET ADDRESS
2165 ;
2166 031660 042777 000020 150414 BIC #QVREN,QCTL ; CLEAR READ ENABLE
2167 031666 052777 000014 150422 BIS #SELUVADR,QMTO1 ; SELECT UVADR
2168 031674 012777 000000 150412 MOV #0,QMTO ; LOAD U SIDE OF QVAD WITH RAM ADDRESS
2169 ;

```


TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```

2170 ; SET UP FOR WRITE TO RAM
2171 ;
2172 031702 042777 000014 150406 BIC #SELUVADR,BQMT1 ; DESELECT UVADR
2173 031710 052777 000030 150400 BIS #SELUVDAT,BQMT1 ; SELECT UVDAT TO WRITE TO RAM
2174 ;
2175 ; LOAD RAM WITH ADDRESSES
2176 ;
2177 031716 101:
2178 031716 010177 150372 MOV R1,BQMT0 ; LOAD QVDA WITH ADDRESS DATA THROUGH QMT0
2179 031722 005201 INC R1 ; INCREMENT COUNTER
2180 031724 020127 004000 CMP R1,#4000 ; HAS IT LOOPED 2K YET?
2181 031730 001372 BNE 101 ; IF NOT DONE, GO AGAIN
2182 ;
2183 ; ALL RAM LOADED, NOW CHECK FOR 1K OR 2K PRESENT
2184 ;
2185 031732 042777 000200 150356 BIC #M10DIS,BQMT1 ; CLEAR MAINTENANCE LOOP-BACK BIT
2186 031740 012737 000002 002530 MOV #2,EXP ; AMOUNT OF RAM TO EXP FOR MESSAGE
2187 031746 012701 000000 MOV #0,R1 ; SET COUNTER FOR 2K RAM
2188 031752 012737 000000 002524 MOV #0,ADD ; POINT ADD TO FIRST RAM ADDRESS
2189 ;
2190 ; SET ADDRESS
2191 ;
2192 031760 052777 000020 150314 BIS #QVREN,BQCTL ; READ ENABLE
2193 031766 012777 000000 150324 MOV #0,BQVAD ; LOAD QVAD WITH RAM ADDRESS
2194 031774 017737 150322 002526 MOV BQVDA,VAL ; READ FIRST RAM ADDRESS
2195 032002 022737 000000 002526 CMP #0,VAL ; IS IT 2K RAM
2196 032010 001450 BEQ 201 ; BRANCH IF 2K RAM
2197 032012 012737 000001 002530 MOV #1,EXP ; ASSUME 1K RAM FOR MESSAGES
2198 032020 012701 002000 MOV #2000,R1 ; LOAD R1 COUNTER WITH 1K
2199 032024 022737 002000 002526 CMP #2000,VAL ; IS IT 1K RAM
2200 032032 001437 BEQ 201 ; BRANCH IF 1K RAM
2201 ;
2202 ; ERROR CONDITION, UNKNOWN RAM SIZE
2203 ; ASSUME 1K FOR REST OF TEST
2204 ;
2205 032034 005237 002532 INC ERROR1 ; INCREMENT ERROR COUNTER
2206 032040 ERRHRD 1,MSG34,ERR5 ; PRINT ERROR HEADER
2207 032040 104456 TRAP C1ERRHRD
2208 032042 000001 .WORD 1
2209 032044 006673 .WORD MSG34
2210 032046 012136 .WORD ERR5
2211 ;
2212 ; TELL OPERATOR UNKNOWN RAM SIZE
2213 ;
2214 PRINTB #FOR4.5 ; PRINT UNKNOWN RAM SIZE MESSAGE
2215 MOV #FOR4.5,-(SP)
2216 MOV #1,-(SP)
2217 MOV SP,R0
2218 TRAP C1PNTB
2219 ADD #4,SP
2220 MOV R1,EXP ; EXPECTED VALUE FROM R1
2221 PRINTB #FOR4,ADD,VAL,EXP ; PRINT ERROR LINE
2222 MOV EXP,-(SP)
2223 MOV VAL,-(SP)
2224 MOV ADD,-(SP)
2225 MOV #FOR4,-(SP)
2226 MOV #4,-(SP)

```

TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```

2213 032120 010600      MOV      SP,R0
2214 032122 104414      TRAP     C:PNTB
2215 032124 062706 000012 ADD     #12,SP
2216 032130 000417      BR       40$
2217
2218 032132 005737 002536 ; END ERROR CONDITION
2219 032136 001014      ; PRINT RAM SIZE IF DETERMINED AND NOT PREVIOUSLY PRINTED
2220 032140
2221 032140 013746 002530 20$: TST      RAM      ; SEE IF RAM SIZE ALREADY PRINTED
2222 032144 012746 004227      BNE     40$      ; IF RAM SIZE PRINTED, BRANCH
2223 032150 012746 000002      PRINTB  #FOR9,EXP  ; TELL OPERATOR HOW MUCH RAM PRESENT
2224 032154 010600
2225 032156 104414
2226 032160 062706 000006      MOV     EXP,-(SP)
2227 032164 005237 002536      MOV     #FOR9,-(SP)
2228 032170
2229 032170 005201
2230 032172 005237 002524      MOV     #2,-(SP)
2231 032176 017737 150120 002526      MOV     SP,R0
2232 032204 010137 002530      TRAP     C:PNTB
2233 032210 023737 002530 002526      ADD     #6,SP
2234 032216 001433      INC      RAM      ; SET RAM FLAG MESSAGE PRINTED
2235
2236 032220 005737 002532      ; NOW LOOP THROUGH AND CHECK REST OF RAM
2237 032224 001004
2238
2239 032226
2240 032226 104456      40$: INC      R1      ; INCREMENT COUNTER
2241 032230 000002      INC      ADD      ; INCREMENT RAM ADDRESS POINTER
2242 032232 006673      MOV     $QVDA,VAL  ; READ NEXT RAM ADDRESS
2243 032234 012136      MOV     R1,EXP  ; GET EXPECTED IN EXP
2244 032236 022737 000023 002532 45$: CMP     EXP,VAL  ; COMPARE EXPECTED TO VALUE
2245 032244 100416      BEQ     50$      ; IF OK, BRANCH
2246 032246
2247
2248 032246 013746 002530      ; ERROR CONDITION
2249 032252 013746 002526
2250 032256 013746 002524
2251 032262 012746 003552
2252 032266 012746 000004
2253 032272 010600
2254 032274 104414
2255 032276 062706 000012
2256 032302 005237 002532      TST      ERROR1  ; HAVE ANY ERRORS OCCURED YET?
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500

```

TEST29: CHECK RAM ADDRESSING (ADDRESS VALUE)

```

2249 032306 020127 003777      CMP      R1,03777      ; END OF TEST
2250 032312 001326              BNE      604          ; 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      604          ; 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
      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              604:
2262 032366              L10115:
      032366 104401              TRAP     C#ETST

```


TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

```

2264 .SBTTL TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)
2265 032370 STATST <CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)>
2266 032370 040 103 110 T30MSG:: .ASCIZ / CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)/
2267 ;*****
2268 ;**
2269 ;TEST DESCRIPTION-
2270 ; CHECKS VIEW DATA RAM ADDRESSING
2271 ; LOADS ALL RAM LOCATIONS WITH ADDRESS
2272 ; CHECKS ALL RAM LOCATIONS
2273 ; USES QVAD (Q-REGISTER) FOR ADDRESS
2274 ; USES LDQVAD - LDQVAD LOOP-BACK
2275 ; USES QTSOUT - QTSIN LOOP-BACK
2276 ;TEST STEPS-
2277 ; SELECT FUNCTION UVADR
2278 ; LOAD QVAD WITH ZERO'S THROUGH QMTO
2279 ; DESELECT FUNCTION UVADR
2280 ; SELECT FUNCTION UVDAT
2281 ; LOAD QVDA WITH COMPLIMENTED ADDRESS VALUE THROUGH QMTO
2282 ; REPEAT LOAD QVDA THROUGH QMTO FOR 2K ADDRESSES
2283 ; CLEAR MIODIS MAINTENANCE LOOP-BACK BIT
2284 ; LOAD QVAD WITH ZERO
2285 ; READ QVDA, CHECK FOR ZERO OR 2000 (CONFIRMS 1K OR 2K)
2286 ; TELL OPERATOR 1K RAM, 2K RAM OR UNKNOWN SIZE RAM
2287 ; READ QVDA, CHECK DATA
2288 ; REPEAT READ QVDA FOR ALL RAM
2289 ;
2290 ;--
2291 ;*****
2292
2293 032452 BGNTST
2294 032452 T30::
2295 032452 CKPNT
2296 032452 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
2297 032454 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
2298 032460 001012 BNE 644 ; BRANCH IF NOT SET
2299 032462 012746 032370 MOV #T30MSG,-(SP)
2300 032466 012746 002270 MOV #PNTMSG,-(SP)
2301 032472 012746 000002 MOV #2,-(SP)
2302 032476 010600 MOV SP,R0
2303 032500 104417 TRAP C#PNTF
2304 032502 062706 000006 ADD #6,SP
2305 032506 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
2306 032512 INIT
2307 ;INITIALIZE DEVICE
2308 032512 012777 000100 147576 MOV #MGMHALT,@QMT1 ; INITIALIZE Q-BRIDGE
2309 032520 013777 002520 147570 MOV INITWORD,@QMT1 ; SET UP LOOPBACK OPTION
2310 032526 005077 147546 CLR @QINT ; CLEAR QINT AFTER LOOPBACK ENABLED
2311 032532 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
2312 032536 012701 177777 MOV #-1,R1 ; FIRST 1'S COMPLEMENT TO R1
2313 ;
2314 ; SET ADDRESS
2315 ;
2316 2300 032542 042777 000020 147532 BIC #QVREN,@QCTL ; CLEAR READ ENABLE
2317 2301 032550 052777 000014 147540 BIS #SELUVADR,@QMT1 ; SELECT UVADR
2318 2302 032556 012777 000000 147530 MOV #0,@QMT0 ; LOAD U SIDE OF QVAD WITH RAM ADDRESS
2319 2303 ;

```

TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

```

2304      ; SET UP FOR WRITE TO RAM
2305
2306 032564 042777 000014 147524      BIC      #SELUVADR,8QMT1      ; Deselect UVADR
2307 032572 052777 000030 147516      BIS      #SELUVDAT,8QMT1      ; Select UVDAT to write to RAM
2308
2309      ; LOAD RAM WITH ADDRESSES
2310
2311 032600      100:
2312 032600 010177 147510      MOV      R1,8QMT0      ; LOAD QVDA WITH ADDRESS DATA THROUGH QMT0
2313 032604 005301      DEC      R1      ; DECREMENT COUNTER
2314 032606 020127 173777      CMP      R1,#C4000      ; HAS IT LOOPED 2K YET?
2315 032612 001372      BNE      100      ; 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,8QCTL      ; READ ENABLE
2320 032622 042777 000200 147466      BIC      #MIODIS,8QMT1      ; 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      #0,8QVAD      ; LOAD QVAD WITH RAM ADDRESS
2328 032656 017737 147440 002526      MOV      8QVDA,VAL      ; READ FIRST RAM ADDRESS
2329 032664 022737 177777 000526      CMP      #C0,VAL      ; IS IT 2K RAM
2330 032672 001452      BEQ      200      ; 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      200      ; 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
2341      032722 104456      TRAP      C0ERRHRD
2342      032724 000001      .WORD      1
2343      032726 006673      .WORD      MSG34
2344      032730 012136      .WORD      ERR5
2345
2346      ; TELL OPERATOR UNKNOWN RAM SIZE
2347
2348 032732      PRINTB #FOR4.5      ; PRINT UNKNOWN RAM SIZE MESSAGE
2349 032732 012746 003600      MOV      #FOR4.5,-(SP)
2350 032736 012746 000001      MOV      #1,-(SP)
2351 032742 010600      MOV      SP,R0
2352 032744 104414      TRAP      C0PNTB
2353 032746 062706 000004      ADD      #4,SP
2354 032752 010137 002530      MOV      R1,EXP      ; EXPECTED VALUE FROM R1
2355 032756 005137 002530      COM      EXP      ; COMPLEMENT EXPRESSION
2356 032762      PRINTB #FOR4,ADD,VAL,EXP      ; PRINT ERROR LINE
2357 032762 013746 002530      MOV      EXP,-(SP)
2358 032766 013746 002526      MOV      VAL,-(SP)
2359 032772 013746 002524      MOV      ADD,-(SP)
2360 032776 012746 003552      MOV      #FOR4,-(SP)

```


TEST30: CHECK RAM ADDRESSING (COMPLEMENT ADDRESS VALUE)

```

033002 012746 000004      MOV      04, -(SP)
033006 010600      MOV      SP, R0
033010 104414      TRAP     C1PNTB
033012 062706 000012      ADD      012, SP
2348 033016 000417      BR      401
2349
2350      ; END ERROR CONDITION
2351      ; PRINT RAM SIZE IF DETERMINED AND NOT PREVIOUSLY PRINTED
2352
2353 033020 005737 002536 201:      TST      RAM      ; SEE IF RAM SIZE ALREADY PRINTED
2354 033024 001014      BNE      401      ; IF RAM SIZE PRINTED, BRANCH
2355 033026      PRINTB   0FOR9, EXP      ; TELL OPERATOR HOW MUCH RAM PRESENT
033026 013746 002530      MOV      EXP, -(SP)
033032 012746 004227      MOV      0FOR9, -(SP)
033036 012746 000002      MOV      02, -(SP)
033042 010600      MOV      SP, R0
033044 104414      TRAP     C1PNTB
033046 062706 000006      ADD      06, 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      401:
2361 033056 005201      INC      R1      ; INCREMENT COUNTER
2362 033060 005237 002524      INC      ADD      ; INCREMENT RAM ADDRESS POINTER
2363 033064 017737 147232 002526      MOV      0QVDA, VAL      ; READ NEXT RAM ADDRESS
2364 033072 010137 002530      MOV      R1, EXP      ; GET EXPECTED IN EXP
2365 033076 005137 002530      COM      EXP      ; COMPLEMENT EXPRESSION
2366 033102 023737 002530 002526      CMP      EXP, VAL      ; COMPARE EXPECTED TO VALUE
2367 033110 001432      BEQ      501      ; 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      ERRHRD   2, MSG34, ERR5      ; PRINT ERROR HEADER
033116 104456      TRAP     C1ERRRD
033120 000002      .WORD    2
033122 006673      .WORD    MSG34
033124 012136      .WORD    ERR5
2376 033126 022737 000023 002532 431:      CMP      019, ERROR1      ; SEE IF TOO MANY ERRORS TO PRINT
2377 033134 100416      BHI      481      ; BRANCH IF MORE THEN 20 PRINTED
2378 033136      PRINTB   0FOR4, 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      0FOR4, -(SP)
033156 012746 000004      MOV      04, -(SP)
033162 010600      MOV      SP, R0
033164 104414      TRAP     C1PNTB
033166 062706 000012      ADD      012, SP
2379 033172 005237 002532 481:      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			504:		
2384	033176	020127	003777		CMP	R1,03777 ; END OF TEST
2385	033202	001325			BNE	404 ; BRANCH IF TEST NOT DONE
2386						
2387						
2388						
2389	033204	005737	002532		TST	ERROR1 ; ANY ERRORS?
2390	033210	001422			BEQ	604 ; IF NO, BRANCH
2391						
2392						
2393						
2394	033212					
	033212	013746	002532		PRINTB	#FOR6,ERROR1 ; PRINT NUMBER OF ERRORS
	033216	012746	004054		MOV	ERROR1,-(SP)
	033222	012746	000002		MOV	#FOR6,-(SP)
	033226	010600			MOV	#2,-(SP)
	033230	104414			MOV	SP,R0
	033232	062706	000006		TRAP	C#PNTB
2395	033236				ADD	#6,SP
	033236	012746	003333		PRINTB	#FOR0 ; LINE FEED CARRIAGE RETURN
	033242	012746	000001		MOV	#FOR0,-(SP)
	033246	010600			MOV	#1,-(SP)
	033250	104414			MOV	SP,R0
	033252	062706	000004		TRAP	C#PNTB
2396	033256				ADD	#4,SP
2397	033256			604:		
	033256				ENDTST	
	033256	104401		L10116:		
					TRAP	C#ETST

TEST31: CHECK QBA INCREMENT

```

2399 .SBTTL TEST31: CHECK QBA INCREMENT
2400 033260 STATST <CHECK QBA INCREMENT>
      033260 040 103 110 T31MSG:: .ASCIZ / CHECK QBA INCREMENT/

2401 ;*****
2402 ;**
2403 ;TEST DESCRIPTION
2404 ; CHECKS BUS ADDRESS COUNTER INCREMENT
2405 ;TEST STEPS-
2406 ; LOAD 177776 IN QBA
2407 ; LOAD 1'S IN QDMA BITS 8-13
2408 ; IF U-BRIDGE THEN LOAD BITS 8,9
2409 ; SET QDMA GO AND DMAWT BITS
2410 ; READ QBA, CHECK FOR 0
2411 ; READ QDMA, CHECK BITS 8-13 FOR 0
2412 ; IF U-BRIDGE THEN CHECK BITS 8,9
2413 ;
2414 ;--
2415 ;*****
2416
2417
2418 033306 BGNTST
2419 033306 T31::
      033306 104421 CKPNT
      033310 022700 001000 TRAP C1RFLA ; PRINT OUT TEST TITLE IF PNT SET
      033314 001012 CMP #PNT,RO ; IS PNT FLAG SET?
      033316 012746 033260 BNE 641 ; BRANCH IF NOT SET
      033322 012746 002270 MOV #T31MSG,-(SP)
      033326 012746 000002 MOV #PNTMSG,-(SP)
      033332 010600 MOV #2,-(SP)
      033334 104417 MOV SP,RO
      033336 062706 000006 TRAP C1PNTF
      033342 005037 002532 ADD #6,SP
      641: CLR ERROR1 ; CLEAR ERROR COUNTER
      INIT
;INITIALIZE DEVICE
      033346 012777 000100 146742 MOV #MHALT,BGNT1 ;INITIALIZE Q-BRIDGE
      033354 013777 002520 146734 MOV INITWORD,BGNT1 ;SET UP LOOPBACK OPTION
      033362 005077 146712 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
      033366 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG

2421 ;
2422 ; CHECK QBA INCREMENT
2423 ;
2424 033372 012777 177776 146710 MOV #177776,BQBA ; ALL ONES TO QBA
2425 033400 012777 037400 146676 MOV #37400,BQDMA ; SET BITS 8-13 IN QDMA
2426 033406 005737 002500 TST BRIDGETYPE ; TEST BRIDGE TYPE
2427 033412 001403 BEQ 101 ; IF EQ THEN Q-BRIDGE
2428 033414 012777 001400 146662 MOV #01400,BQDMA ; SET BITS 8,9 IN QDMA (U-BRIDGE)
2429 033422 052777 000003 146654 101: BIS #DMAWT!GO,BQDMA ; INITIATE DMA TRANSFER
2430 033430 017737 146654 002526 MOV BQBA,VAL ; MOVE QBA TO VALUE
2431 033436 012737 000000 002530 MOV #0,EXP ; MOVE EXPECTED TO EXP
2432 033444 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
2433 033452 001410 BEQ 11 ; IF OK, BRANCH
2434 ;
2435 ; ERROR CONDITION
2436 ;
2437 033454 012701 000006 MOV #6,R1 ; REGISTER NUMBER TO R1
2438 033460 ERRMRD 1,MSG36,ERR4 ; TELL OPERATOR ERROR

```

```

033460 104456          TRAP      C$ERHRD
033462 000001          .WORD     1
033464 006777          .WORD     MSG36
033466 011776          .WORD     ERR4
2439 033470 004737 013230      JSR      PC,DUMP          ; DO OPTIONAL REGISTER DUMP
2440
2441          ; NOW CHECK QDMA BITS 8-13 FOR 0
2442          ;
2443 033474          10:
2444 033474 017737 146604 002526      MOV      @QDMA,VAL          ; READ QDMA INTO VAL
2445 033502 042737 140377 002526      BIC      @140377,VAL          ; CLEAR UNTESTED BITS
2446 033510 005737 002500          TST      BRIDGETYPE          ; TEST BRIDGE TYPE
2447 033514 001403          BEQ      110          ; IF EQ THEN Q-BRIDGE
2448 033516 042737 176377 002526      BIC      @176377,VAL          ; CLEAR UNTESTED BITS (U-BRIDGE)
2449 033524 012737 000000 002530 110:      MOV      @0,EXP          ; MOVE EXPECTED TO EXP
2450 033532 023737 002530 002526      CMP      EXP,VAL          ; COMPARE EXPECTED WITH VALUE
2451 033540 001410          BEQ      20          ; IF OK, BRANCH
2452
2453          ; ERROR CONDITION
2454          ;
2455 033542 012701 000004          MOV      @4,R1          ; REGISTER NUMBER TO R1
2456 033546          ERRHRD      2,MSG36.5,ERR4          ; TELL OPERATOR ERROR
          033546 104456          TRAP      C$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
2459 033562          20:
          033562          ENDTST
          033562 104401          L10117:      TRAP      C$ETST

```


TEST32: CHECK QWC OVERFLOW (READ)

```

2461 .SBTTL TEST32: CHECK QWC OVERFLOW (READ)
2462 033564 STATST <CHECK QWC OVERFLOW (READ)>
033564 T32MSG:: .ASCIZ / CHECK QWC OVERFLOW (READ)/

2463 ;*****
2464 ;**
2465 ;TEST DESCRIPTION-
2466 ; CHECKS THAT GO WITH DMAWT CAUSES IMMEDIATE DMA REQ
2467 ; CHECKS WC INCREMENT, OVERFLOW
2468 ;TEST STEPS-
2469 ; LOAD QWC WITH 1'S
2470 ; LOAD QBA WITH VALID ADDRESS
2471 ; SET QDMA AND GO BIT
2472 ; READ QWC, CHECK FOR 0
2473 ; CHECK QDMA RDY SET
2474 ;
2475 ;
2476 ;--
2477 ;*****
2478
2479 033620 BGNTST
033620 T32::
2480 033620 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
033620 104421 TRAP C0RFLA
033622 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
033626 001012 BNE 644 ; 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 C0PNTF
033650 062706 000006 ADD #6,SP
033654 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
2481 033660 INIT
; INITIALIZE DEVICE
033660 012777 000100 146430 MOV #MHALT,BQMT1 ; INITIALIZE Q-BRIDGE
033666 013777 002520 146422 MOV INITWORD,BQMT1 ; SET UP LOOPBACK OPTION
033674 005077 146400 CLR BQINT ; CLEAR QINT AFTER LOOPBACK ENABLED
033700 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
2482 033704 012777 177777 146400 MOV #1,BQWC ; LOAD QWC WITH 1'S
2483 033712 012777 041660 146370 MOV #BPATCH,BQBA ; LOAD QBA WITH VALID ADDRESS
2484 033720 052777 000001 146356 BIS #GO,BQDMA ; SET QDMA AND GO BIT
2485 033726 017737 146360 002526 MOV BQWC,VAL ; READ QWC
2486 033734 012737 000000 002530 MOV #0,EXP ; EXPECTED VALUE IN EXP
2487 033742 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
2488 033750 001410 BEQ 104 ; BRANCH IF OK
2489
2490 ; ERROR CONDITION
2491
2492 033752 012701 000007 MOV #7,R1 ; REGISTER NUMBER TO R1
2493 033756 ERRHRD 1,MSG40,ERR4 ; TELL OPERATOR ERROR
033756 104456 TRAP C0ERRRD
033760 000001 .WORD 1
033762 007337 .WORD MSG40
033764 011776 .WORD ERR4
2494 033766 004737 013230 JSR PC,DUMP ; DO OPTIONAL REGISTER DUMP
2495
2496 ; END ERROR CONDITION, CHECK FOR RDY SET

```

TEST32: CHECK QWC OVERFLOW (READ)

```

2497
2498 033772
2499 033772 017737 146306 002526
2500 034000 032737 000200 002526
2501 034006 001006
2502
2503
2504
2505 034010
      034010 104456
      034012 000002
      034014 007423
      034016 000000
2506 034020 004737 013230
2507
2508
2509
2510 034024
2511 034024
      034024
      034024 104401

;
104:
      MOV      @QDMA,VAL
      BIT      @RDY,VAL
      BNE      204
; READ QDMA REGISTER
; CHECK FOR RDY SET
; BRANCH IF OK

;
; ERROR CONDITION
;
      ERRHRD   2,MSG41
      TRAP     C#ERHRD
      .WORD    2
      .WORD    MSG41
      .WORD    0
      JSR      PC,DUMP
; TELL OPERATOR ERROR

; DO OPTIONAL REGISTER DUMP

;
; END ERROR CONDITION
;
204:
      ENDTST
L10120:
      TRAP     C#ETST

```

TEST33: CHECK QWC OVERFLOW (WRITE)

```

2513 .SBTTL TEST33: CHECK QWC OVERFLOW (WRITE)
2514 034026 STATST <CHECK QWC OVERFLOW (WRITE)>
      034026 040 103 110 T33MSG:: .ASCIZ / CHECK QWC OVERFLOW (WRITE)/
2515
2516 ;*****
2517 ;**
2518 ;TEST DESCRIPTION-
2519 ; CHECKS THAT GO CLEARS RDY BIT
2520 ; CHECKS THAT GO CAUSES DMA REQ
2521 ; CHECKS THAT DMA REQ CAUSES WC INCREMENT
2522 ; CHECKS THAT WC OVERFLOW SETS RDY BIT
2523 ;TEST STEPS
2524 ; LOAD QWC WITH 177776
2525 ; LOAD QBA WITH ANY VALID ADDRESS
2526 ; SET QMT1 FUNC = 5 (SELDMA DAT)
2527 ; SET QDMA GO AND DMAWT BITS
2528 ; CHECK QDMA RDY CLEARED
2529 ; SET QDMA GO AND DMAWT BITS
2530 ; READ QWC, CHECK FOR 0
2531 ; CHECK QDMA RDY SET
2532 ;
2533 ;--
2534 ;*****
2535
2536 034062 BGNTST
      034062 T33::
2537 034062 CKPNT
      034062 TRAP C:RFLA ; PRINT OUT TEST TITLE IF PNT SET
      034064 022700 001000 CMP #PNT,RO ; IS PNT FLAG SET?
      034070 001012 BNE 641 ; 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,RO
      034110 104417 TRAP C:PNTF
      034112 062706 000006 ADD #6,SP
      034116 005037 002532 641: CLR ERROR1 ; CLEAR ERROR COUNTER
2538 034122 INIT
      034122 012777 000100 146166 ;INITIALIZE DEVICE
      034130 013777 002520 146160 MOV #MCMHALT,BQMT1
      034136 005077 146136 MOV INITWORD,BQMT1
      034142 005037 002532 CLR BQINT ;INITIALIZE Q-BRIDGE
      2539 034146 012777 177776 146136 CLR ERROR1 ;SET UP LOOPBACK OPTION
      2540 034154 012777 041660 146126 MOV #177776,BQWC ;CLEAR QINT AFTER LOOPBACK ENABLED
      2541 034162 052777 000024 146126 MOV #PATCH,QBA ;CLEAR ERROR FLAG
      2542 034170 052777 000003 146106 BIS #SELDMA DAT,BQMT1 ;LOAD QWC WITH ALL 1'S
      2543 034176 017737 146102 002526 BIS #GO!DMAWT,BQDMA ;LOAD QBA WITH VALID ADDRESS
      2544 034204 032737 000200 002526 MOV BQDMA,VAL ;SEL QMT1 FUNC = 5
      2545 034212 001406 BIT #RDY,VAL ;SET QDMA GO BIT
      2546 ; ; READ QDMA REGISTER
      2547 ; ; CHECK FOR RDY BIT CLEARED
      2548 ; ; BRANCH IF OK
      2549 034214 104456 ERRHRD 1,MSG37 ; TELL OPERATOR ERROR
      034214 000001 TRAP C:ERRHRD
      034216 007103 .WORD 1
      034220 .WORD MSG37

```


TEST33: CHECK QWC OVERFLOW (WRITE)

```

2550 034222 000000          .WORD 0
034224 004737 013230      JSR    PC,DUMP          ; DO OPTIONAL REGISTER DUMP
2551
2552                      ; END ERROR CONDITION, CHECK FOR QWC INCREMENT
2553
2554 034230
2555 034230 052777 000003 146046      100:  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    200          ; 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
2565                      TRAP    C#ERRHRD
2566                      .WORD 2
2567                      .WORD MSG38
2568                      .WORD ERR4
2569 034276 004737 013230      JSR    PC,DUMP          ; DO OPTIONAL REGISTER DUMP
2570
2571                      ; END ERROR CONDITION, CHECK FOR QDMA RDY SET
2572
2573                      200:
2574 034302                      MOV    @QDMA,VAL          ; READ QDMA REGISTER
2575 034302 017737 145776 002526      BIT    @RDY,VAL          ; CHECK RDY BIT FOR SET
2576 034310 032737 000200 002526      BNE    300          ; BRANCH IF OK
2577 034316 001006
2578
2579                      ; ERROR CONDITION
2580
2581                      300:
2582 034320                      ERRHRD 3,MSG39          ; TELL OPERATOR ERROR
2583 034320 104456                      TRAP    C#ERRHRD
2584 034322 000003                      .WORD 3
2585 034324 007254                      .WORD MSG39
2586 034326 000000                      .WORD 0
2587 034330 004737 013230      JSR    PC,DUMP          ; DO OPTIONAL REGISTER DUMP
2588
2589                      300:
2590 034334                      ENDTST
2591 034334
2592 034334 104401      L10121: TRAP    C#ETST

```

TEST34: DMA 1 WORD (WRITE)

```

2581 .SBTTL TEST34: DMA 1 WORD (WRITE)
2582 034336 STATST <DMA 1 WORD (WRITE)>
      034336 040 104 115 T34MSG:: .ASCIZ / DMA 1 WORD (WRITE)/
2583
2584 ;*****
2585 ;**
2586 ;TEST DESCRIPTION-
2587 ; CHECKS DMA FROM MEMORY FROM Q-BUS TO UNIBUS
2588 ; USES QTSOUT - QTSIN LOOP-BACK
2589 ;TEST STEPS-
2590 ; LOAD DATA WORD IN BUFFER
2591 ; LOAD QMC WITH -1
2592 ; LOAD QBA WITH BUFFER ADDRESS
2593 ; SET QDMA DMAWT AND GO BITS
2594 ; SET QMT1 FUNC = 5 (SELDMA DAT)
2595 ; READ QMT0, CHECK DATA WORD
2596 ; CLEAR QMT1 FUNC = 5 (SELDMA DAT)
2597 ; REPEAT WITH COMPLEMENT DATA PATTERN
2598 ; LOOP 100 TIMES
2599 ;
2600 ;--
2601 ;*****
2602
2603 034362 BGNTST
2604 034362 T34::
      034362 CKPNT
      034364 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
      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
2605 034422 INIT
      034422 ;INITIALIZE DEVICE
      034430 012777 000100 145666 MOV #QHALT,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
2606 034446 012705 000144 MOV #100,R5 ; NUMBER OF LOOPS
2607 034452 012737 125252 002530 MOV #125252,EXP ; STARTING DATA PATTERN TO DMA
2608 034460 BGNSUB
      034460 T34.1:
2609 034462 L34: TRAP C#BSUB
      034462 104404 TRAP C#BSEG
      034462 104404 TRAP C#BSEG
2610 034464 012777 177777 145620 MOV #-1,QMC ; -1 TO QMC
2611 034472 012777 002530 145610 MOV #EXP,QBA ; RECEIVING BUFFER TO QBA
2612 034500 052777 000003 145576 BIS #DMAWT:GO,QDMA ; INITIATE DMA
2613 034506 052777 000024 145602 BIS #SELDMA DAT,QMT1 ; SET QMT1 FUNC = 5
2614 034514 017737 145574 002526 MOV QMT0,VAL ; READ QMT0 (RECEIVED DMA DATA)
2615 034522 042777 000024 145566 BIC #SELDMA DAT,QMT1 ; CLEAR QMT1 FUNC = 5
2616 034530 023737 002530 002526 CMP EXP,VAL ; COMPARE EXPECTED WITH VALUE
2617 034536 001430 BEQ 20# ; BRANCH IF OK

```

TEST34: DMA 1 WORD (WRITE)

```

2618
2619
2620
2621 034540
      034540 104456
      034542 000001
      034544 007520
      034546 000000
2622 034550 012704 000145
2623 034554 160504
2624 034556
      034556 010446
      034560 013746 002526
      034564 013746 002530
      034570 012746 004261
      034574 012746 000004
      034600 010600
      034602 104414
      034604 062706 000012
2625 034610 004737 013230
2626 034614 005237 002532
2627 034620
2628 034620
      034620 104405
2629 034622 005137 002530
2630 034626 005305
2631 034630 001405
2632 034632 005737 002532
2633 034636 001002
2634 034640 000137 034462
2635 034644
2636 034644
      034644 104403
2637 034646
      034646 104401
      034646 104401

; ERROR CONDITION
;
ERRHRD 1,MSG42
TRAP C#ERRHRD
; TELL OPERATOR ERROR
;
; WORD 1
; WORD MSG42
; WORD 0
; CALCULATE LOOP NUMBER
MOV #101.,R4
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
; DO OPTIONAL REGISTER DUMP
JSR PC,DUMP
; SET ERROR FLAG
INC ERROR1

200:
ENDSEG
100000:
TRAP C#ESEG
COM EXP
; COMPLEMENT DATA PATTERN
DEC R5
; DECREMENT LOOP COUNTER
BEQ 300
; BRANCH IF DONE WITH LOOPS
TST ERROR1
; CHECK FOR ERRORS
BNE 300
; IF ANY ERRORS, END TEST
JMP L34
300:
ENDSUB
L10123:
TRAP C#ESUB
ENDTST
L10122:
TRAP C#ETST

```


TEST35: DMA 1 WORD (READ)

```

2639 .SBTTL TEST35: DMA 1 WORD (READ)
2640 034650 STATST <DMA 1 WORD (READ)>
034650 040 104 115 T35MSG:: .ASCIZ / DMA 1 WORD (READ)/

2641 ;*****
2642 ;**
2643 ;TEST DESCRIPTION-
2644 ; CHECKS DMA TO MEMORY FROM UNIBUS TO Q-BUS
2645 ; USES UDMAENB - QDMAENB LOOP-BACK
2646 ; USES QTSOUT - QTSIN LOOP-BACK
2647 ;TEST STEPS-
2648 ; SET QMT1 FUNC = 7 (SEL DMA READ)
2649 ; CLEAR BUFFER
2650 ; LOAD QMT0 WITH DATA PATTERN
2651 ; LOAD QMC WITH -1
2652 ; LOAD QBA WITH BUFFER ADDRESS
2653 ; SET QDMA GO BITS
2654 ; READ DATA IN BUFFER, CHECK
2655 ; REPEAT WITH COMPLEMENT DATA PATTERN
2656 ; LOOP 100 TIMES
2657 ;
2658 ;
2659 ;--
2660 ;*****
2661
2662 034674 BGNTST
034674
2663 034674 T35:: 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
2664 034734 INIT
;INITIALIZE DEVICE
034734 012777 000100 145354 MOV #QHALT,QMT1 ;INITIALIZE Q-BRIDGE
034742 013777 002520 145346 MOV INITWORD,QMT1 ;SET UP LOOPBACK OPTION
034750 005077 145324 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
034754 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
2665 034760 012705 000144 MOV #100.,R5 ; NUMBER OF LOOPS
2666 034764 012737 125252 002530 MOV #125252,EXP ; STARTING DATA PATTERN TO DMA
2667 034772 BGNSUB
034772
T35.1: TRAP C#BSUB
2668 034774 L35: BGNSEG
034774 104404 TRAP C#BSEG
034776 052777 000034 145312 BIS #SELRDDMA,QMT1 ; SET QMT1 FUNC = 7
2670 035004 012737 000000 041660 MOV #0,#PATCH ; CLEAR BUFFER
2671 035012 013777 002530 145274 MOV EXP,QMT0 ; LOAD DATA PATTERN INTO QMT0
2672 035020 012777 177777 145264 MOV #-1,QMC ; -1 TO QMC
2673 035026 012777 041660 145254 MOV #PATCH,QBA ; RECEIVING BUFFER TO QBA
2674 035034 052777 000001 145242 BIS #GO,QDMA ; INITIATE DMA
2675 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							
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						
2688	035140						
	035140				ENDSEG		
	035140	104405					
2689	035142	005137	002530		TRAP	C#ESEG	
2690	035146	005305			COM	EXP	: COMPLEMENT DATA PATTERN
2691	035150	001405			DEC	R5	: DECREMENT LOOP COUNTER
2692	035152	005737	002532		BEQ	30:	: BRANCH IF DONE WITH LOOPS
2693	035156	001002			TST	ERROR1	: CHECK FOR ERRORS
2694	035160	000137	034774		BNE	30:	: IF ANY ERRORS, END TEST
2695	035164				JMP	L35	: PREFORM TEST AGAIN
2696	035164						
	035164				ENDSUB		
	035164	104403					
2697	035166				TRAP	C#ESUB	
	035166				ENDTST		
	035166	104401					
					L10124:	TRAP	C#ETST

TEST36: DMA 1 WORD HI MEMORY (WRITE)

```

2699 .SBTTL TEST36: DMA 1 WORD HI MEMORY (WRITE)
2700 035170 STATST <DMA 1 WORD HI MEMORY (WRITE)>
      035170 040 104 115 T36MSG:: .ASCIZ / DMA 1 WORD HI MEMORY (WRITE)/
2701
2702 ;*****
2703 ;**
2704 ;TEST DESCRIPTION-
2705 ; CHECKS DMA FROM MEMORY FROM Q-BUS HIGHEST AVAILABLE
2706 ; MEMORY TO UNIBUS
2707 ; USES QTSOUT - QTSIN LOOP-BACK
2708 ;TEST STEPS-
2709 ; CHECK IF > 32K WORD MEMORY
2710 ; IF YES-
2711 ;     SET UP MEMORY MANAGEMENT REGISTERS
2712 ;     GET TOP OF PROGRAM ADDRESS, CALCULATE NEXT 4KW BLOCK
2713 ;     GET TOP THREE BITS OF BLOCK, USE THAT FOR INDEX TO MM REGISTER
2714 ;     THIS ADDRESS STORED (TRANSFER ADDRESS)
2715 ;     GET HIGHEST AVAILABLE ADDRESS
2716 ;     SET EXTENDED ADDRESS BITS 16 AND 17 FROM HIGHEST ADDRESS
2717 ;     CONVERT HIGHEST AVAILABLE ADDRESS TO 16 BIT ADDRESS
2718 ;     ENABLE MAPPING
2719 ;     WRITE EXPECTED DATA PATTERN TO MEMORY AT TRANSFER ADDRESS
2720 ;
2721 ; ELSE TRANSFER AT TOP OF BUFFER
2722 ;
2723 ; LOAD QWC WITH -1
2724 ; LOAD QBA WITH BUFFER ADDRESS
2725 ; SET QDMA DMAWT AND GO BITS
2726 ; SET QMT1 FUNC = 5 (SELDMA DAT)
2727 ; READ QMT0, CHECK DATA WORD
2728 ; CLEAR QMT1 FUNC = 5 (SELDMA DAT)
2729 ; REPEAT WITH COMPLEMENT DATA PATTERN
2730 ; LOOP 100 TIMES
2731 ;--
2732 ;*****
2733
2734 035226 BGNTST
2735 035226 T36::
      035226 104421 CKPNT
      035230 022700 001000 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
      035234 001012 CMP #PNT,RO ; IS PNT FLAG SET?
      035236 012746 035170 BNE 644 ; BRANCH IF NOT SET
      035242 012746 002270 MOV #T36MSG, -(SP)
      035246 012746 000002 MOV #PNTMSG, -(SP)
      035252 010600 MOV #2, -(SP)
      035254 104417 MOV SP,RO
      035256 062706 000006 TRAP C#PNTF
      035262 005037 002532 ADD #6,SP
      644: 035266 005037 002532 CLR ERROR1 ; CLEAR ERROR COUNTER
      035266 INIT
      035266 012777 000100 145022 ;INITIALIZE DEVICE
      035274 013777 002520 145014 MOV #MQHALT, @QMT1
      035302 005077 144772 MOV INITWORD, @QMT1
      035306 005037 002532 CLR @QINT ;INITIALIZE Q-BRIDGE
      2737 035312 012705 000144 CLR ERROR1 ;SET UP LOOPBACK OPTION
      2738 035316 012737 125252 002530 MOV #100, R5 ;CLEAR QINT AFTER LOOPBACK ENABLED
      ; NUMBER OF LOOPS
      ; STARTING DATA PATTERN TO DMA

```


TEST36: DMA 1 WORD HI MEMORY (WRITE)

2739	035324	005002			CLR	R2		
2740	035326				BGNSUB			
	035326							
	035326	104402			T36.1:	TRAP	C#BSUB	
2741	035330	022737	002000	002120		CMP	#2000,L#HIME	: > 32K WORD MEMORY?
2742	035336	002404				BLT	2#	: IF YES BRANCH
2743								
2744								
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.O
2749								
2750								
2751								
2752	035350							
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	
2764	035406	013760	002120	172340		MOV	L#HIME,KISARO(R0)	: TIMES 2 FOR INDEX USE
2765	035414	005002				CLR	R2	: INDEX TO MM REGISTER TO USE
2766	035416	032737	004000	002120		BIT	#BIT11,L#HIME	: 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#HIME	: BIT 10 OF HIGHEST AVAILABLE MEMORY
2770	035440	001402				BEQ	4#	: IF NOT ON
2771	035442	052702	000400			BIS	#QXAD16,R2	: SET BIT 16
2772	035446	012700	000006		4#:	MOV	#6,R0	
2773	035452	013701	002120			MOV	L#HIME,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.O
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	#SELDMA DAT,BQMT1	: SET QMT1 FUNC = 5
2784	035526	017737	144562	002526		MOV	BQMT0,VAL	: READ QMT0 (RECEIVED DMA DATA)
2785	035534	042777	000024	144554		BIC	#SELDMA DAT,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								
2790								
2791	035552							
	035552	104456				ERRHRD	1,MSG42.5	: TELL OPERATOR ERROR
	035554	000001				TRAP	C#ERRHRD	
						.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						
2798	035632			20:	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	:	PERFORM TEST AGAIN : V C.0
2803	035650	012737	000000 177572	MOV	#0,MHRO	:	DISABLE MAPPING
2804							
2805	035656			30:			
2806	035656			L10126:	ENDTST		
	035656						
	035656	104401		TRAP	C#ETST		

TEST37: CHECK WDTO TIME

```

2808 .SBTTL TEST37: CHECK WDTO TIME
2809 035660 STATST <CHECK WDTO TIME>
      035660 040 103 110 T37MSG:: .ASCIZ / CHECK WDTO TIME/

2810 ;*****
2811 ;**
2812 ;TEST DESCRIPTION-
2813 ; CHECKS QCTL BITS
2814 ; CHECKS THAT TIMER IS OPERATING
2815 ; ALLOWS OPERATOR TO CHECK TIMES
2816 ; USES UPWRFail - QMOTO LOOP-BACK
2817 ;TEST STEPS-
2818 ; SET TIME CODE IN QCTL
2819 ; READ QCTL TIME CODE BITS, CHECK
2820 ; SET QCTL THRENB
2821 ; READ QCTL THRENB, CHECK
2822 ; WAIT FOR QINT WPMRFL BIT SET ON TIME-OUT
2823 ; RING BELL
2824 ; REPEAT 5 TIMES
2825 ; REPEAT WITH ALL TIMES CODES
2826 ;
2827 ;
2828 ;--
2829 ;*****
2830
2831 035702 BGNTST
      035702
2832 035702 T37:: CKPNT
      035702 104421 TRAP C:RFLA ; PRINT OUT TEST TITLE IF PNT SET
      035704 022700 001000 CMP #PNT,RO ; IS PNT FLAG SET?
      035710 001012 BNE 644 ; 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,RO
      035730 104417 TRAP C:PNTF
      035732 062706 000006 ADD #6,SP
      035736 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
2833 035742 INIT
      035742 012777 000100 144346 ; INITIALIZE DEVICE
      035750 013777 002520 144340 MOV #QHALT,BQMT1
      035756 005077 144316 MOV INITWORD,BQMT1
      035762 005037 002532 CLR BQINT
      035766 012701 000004 CLR ERROR1
2834 035766 012701 000004 MOV #4,R1
2835 035772 012702 000005 MOV #5,R2
2836 035776 012703 036362 MOV #TABLE36,R3
2837 036002 005737 036372 TST P37
2838 036006 001012 BNE L37
2839 036010 PRINTB #FOR11
      036010 012746 004353 MOV #FOR11,-(SP)
      036014 012746 000001 MOV #1,-(SP)
      036020 010600 MOV SP,RO
      036022 104414 TRAP C:PNTB
      036024 062706 000004 ADD #4,SP
2840 036030 005237 036372 INC P37 ; INCREMENT PRINTED ABOVE MESSAGE FLAG
2841 036034
2842 036034 012777 000000 144236 L37: MOV #0,BQINT ; MAKE SURE UPWRFail IS CLEAR

```


TEST37: CHECK WDTO TIME

```

2843 036042 012777 000000 144232      MOV      #0,BQCTL      ; MAKE SURE TIMER ENABLE IS CLEAR
2844 036050 051377 144226      BIS      (R3),BQCTL    ; SET TIME CODE BITS IN QCTL
2845 036054 017737 144222 002526      MOV      BQCTL,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      104          ; 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$ERRHRD
      036122 000001      .WORD      1
      036124 007663      .WORD      MSG44
      036126 011776      .WORD      ERR4
2857 036130      EXIT      TST          ; ABORT TEST
      036130 104432      TRAP      C$EXIT
      036132 000242      .WORD      L10130-.
2858
2859      ; END ERROR CONDITION
2860
2861 036134      104:
2862 036134 052777 000002 144140      BIS      #THRENB,BQCTL    ; ENABLE TIMER
2863 036142 017737 144134 002526      MOV      BQCTL,VAL    ; READ QCTL
2864 036150 032737 000002 002526      BIT      #THRENB,VAL  ; TEST FOR THRENB BIT SET
2865 036156 001016      BNE      204          ; 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      #THRENB,EXP    ; EXPECTED VALUE TO EXP
2871 036172 042737 177775 002526      BIC      #+CTHRENB,VAL  ; CLEAR UNTESTED BIT FOR MESSAGE
2872 036200      ERRHRD 2,MSG45,ERR4      ; TELL OPERATOR ERROR AND ABORT
      036200 104456      TRAP      C$ERRHRD
      036202 000002      .WORD      2
      036204 007775      .WORD      MSG45
      036206 011776      .WORD      ERR4
2873 036210      EXIT      TST          ; ABORT TEST
      036210 104432      TRAP      C$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      204:
2879 036214 012704 000360      MOV      #240.,R4      ; NUMBER OF .025 SEC TO LOOP
2880 036220      304:
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
036250 104422      TRAP      C4BRK      ; CHECK FOR OPERATOR INPUT
2883 036252 032777 010000 144020  BIT      @UPWRFL,BQINT      ; TEST FOR WDTO COMPLETE (BIT SET)
2884 036260 001012      BNE      404      ; IF TIMER COMPLETE, BRANCH
2885 036262 005304      DEC      R4      ; DECREMENT R4 COUNTER
2886 036264 001355      BNE      304      ; 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      C4ERHRD
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      C4EXIT
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
2898 036306      404:      PRINTB      @FOR12      ; RING BELL
036306 012746 004462      MOV      @FOR12,-(SP)
036312 012746 000001      MOV      @1,-(SP)
036316 010600      MOV      SP,R0
036320 104414      TRAP      C4PNTB
036322 062706 000004      ADD      @4,SP
2899 036326 005302      DEC      R2      ; DECREMENT LOOP COUNTER
2900 036330 001402      BEQ      504      ; 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      504:
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      604      ; BRANCH IF DONE
2910 036352 000137 036034      JMP      L37      ; GO TEST NEXT TIME CODE
2911 036356
2912 036356      604:      EXIT      TST
036356 104432      TRAP      C4EXIT
036360 000014      .WORD      L10130-.
2913
2914      ; TABLE OF 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

```

G12

CZFPA-AO CTRL BRIDGE INTFC DIAG MACRO M1200 09-JAN-84 12:33 PAGE 77-3

SEQ 0149

TEST37: CHECK WDTO TIME

036374
036374 104401

L10130: TRAP C#ETST

TEST38: CHECK WDTO RESET

```

2924 .SBTTL TEST38: CHECK WDTO RESET
2925 036376 STATST <CHECK WDTO RESET>
2926 036376 040 103 110 T38MSG:: .ASCIZ / CHECK WDTO RESET/
2927 ;*****
2928 ;**
2929 ;TEST DESCRIPTION-
2930 ; CHECKS THAT TIMER CAN BE RESET
2931 ; USES UPWRFAIL - QWDTO LOOP-BACK
2932 ;TEST STEPS-
2933 ; SET QCTL THRENB
2934 ; WAIT < 10 MILLISECONDS
2935 ; CHECK QINT UPWRFL FOR 0
2936 ; SET QCTL WDTRST BIT
2937 ; REPEAT 1000 TIMES
2938 ;
2939 ;--
2940 ;*****
2941
2942 036420 BGNTST
2943 036420 T38:: CKPNT
2944 036420 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
2945 036420 104421 001000 143630 143622 MOV #PNT,R0 ; IS PNT FLAG SET?
2946 036422 022700 001000 143622 CMP #64, ; BRANCH IF NOT SET
2947 036426 001012 001000 143622 BNE 64$
2948 036430 012746 036376 143622 MOV #T38MSG,-(SP)
2949 036434 012746 002270 143622 MOV #PNTMSG,-(SP)
2950 036440 012746 000002 143622 MOV #2,-(SP)
2951 036444 010600 000002 143622 MOV SP,R0
2952 036446 104417 000006 143622 TRAP C#PNTF
2953 036450 062706 000006 143622 ADD #6,SP
2954 036454 005037 002532 143622 CLR ERROR1 ; CLEAR ERROR COUNTER
2955 036460 005037 002532 143622 INIT
2956 036460 012777 000100 143630 ; INITIALIZE DEVICE
2957 036466 013777 002520 143622 MOV #MHALT,BQMT1
2958 036474 005077 143600 143622 MOV INITWORD,BQMT1
2959 036500 005037 002532 143622 CLR BQINT
2960 036504 012701 001750 143622 CLR ERROR1
2961 036510 012701 001750 143622 MOV #1000,,R1
2962 036510 012701 001750 143622 BGNSUB
2963 036510 012701 001750 143622 T38.1:
2964 036512 052777 000002 143562 L38: TRAP C#BSUB
2965 036520 012727 000003 143562 BIS #THRENB,BQCTL ; SET THRENB BIT IN QCTL
2966 036524 000000 000003 143562 DELAY 3 ; WAIT ABOUT 7.5 MS
2967 036526 013727 002116 143562 MOV #3,(PC)+
2968 036532 000000 000003 143562 .WORD 0
2969 036534 005367 177772 143562 MOV L#DLY,(PC)+
2970 036540 001375 177756 143562 .WORD 0
2971 036542 005367 177756 143562 DEC -6(PC)
2972 036546 001367 177756 143562 BNE -.4
2973 036550 032777 010000 143522 DEC -22(PC)
2974 036556 001410 001410 143522 BNE -.20
2975 036556 001410 001410 143522 BIT #UPWRFL,BQINT ; CHECK UPWRFL FOR CLEAR
2976 036556 001410 001410 143522 BEQ 10$ ; BRANCH IF OK
2977 036556 001410 001410 143522 ;
2978 036556 001410 001410 143522 ; ERROR CONDITION
2979 036556 001410 001410 143522 ;

```

SEQ 0151

2954	036560				ERRHRD	1,MSG47		; TELL OPERATOR ERROR
	036560	104456			TRAP	C#ERRRD		
	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								
2959								
2960	036600							
2961	036600	052777	000001	143474	BIS	#WDTRST,BQCTL		; 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		

TEST39: CHECK VNTACK INTERRUPT

```

2967 .SBTTL TEST39: CHECK VNTACK INTERRUPT
2968 036616 STATST <CHECK VNTACK INTERRUPT>
      036616 040 103 110 T39MSG: .ASCIZ / CHECK VNTACK INTERRUPT/

2969 ;*****
2970 ;**
2971 ;TEST DESCRIPTION-
2972 ; CHECKS THAT VNTACK SETTING CAN CAUSE INTERRUPT
2973 ; USES QCMDACK - UEVENTACK LOOP-BACK
2974 ;TEST STEPS-
2975 ; SET INTERRUPT VECTOR
2976 ; SET QVNT ACKIE BIT
2977 ; CHECK FOR BIT SET
2978 ; SET QCMD CMDACK BIT
2979 ; CHECK FOR IMMEDIATE INTERRUPT
2980 ; RESTORE INTERRUPT VECTOR
2981 ;
2982 ;--
2983 ;*****
2984
2985
2986 036646 BGNTST
      036646
2987 036646 T39:
      036646 CKPNT
      036646 TRAP C:RFLA ; PRINT OUT TEST TITLE IF PNT SET
      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
2988 036706 INIT
      036706 012777 000100 143402 ;INITIALIZE DEVICE
      036714 013777 002520 143374 MOV #MHALT,BQMT1
      036722 005077 143352 MOV INITWORD,BQMT1
      036726 005037 002532 CLR BQINT
      036732 005037 002532 CLR ERROR1
2989 036732 SETVEC VECTOR,#L39,#PRI07 ; INITIALIZE Q-BRIDGE
      036732 012746 000340 MOV #PRI07,-(SP) ; SET UP LOOPBACK OPTION
      036736 012746 037060 MOV #L39,-(SP) ; CLEAR QINT AFTER LOOPBACK ENABLED
      036742 013746 002512 MOV VECTOR,-(SP) ; CLEAR ERROR FLAG
      036746 012746 000003 MOV #3,-(SP) ; SET INTERRUPT VECTOR
      036752 104437 TRAP C:SVEC
      036754 062706 000010 ADD #10,SP
2990 036760 052777 000100 143310 BIS #ACKIE,BQVNT ; SET ACKIE BIT
2991 036766 017737 143304 002526 MOV BQVNT,VAL ; READ QVNT
2992 036774 032737 000100 002526 BIT #ACKIE,VAL ; IS ACKIE BIT REALLY SET
2993 037002 001012 BNE 10: ; BRANCH IF OK
2994
2995 ; ERROR CONDITION
2996
2997 037004 012701 000001 MOV #1,R1 ; REGISTER NUMBER TO R1
2998 037010 012737 000100 002530 MOV #ACKIE,EXP ; EXPECTED PATTERN TO EXP
2999 037016 037016 104456 ERRHRD 1,MSG48,ERR4 ; TELL OPERATOR ERROR
      037016 104456 TRAP C:ERRRD

```


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
3005 037030 052777 000001 143236 104:  BIS #CMDACK,BQCMD      ; 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
037042 104456          ERRHRD 2,MSG49      ; TELL OPERATOR ERROR
037044 000002          TRAP C#ERHRD
037046 010247          .WORD 2
037050 000000          .WORD MSG49
037052 004737 013230    .WORD 0
3013 037056 000403      JSR PC,DUMP      ; DO OPTIONAL REGISTER DUMP
037056 000403      BR E39          ; SKIP INTERRUPT SERVICE ROUTINE
3014
3015          ; END ERROR CONDITION
3016          ; INTERRUPT SERVICE ROUTINE
3017
3018 037060
037060          BGNSRV L39
3019 037060 012716 037066 L39::      MOV #E39,(SP)
3020 037064          ENDSRV
037064          L10134:
037064 000002          RTI
3021
3022          ; END INTERRUPT SERVICE ROUTINE
3023
3024 037066
3025 037066          E39:
037066 013700 002512    CLRVEC VECTOR      ; CLEAR VECTOR
037072 104436          MOV VECTOR,R0
037074          TRAP C#CVEC
3026 037074          ENDTST
037074 104401          L10133:
037074 104401          TRAP C#ETST

```

TEST40: CHECK CMDPRS INTERRUPT

```

3028 .SBTTL TEST40: CHECK CMDPRS INTERRUPT
3029 037076 STATST <CHECK CMDPRS INTERRUPT>
      037076 040 103 110 T40MSG:: .ASCIZ / CHECK CMDPRS INTERRUPT/
3030
3031 ;*****
3032 ;**
3033 ;TEST DESCRIPTION-
3034 ; CHECKS THAT CMDPRS SETTING CAN CAUSE INTERRUPT
3035 ; USES QEVENTPRS - UCMDPRS LOOP-BACK
3036 ;TEST STEPS-
3037 ; SET INTERRUPT VECTOR
3038 ; SET QCMD PRSIE BIT
3039 ; CHECK FOR BIT SET
3040 ; SET QVNT VNTPRS BIT
3041 ; CHECK FOR IMMEDIATE INTERRUPT
3042 ; RESTORE INTERRUPT VECTOR
3043 ;
3044 ;--
3045 ;*****
3046
3047 037126 BGNTST
      037126
3048 037126 T40:: CKPNT
      037126 104421 TRAP C#RFLA ; PRINT OUT TEST TITLE IF PNT SET
      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
3049 037166 INIT
      037166 012777 000100 143122 ;INITIALIZE DEVICE
      037174 013777 002520 143114 MOV #MGMHALT,BQMT1
      037202 005077 143072 MOV INITWORD,BQMT1
      037206 005037 002532 CLR BQINT ;INITIALIZE Q-BRIDGE
      037212 012746 000340 CLR ERROR1 ;SET UP LOOPBACK OPTION
      037216 012746 037340 SETVEC VECTOR,#L40,#PRI07 ;CLEAR QINT AFTER LOOPBACK ENABLED
      037222 013746 002512 MOV #PRI07,-(SP) ;CLEAR ERROR FLAG
      037226 012746 000003 MOV #L40,-(SP) ; SET INTERRUPT VECTOR
      037232 104437 TRAP C#SVEC
      037234 062706 000010 ADD #10,SP
      037240 012777 000100 143026 MOV #PRSIE,BQCMD ; SET QCMD PRSIE BIT
      037246 017737 143022 002526 MOV BQCMD,VAL ; READ QCMD
      037254 032737 000100 002526 BIT #PRSIE,VAL ; IS PRSIE BIT SET
      037262 001012 BNE 10# ; BRANCH IF OK
3055 ;
3056 ; ERROR CONDITION
3057 ;
3058 037264 012701 000000 MOV #0,R1 ; REGISTER NUMBER TO R1
3059 037270 012737 000100 002530 MOV #PRSIE,EXP ; EXPECTED PATTERN TO EXP
3060 037276 104456 ERHRD 1,MSG50,ERR4 ; TELL OPERATOR ERROR
      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          ;
3064          ; END ERROR CONDITION
3065 037310
3066 037310 052777 000001 142760      106:  BIS #VNTPRS,@QVNT      ; SET VNTPRS BIT, SHOULD INTERRUPT NOW
3067 037316 000240          NOP          ; ALLOW SMALL AMOUNT OF TIME
3068 037320 000240          NOP
3069
3070          ;
3071          ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
3072          ;
3072 037322          ERRHRD 2,MSG51      ; TELL OPERATOR ERROR
3073 037322 104456      TRAP C#ERHRD
3074 037324 000002          .WORD 2
3075 037326 010407          .WORD MSG51
3076 037330 000000          .WORD 0
3077 037332 004737 013230      JSR PC,DUMP      ; DO OPTIONAL REGISTER DUMP
3078 037336 000403          BR E40          ; SKIP INTERRUPT SERVICE ROUTINE
3079
3080          ;
3081          ; END ERROR CONDITION
3082          ; INTERRUPT SERVICE ROUTINE
3083          ;
3084          ;
3085 037340          BGNSRV L40
3086 037340          L40::
3087 037340 012716 037346      MOV #E40,(SP)
3088 037344          ENDSRV
3089 037344          L10136:
3090 037344 000002          RTI
3091
3092          ;
3093          ; END INTERRUPT SERVICE ROUTINE
3094          ;
3095 037346          E40:
3096 037346          CLRVEC VECTOR      ; CLEAR VECTOR
3097 037346 013700 002512      MOV VECTOR,R0
3098 037352 104436          TRAP C#CVEC
3099 037354          ENDTST
3100 037354          L10135:
3101 037354 104401          TRAP C#ETST

```


TEST41: CHECK RDY INTERRUPT

```

3089 .SBTTL TEST41: CHECK RDY INTERRUPT
3090 037356 STATST <CHECK RDY INTERRUPT>
      037356 040 103 110 T41MSG:: .ASCIZ / CHECK RDY INTERRUPT/
3091
3092 ;*****
3093 ;**
3094 ;TEST DESCRIPTION-
3095 ; CHECKS THAT RDY BIT SET CAUSES INTERRUPT
3096 ;TEST STEPS-
3097 ; SET INTERRUPT VECTOR
3098 ; SET QDMA RDYIE BIT
3099 ; CHECK FOR BIT SET
3100 ; SET QWC = -1
3101 ; SET QBA = (ANY LEGAL ADDRESS)
3102 ; SET QDMA DMAWT, GO BITS
3103 ; CHECK FOR IMMEDIATE INTERRUPT
3104 ; RESTORE INTERRUPT VECTOR
3105 ;
3106 ;--
3107 ;*****
3108
3109 037404 BGNTST
      037404 T41::
3110 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
3111 037444 INIT
      037444 012777 000100 142644 ;INITIALIZE DEVICE
      037452 013777 002520 142636 MOV #QHALT,QGMT1
      037460 005077 142614 MOV INITWORD,QGMT1
      037464 005037 002532 CLR QINT ;INITIALIZE Q-BRIDGE
      037470 012746 000340 142560 CLR ERROR1 ;SET UP LOOPBACK OPTION
      037474 012746 037632 002526 CLR VECTOR, #L41, #PRI07 ;CLEAR QINT AFTER LOOPBACK ENABLED
      037500 013746 002512 000003 MOV #PRI07,-(SP) ;CLEAR ERROR FLAG
      037504 012746 000003 MOV #L41,-(SP) ; SET INTERRUPT VECTOR
      037510 104437 TRAP C#SVEC
      037512 062706 000010 ADD #10,SP
3113 037516 052777 000100 142560 BIS #RDYIE,QDMA ; SET QDMA RDYIE BIT
3114 037524 017737 142554 002526 MOV QDMA,VAL ; READ QDMA INTO VAL
3115 037532 032737 000100 002526 BIT #RDYIE,VAL ; IS RDYIE BIT REALLY SET
3116 037540 001012 BNE 10# ; BRANCH IF OK
3117
3118 ; ERROR CONDITION
3119
3120 037542 012701 000004 MOV #4,R1 ; REGISTER NUMBER TO R1
3121 037546 012737 000100 002530 MOV #RDYIE,EXP ; EXPECTED PATTERN TO EXP
3122 037554 ERRHRD 1,MSG52,ERR4 ; TELL OPERATOR ERROR

```

TEST41: CHECK RDY INTERRUPT

037554	104456			TRAP	C0ERHRD	
037556	000001			.WORD	1	
037560	010477			.WORD	MSG52	
037562	011776			.WORD	ERR4	
3123	037564	000425		BR	E41	: BRANCH TO END
3124						
3125						
3126						
3127	037566					
3128	037566	012777	177777	101:	MOV	#-1,BQWC
3129	037574	012777	041660		MOV	#PATCH,BQBA
3130	037602	052777	000003		BIS	#GO!DMAWT,BQDMA
3131	037610	000240	142516		NOP	
3132	037612	000240	142506		NOP	
3133						
3134						
3135						
3136	037614					
	037614	104456		ERRHRD	2,MSG53	: TELL OPERATOR ERROR
	037616	000002		TRAP	C0ERHRD	
	037620	010547		.WORD	2	
	037622	000000		.WORD	MSG53	
3137	037624	004737	013230	.WORD	0	
3138	037630	000403		JSR	PC,DUMP	: DO OPTIONAL REGISTER DUMP
3139				BR	E41	: BRANCH OVER INTERRUPT AREA
3140						
3141						
3142						
3143	037632					
	037632					
3144	037632	012716	037640	L41::	MOV	#E41,(SP)
3145	037636				ENDSRV	: ADJUST STACK
	037636					
	037636	000002		L10140:	RTI	
3146	037640			E41:		
3147	037640				CLRVEC	VECTOR
	037640	013700	002512		MOV	VECTOR,R0
	037644	104436			TRAP	C0CVEC
3148	037646				ENDTST	: CLEAR VECTOR
	037646					
	037646	104401		L10137:	TRAP	C0ETST

TEST42: CHECK ERR INTERRUPT (UNEX)

```

3150 .SBTTL TEST42: CHECK ERR INTERRUPT (UNEX)
3151 037650 STATST <CHECK ERR INTERRUPT (UNEX)>
3152 037650 040 103 110 T42MSG:: .ASCIZ / CHECK ERR INTERRUPT (UNEX)/
3153 ;*****
3154 ;**
3155 ;TEST DESCRIPTION-
3156 ; CHECKS THAT UNEX CAN CAUSE INTERRUPT
3157 ; USES QPWRFL - UNEX LOOP-BACK
3158 ;TEST STEPS-
3159 ; SET INTERRUPT VECTOR
3160 ; SET QINT ERRIE BIT, CHECK
3161 ; SET QMT1 MOPFL BIT
3162 ; CHECK FOR IMMEDIATE INTERRUPT
3163 ; CHECK QINT ERR, UNEX BITS
3164 ; CLEAR QMT1 MOPFL BIT
3165 ; RESTORE INTERRUPT VECTOR
3166 ;
3167 ;--
3168 ;*****
3169
3170 037704 BGNTST
3171 037704 T42::
3172 037704 CKPNT
3173 037704 TRAP C1RFLA ; PRINT OUT TEST TITLE IF PNT SET
3174 037706 022700 001000 CMP #PNT,RO ; IS PNT FLAG SET?
3175 037712 001012 001000 BNE 641 ; BRANCH IF NOT SET
3176 037714 012746 037650 MOV #T42MSG,-(SP)
3177 037720 012746 002270 MOV #PNTMSG,-(SP)
3178 037724 012746 000002 MOV #2,-(SP)
3179 037730 010600 MOV SP,RO
3180 037732 104417 TRAP C1PNTF
3181 037734 062706 000006 ADD #6,SP
3182 037740 005037 002532 641: CLR ERROR1 ; CLEAR ERROR COUNTER
3183 037744 INIT
3184 ;INITIALIZE DEVICE
3185 037744 012777 000100 142344 MOV #QHALT,QMT1 ; INITIALIZE Q-BRIDGE
3186 037752 013777 002520 142336 MOV INITWORD,QMT1 ; SET UP LOOPBACK OPTION
3187 037760 005077 142314 CLR QINT ; CLEAR QINT AFTER LOOPBACK ENABLED
3188 037764 005037 002532 CLR ERROR1 ; CLEAR ERROR FLAG
3189 3173 037770 SETVEC VECTOR,#L42,#PRI07 ; SET INTERRUPT VECTOR
3190 037770 012746 000340 MOV #PRI07,-(SP)
3191 037774 012746 040116 MOV #L42,-(SP)
3192 040000 013746 002512 MOV VECTOR,-(SP)
3193 040004 012746 000003 MOV #3,-(SP)
3194 040010 104437 TRAP C1SVEC
3195 040012 062706 000010 ADD #10,SP
3196 3174 040016 052777 000100 142254 BIS #ERRIE,QINT ; SET QINT ERRIE BIT
3197 3175 040024 017737 142250 002526 MOV QINT,VAL ; READ QINT INTO VAL
3198 3176 040032 032737 000100 002526 BIT #ERRIE,VAL ; IS ERRIE BIT REALLY SET
3199 3177 040040 001012 BNE 106 ; BRANCH IF OK
3200 3178
3201 3179
3202 3180 ; ERROR CONDITION
3203 3181 040042 012701 000002 MOV #2,R1 ; REGISTER NUMBER TO R1
3204 3182 040046 012737 000100 002530 MOV #ERRIE,EXP ; EXPECTED PATTERN TO EXP
3205 3183 040054 ERRHRD 1,MSG54,ERR4 ; TELL OPERATOR ERROR

```


TEST42: CHECK ERR INTERRUPT (UNEX)

```

040054 104456      TRAP      C1ERHRD
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
3189 040066 052777 000040 142222      101:      BIS      #MQPFL,BQMT1      ; 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      C1ERHRD
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
3203 040116 017737 142156 002526      L42::      MOV      BQINT,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      201                ; 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      C1ERHRD
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      201:      MOV      #E42,(SP)      ; ADJUST STACK
3220 040204      ENDSRV
040204
040204 000002      L10142:      RTI
3221 040206
3222 040206 042777 000040 142102      E42:      BIC      #MQPFL,BQMT1      ; CLEAR MQPFL BIT IN QMT1
3223 040214      CLRVEC     VECTOR      ; CLEAR VECTOR
040214 013700 002512      MOV      VECTOR,R0
040220 104436      TRAP      C1CVEC

```

E13

CZFPA-A0 CTRL BRIDGE INTFC DIAG MACRO M1200 09-JAN-84 12:33 PAGE 82-2

SEQ 0160

TEST42: CHECK ERR INTERRUPT (UNIX)

3224 040222
040222
040222 104401

ENDTST
L10141: TRAP C#ETST

TEST43: CHECK ERR INTERRUPT (QNEK)

```

3226 .SBTTL TEST43: CHECK ERR INTERRUPT (QNEK)
3227 040224 STATST <CHECK ERR INTERRUPT (QNEK)>
      040224 040 103 110 T43MSG:: .ASCIZ / CHECK ERR INTERRUPT (QNEK)/
3228 ;*****
3229 ;**
3230 ;
3231 ;TEST DESCRIPTION-
3232 ; CHECKS THAT QNEK CAN CAUSE INTERRUPT
3233 ;TEST STEPS-
3234 ; SET INTERRUPT VECTOR
3235 ; SET QINT ERRIE BIT
3236 ; SET QMC = -1
3237 ; SET QBA = 160000 (ILLEGAL ADDRESS)
3238 ; SET QDMA = 37400
3239 ; SET QDMA DMAWT, GO BITS
3240 ; CHECK FOR IMMEDIATE INTERRUPT (CAN BE 10 MICROSEC.)
3241 ; CHECK QINT ERR, QNEK BITS
3242 ; RESTORE INTERRUPT VECTOR
3243 ;
3244 ;--
3245 ;*****
3246
3247 040260 BGNTST
      040260
3248 040260 T43::
      040260 CKPNT
      040260 104421 TRAP C1RFLA ; PRINT OUT TEST TITLE IF PNT SET
      040262 022700 001000 CMP #PNT,RO ; IS PNT FLAG SET?
      040266 001012 BNE 644 ; 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,RO
      040306 104417 TRAP C1PNTF
      040310 062706 000006 ADD #6,SP
      040314 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
3249 040320 INIT
      040320 ;INITIALIZE DEVICE
      040320 012777 000100 141770 MOV #QDMHALT,QDMT1 ;INITIALIZE Q-BRIDGE
      040326 013777 002520 141762 MOV INITWORD,QDMT1 ;SET UP LOOPBACK OPTION
      040334 005077 141740 CLR QINT ;CLEAR QINT AFTER LOOPBACK ENABLED
      040340 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
3250 040344 SETVEC VECTOR,#L43,#PRI07 ; SET INTERRUPT VECTOR
      040344 012746 000340 MOV #PRI07,-(SP)
      040350 012746 040454 MOV #L43,-(SP)
      040354 013746 002512 MOV VECTOR,-(SP)
      040360 012746 000003 MOV #3,-(SP)
      040364 104437 TRAP C1SVEC
      040366 062706 000010 ADD #10,SP
3251 040372 052777 000100 141700 BIS #ERRIE,QINT ; SET ERRIE BIT IN QINT
3252 040400 012777 177777 141704 MOV #-1,QMC ; ALL 1'S TO QMC
3253 040406 012777 160000 141674 MOV #160000,QBA ; ILLEGAL ADDRESS 160000 TO QBA
3254 040414 012777 037400 141662 MOV #37400,QDMA ; 37400 TO QDMA
3255 040422 052777 000003 141654 BIS #DMAWT!GO,QDMA ; INITIATE INTERRUPT
3256 040430 000240 NOP ; SMALL TIME DELAY
3257 040432 000240 NOP
3258 040434 000240 NOP
3259

```


TEST43: CHECK ERR INTERRUPT (GNEX)

```

3260                                     ; ERROR CONDITION, INTERRUPT DID NOT WORK IF HERE
3261                                     ;
3262 040436 104456                       ERRHRD 1,MSG57           ; TELL OPERATOR ERROR
      040436 000001                     TRAP  C$ERRHRD
      040442 011101                     .WORD 1
      040444 000000                     .WORD MSG57
      040446 004737 013230             .WORD 0
      040452 000434                     JSR  PC,DUMP           ; DO OPTIONAL REGISTER DUMP
                                     BR  E43                   ; BRANCH OVER INTERRUPT AREA
3265                                     ;
3266                                     ; END ERROR CONDITION
3267                                     ; INTERRUPT SERVICE ROUTINE
3268                                     ;
3269 040454                               BGNSRV L43
      040454                               L43::
3270 040454 017737 141620 002526       MOV  @QINT,VAL           ; READ QINT REGISTER
3271 040462 012737 120000 002530       MOV  @GNEX!ERR,EXP       ; EXPECTED PATTERN TO EXP
3272 040470 005137 002530              COM  EXP                 ; COMPLEMENT EXP FOR CLEARING
3273 040474 043737 002530 002526       BIC  EXP,VAL            ; CLEAR UNTESTED BITS
3274 040502 005137 002530              COM  EXP                 ; RESTORE EXPECTED PATTERN
3275 040506 023737 002530 002526       CMP  EXP,VAL            ; COMPARE EXPECTED WITH VALUE
3276 040514 001410                     BEQ  204                 ; BRANCH IF OK
3277                                     ;
3278                                     ; ERROR CONDITION
3279                                     ;
3280 040516 012701 000002               MOV  #2,R1              ; REGISTER NUMBER TO R1
3281 040522 104456                       ERRHRD 2,MSG58,ERR4     ; TELL OPERATOR ERROR
      040522 000002                     TRAP  C$ERRHRD
      040524 000002                     .WORD 2
      040526 011205                     .WORD MSG58
      040530 011776                     .WORD ERR4
3282 040532 004737 013230             JSR  PC,DUMP           ; DO OPTIONAL REGISTER DUMP
3283                                     ;
3284                                     ; END ERROR CONDITION
3285                                     ;
3286 040536 012716 040544               204: MOV  #E43,(SP)      ; ADJUST STACK
3287 040542                               ENDSRV
      040542                               L10144:
      040542 000002                               RTI
3288 040544                               E43:
3289 040544 042777 000040 141544       BIC  @MQPFL,@QMT1       ; CLEAR MQPFL BIT IN QMT1
3290 040552 013700 002512             CLRVEC VECTOR           ; CLEAR VECTOR
      040552 104436                     MOV  VECTOR,R0
      040556 104436                     TRAP  C$CVEC
3291 040560                               L10143:
      040560 104401                     TRAP  C$ETST

```

TEST44: CHECK ERR INTERRUPT (UPWRFL)

```

3293 .SBTTL TEST44: CHECK ERR INTERRUPT (UPWRFL)
3294 040562 STATST <CHECK ERR INTERRUPT (UPWRFL)>
      040562 040 103 110 T44MSG:: .ASCIZ / CHECK ERR INTERRUPT (UPWRFL)/
3295 ;*****
3296 ;**
3297 ;TEST DESCRIPTION-
3298 ; CHECKS THAT UPWRFL CAN CAUSE INTERRUPT
3299 ; USED WOTO - UPWRFL LOOP-BACK
3300 ;TEST STEPS
3301 ; SET INTERRUPT VECTOR
3302 ; SET QINT ERRIE BIT
3303 ; SET QCTL THRENB
3304 ; WAIT FOR INTERRUPT ON TIME-OUT
3305 ; CHECKS QINT UPWRFL, ERR BITS
3306 ; RESTORE INTERRUPT VECTOR
3307 ;
3308 ;
3309 ;--
3310 ;*****
3311
3312 040620 BGNTST
      040620 T44::
3313 040620 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
      040620 104421 TRAP C0RFLA
      040622 022700 001000 CMP @PNT,R0 ; IS PNT FLAG SET?
      040626 001012 BNE 644 ; 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 C0PNTF
      040650 062706 000006 ADD @6,SP
      040654 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
3314 040660 INIT
      040660 012777 000100 141430 ;INITIALIZE DEVICE
      040666 013777 002520 141422 MOV @MHALT,@QMT1
      040674 005077 141400 MOV INITWORD,@QMT1
      040700 005037 002532 CLR @QINT ;INITIALIZE Q-BRIDGE
      040704 012746 000340 CLR ERROR1 ;SET UP LOOPBACK OPTION
      040710 012746 041032 SETVEC VECTOR,@L44,@PRI07 ;CLEAR QINT AFTER LOOPBACK ENABLED
      040714 013746 002512 MOV @PRI07,-(SP) ;CLEAR ERROR FLAG
      040720 012746 000003 MOV @L44,-(SP) ; SET INTERRUPT VECTOR
      040724 104437 TRAP C0SVEC
      040726 062706 000010 ADD @10,SP
3316 040732 052777 000100 141340 BIS @ERRIE,@QINT ; SET ERRIE BIT IN QINT
3317 040740 052777 000002 141334 BIS @THRENB,@QCTL ; SET THRENB IN QCTL
3318 ;
3319 ; NOW WAIT FOR MAXIMUM OF 8 SECONDS FOR WOTO INTERRUPT
3320 ;
3321 040746 012704 000360 MOV @240.,R4 ; NUMBER OF .025 SEC TO LOOP
3322 040752 104: DELAY 25. ; .025 SEC DELAY
3323 040752 012727 000031 MOV @25.,(PC)+
      040756 000000 .WORD 0
      040760 013727 002116 MOV L@DLY,(PC)+

```

Address	Instruction	Comments
040764	000000	
040766	005367 177772	
040772	001375	
040774	005367 177756	
041000	001367	
3324	041002	
	041002 104422	
3325	041004 001002	
3326	041006 005304	
3327	041010 001360	
3328		
3329		
3330		
3331	041012	
3332	041012	
	041012 104456	
	041014 000001	
	041016 011274	
	041020 000000	
3333	041022 004737 013230	
3334	041026	
	041026 104432	
	041030 000100	
3335		
3336		
3337		
3338		
3339	041032	
	041032	
3340	041032 017737 141242 002526	
3341	041040 012737 110000 002530	
3342	041046 005137 002530	
3343	041052 043737 002530 002526	
3344	041060 005137 002530	
3345	041064 023737 002530 002526	
3346	041072 001410	
3347		
3348		
3349		
3350	041074 012701 000002	
3351	041100	
	041100 104456	
	041102 000002	
	041104 011351	
	041106 011776	
3352	041110 004737 013230	
3353		
3354		
3355		
3356	041114 012716 041122	
3357	041120	
	041120	
	041120 000002	
3358	041122	
3359	041122	
	041122 013700 002512	
	041126 104436	

J13

CZFPA-AO CTRL BRIDGE INTFC DIAG MACRO M1200 09-JAN-84 12:33 PAGE 84-2

SEQ 0165

TEST44: CHECK ERR INTERRUPT (UPWRFL)

3360 041130
041130
041130 104401

L10145: ENDTST
TRAP C#ETST

TEST45: CHECK BR LEVEL

```

3362 .SBTTL TEST45: CHECK BR LEVEL
3363 041132 STATST <CHECK BR LEVEL>
3364 041132 040 103 110 T45MSG:: .ASCIZ / CHECK BR LEVEL/
3365
3366 ;*****
3367 ;**
3368 ;TEST DESCRIPTION-
3369 ; CHECKS FOR INTERRUPT WHEN PS IS BELOW DEVICE BR LEVEL
3370 ; CHECKS FOR NO INTERRUPT WHEN PS IS EQUAL OR ABOVE DEVICE LEVEL
3371 ;TEST STEPS-
3372 ; SET INTERRUPT VECTOR
3373 ; SET CPU PRIORITY = 7
3374 ; SET QVNT ACKIE BIT
3375 ; SET QCMD CMDACK BIT TO CAUSE INTERRUPT
3376 ; IF NOT INTERRUPT:
3377 ; CHECK FOR DEVICE LEVEL < OR = CPU LEVEL
3378 ; IF INTERRUPT:
3379 ; CHECK FOR DEVICE LEVEL > CPU LEVEL
3380 ; DROP CPU LEVEL AND REPEAT
3381 ; RESTORE INTERRUPT VECTOR
3382 ;
3383 ;--
3384 ;*****
3385 041152 BGNTST
3386 041152 T45::
3387 041152 CKPNT ; PRINT OUT TEST TITLE IF PNT SET
3388 041152 TRAP C:RFLA
3389 041154 022700 001000 CMP #PNT,R0 ; IS PNT FLAG SET?
3390 041160 001012 BNE 644 ; BRANCH IF NOT SET
3391 041162 012746 041132 MOV #T45MSG,-(SP)
3392 041166 012746 002270 MOV #PNTMSG,-(SP)
3393 041172 012746 000002 MOV #2,-(SP)
3394 041176 010600 MOV SP,R0
3395 041200 104417 TRAP C:PNTF
3396 041202 062706 000006 ADD #6,SP
3397 041206 005037 002532 644: CLR ERROR1 ; CLEAR ERROR COUNTER
3398 041212 INIT
3399 041212 ;INITIALIZE DEVICE
3400 041212 MOV #QHALT,BQMT1 ;INITIALIZE Q-BRIDGE
3401 041220 013777 002520 141076 MOV INITWORD,BQMT1 ;SET UP LOOPBACK OPTION
3402 041226 005077 141046 CLR BQINT ;CLEAR QINT AFTER LOOPBACK ENABLED
3403 041232 005037 002532 CLR ERROR1 ;CLEAR ERROR FLAG
3404 041236 SETVEC VECTOR,#L45,BRLEVEL ; SET INTERRUPT VECTOR
3405 041236 013746 002516 MOV BRLEVEL,-(SP)
3406 041242 012746 041432 MOV #L45,-(SP)
3407 041246 013746 002512 MOV VECTOR,-(SP)
3408 041252 012746 000003 MOV #3,-(SP)
3409 041256 104437 TRAP C:SVEC
3410 041260 062706 000010 ADD #10,SP
3411 041264 012737 000340 002540 104: MOV #PRI07,VALUE ; STORE CURRENT CPU LEVEL
3412 041272 SETPRI VALUE ; SET UP CURRENT CPU PRIORITY LEVEL
3413 041272 013700 002540 MOV VALUE,R0
3414 041276 104441 TRAP C:SPRI
3415 041300 005037 002542 CLR INTFLG ; CLEAR INTRRRUPT FLAG
3416 041304 052777 000100 140764 BIS #ACKIE,BQVNT ; SET ACKIE BIT IN QVNT

```

TEST45: CHECK BR LEVEL

```

3394 041312 052777 000001 140754      BIS      #CMDACK,BQCMD      ; 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$ERRHRD
      041342 000001      .WORD      1
      041344 011434      .WORD      MSG61
      041346 012160      .WORD      ERR6
3407 041350 000410      BR      30$      ; 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$ERRHRD
      041364 000002      .WORD      2
      041366 011521      .WORD      MSG62
      041370 012160      .WORD      ERR6
3417
3418      ; END ERROR CONDITION, TRY NEXT LOWER CPU LEVEL
3419
3420 041372
3421 041372 012777 000001 140676      MOV      #VNTPRS,BQVNT      ; RESET VNTACK FOR NEXT LOOP
3422 041400 162737 000040 002540      SUB      #PRI01,VALUE      ; TRY NEXT CPU LEVEL
3423 041406 100401                      BMI      40$      ; IF DONE, BRANCH
3424 041410 000730      BR      10$      ; GO DO NEXT TEST
3425
3426      ; END OF TEST, RESTORE
3427
3428 041412
3429 041412 013700 002512      CLRVEC  VECTOR      ; CLEAR VECTOR
      041412 104436      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						

TEST45: CHECK BR LEVEL

```

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
55 041442      .WORD L10151-L#HARD/2
56 041444      L#HARD::
57 041444      GPRMA G1,0,0,0,177776,YES      ;BASE ADDRESS
58 041444      .WORD T#CODE
59 041446      .WORD G1
60 041450      .WORD T#LOLIM
61 041452      .WORD T#HILIM
62 041454      GPRMA G2,2,0,0,1000,YES      ;VECTOR
63 041454      .WORD T#CODE
64 041456      .WORD G2
65 041460      .WORD T#LOLIM
66 041462      .WORD T#HILIM
67 041464      GPRML G3,4,1,YES      ;LOOPBACK CABLE INSTALLED?
68 041464      .WORD T#CODE
69 041466      .WORD G3
70 041470      .WORD 1
71 041472      GPRMD G4,6,0,340,0,7,YES      ;BR LEVEL
72 041472      .WORD T#CODE
73 041474      .WORD G4
74 041476      .WORD 340
75 041500      .WORD T#LOLIM
76 041502      .WORD T#HILIM
77
84          .ENDHRD
          .EVEN
          L10151:
          111 G1: .ASCIZ /BRIDGE MODULE BASE ADDRESS/
          111 G2: .ASCIZ /BRIDGE MODULE VECTOR /
          117 G3: .ASCIZ /LOOPBACK CABLE INSTALLED? /
          040 G4: .ASCIZ /BR LEVEL /
          .EVEN

```

SOFTWARE PARAMETER CODING SECTION

```

87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102 041660
103 041660
104
111
112 042060
      042060 042100
      042062 000006
      042064
113
114 042064
115 042064
      042064 000000
      042066 000004
      042070
116 042070 160500
117 042072 000500
118 042074 000000
119 042076 000200
120 042100
      042100
121 042100
122      000001

```

.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
      .WORD T$FREE
      .WORD T$SIZE
L$LAST::
;HARDCODED P-TABLE
      BGNSETUP 1
      BGNPTAB
      .WORD 0
      .WORD L10154-./2-1
L10152:
      .WORD QCMDAD
      .WORD VEC
      .WORD NO
      .WORD $B10000000
      ENDP TAB
L10154:
      ENDSETUP
.END

```

```

;HARDWARE UNIT 0 DEFAULT BASE ADDRESS
;HARDWARE UNIT 0 DEFAULT VECTOR
;LOOPBACK CABLE INSTALLED DEFAULT
;DEFAULT BR INTERRUPT LEVEL 4

```


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				F#DU	=	000016		INTBIT		002712	G
BITS#		012506	G	C#EDIT-	000003	ERR2		011702	G	F#END	=	000041		INTFLG		002542	G
BIT#		012466	G	C#ERDF-	000055	ERR4		011776	G	F#HARD-	=	000004		ISR	=	000100	G
BIT0	=	000001	G	C#ERMR-	000056	ERR5		012136	G	F#HM	=	000013		IXE	=	004000	G
BIT00	=	000001	G	C#ERRO-	000060	ERR6		012160	G	F#INIT-	=	000006		I#AU	=	000041	
BIT01	=	000002	G	C#ERSF-	000054	ERR7		012246	G	F#JMP	=	000050		I#AUTO-	=	000041	
BIT02	=	000004	G	C#ERSQ-	000057	EVL	=	000004	G	F#MOD	=	000000		I#CLN	=	000041	
BIT03	=	000010	G	C#ESCA-	000010	EXP		002530	G	F#MSG	=	000011		I#DU	=	000041	
BIT04	=	000020	G	C#ESEG-	000005	EXPECT		002354	G	F#PROT-	=	000021		I#HRD	=	000041	
BIT05	=	000040	G	C#ESUB-	000003	EXQBA		002370	G	F#PWR	=	000017		I#INIT-	=	000041	
BIT06	=	000100	G	C#ETST-	000001	EXQCMD		002354	G	F#RPT	=	000012		I#MOD	=	000040	
BIT07	=	000200	G	C#EXIT-	000032	EXQCTL		002362	G	F#SEG	=	000003		I#MSG	=	000041	
BIT08	=	000400	G	C#GETB-	000026	EXQDMA		002364	G	F#SOFT-	=	000005		I#PROT-	=	000040	
BIT09	=	001000	G	C#GETW-	000027	EXQINT		002360	G	F#SRV	=	000010		I#PTAB-	=	000041	
BIT1	=	000002	G	C#GMAN-	000043	EXQMT0		002374	G	F#SUB	=	000002		I#PWR	=	000041	
BIT10	=	002000	G	C#GPHR-	000042	EXQMT1		002376	G	F#SW	=	000014		I#RPT	=	000041	
BIT11	=	004000	G	C#GPLD-	000030	EXQUBA		002366	G	F#TEST-	=	000001		I#SEG	=	000041	
BIT12	=	010000	G	C#GPRI-	000040	EXQVAD		002400	G	GO	=	000001	G	I#SETU-	=	000041	
BIT13	=	020000	G	C#INIT-	000011	EXQVDA		002402	G	G#CNT0-	=	000200		I#SRV	=	000041	
BIT14	=	040000	G	C#INLP-	000020	EXQVNT		002356	G	G#DELM-	=	000372		I#SUB	=	000041	
BIT15	=	100000	G	C#MANI-	000050	EXQMC		002372	G	G#DISP-	=	000003		I#TST	=	000041	
BIT2	=	000004	G	C#MEM	=	000031				G#EXCP-	=	000400		J#JMP	=	000167	
BIT3	=	000010	G	C#MSG	=	000023				G#HILI-	=	000002		KISARO-		172340	G
BIT4	=	000020	G	C#OPEN-	000034	E#END	=	002100		G#LOLI-	=	000001		KISDRO-		172300	G
BIT5																	

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	0\$APTS	= 000000	
L\$HPTP	002022	G	L10043	021242	L10134	037064	MSG24	006241	G	0\$AU	= 000000	
L\$HM	002260	G	L10044	021212	L10135	037354	MSG25	006267	G	0\$BGNR	= 000001	
L\$ICP	002104	G	L10045	021536	L10136	037344	MSG26	006315	G	0\$BGNS	= 000000	
L\$INIT	014466	G	L10046	021506	L10137	037646	MSG27	006342	G	0\$DU	= 000000	
L\$LADP	002026	G	L10047	022054	L10140	037636	MSG28	006367	G	0\$ERRT	= 000000	
L\$LAST	042064	G	L10050	022024	L10141	040222	MSG29	006430	G	0\$GNSW	= 000000	
L\$LOAD	002100	G	L10051	022452	L10142	040204	MSG3	004632	G	0\$POIN	= 000001	
L\$LUN	002074	G	L10052	022726	L10143	040560	MSG30	006471	G	0\$SETU	= 000001	
L\$PREV	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	MSG36	006777	G	PRI00	= 000000	G
L\$REV	002010	G	L10061	024114	L10152	042070	MSG37	007023	G	PRI01	= 000040	G
L\$RPT	014446	G	L10062	024544	L10154	042100	MSG38	007103	G	PRI02	= 000100	G
L\$SPC	002056	G	L10063	024336	L28	030426	MSG39	007172	G	PRI03	= 000140	G
L\$SPCP	002020	G	L10064	024412	L34	034462	MSG4	004672	G	PRI04	= 000200	G
L\$SPTP	002024	G	L10065	024466	L35	034774	MSG40	007537	G	PRI05	= 000240	G
L\$STA	002030	G	L10066	024542	L36	035472	MSG41	007423	G	PRI06	= 000300	G
L\$TEST	002114	G	L10067	025030	L37	036034	MSG42	007520	G	PRI07	= 000340	G
L\$TIML	002014	G	L10070	025000	L38	036512	MSG43	007556	G	PR\$IE	= 000100	G
L\$UNIT	002012	G	L10071	025330	L39	037060	MSG44	007626	G	P1	013720	
L10000	002270		L10072	025300	L40	037340	MSG45	007663	G	P2	013772	
L10001	011774		L10073	025604	L41	037632	MSG46	007775	G	P37	036372	
L10002	012134		L10074	025554	L42	040116	MSG47	010077	G	QBA	002310	G
L10003	012156		L10075	026074	L43	040454	MSG48	010140	G	QBA00	= 000001	G
L10004	012244		L10076	026044	L44	041032	MSG49	010177	G	QBA01	= 000002	G
L10005	012364		L10077	026376	L45	041432	MSG5	010247	G	QBA02	= 000004	G
L10006	014444		L10100	026346	MAXUNI	= 000010	MSG50	010337	G	QBA03	= 000010	G
L10007	014456		L10101	026714	MEMSIZ	= 002550	MSG51	010407	G	QBA04	= 000020	G
L10011	015036		L10102	026664	MES10	012766	MSG52	010477	G	QBA05	= 000040	G
L10012	015042		L10103	027254	MES20	013015	MSG53	010547	G	QBA06	= 000100	G
L10013	015120		L10104	027224	MES20L	= 000204	MSG54	010653	G	QBA07	= 000200	G
L10014	015130		L10105	027604	MIODIS	= 000200	MSG55	010723	G	QBA08	= 000400	G
L10015	015140		L10106	027554	MMINIT	014370	MSG56	011012	G	QBA09	= 001000	G
L10016	016020		L10107	030016	MMRO	= 177572	MSG57	011101	G	QBA10	= 002000	G
L10017	015424		L10110	030014	MMR3	= 172516	MSG58	011205	G	QBA11	= 004000	G
L10020	015506		L10111	030300	MMHALT	= 000100	MSG59	011274	G	QBA12	= 010000	G
L10021	015562		L10112	030276	MQPFL	= 000040	MSG6	004771	G	QBA13	= 020000	G
L10022	015636		L10113	031520	MRDY	= 001000	MSG60	011351	G	QBA14	= 040000	G
L10023	015712		L10114	031516	MSG1	004544	MSG61	011434	G	QBA15	= 100000	G
L10024	015774		L10115	032366	MSG10	005161	MSG62	011521	G	CSUS	003316	G
L10025	016256		L10116	033256	MSG11	005224	MSG7	005012	G	QCND	002274	G
L10026	016166		L10117	033562	MSG12	005266	MSG8	005054	G	QCNDAD	= 160500	G
L10027	016254		L10120	034024	MSG13	005330	MSG9	005116	G	QCTL	002302	G
L10030	016750		L10121	034334	MSG14	005373	MT18	003161	G	QDMA	002304	G
L10031	017256		L10122	034646	MSG15	005436	MT1BIT	003161	G	QINT	002300	G
L10032	017226		L10123	034644	MSG16	005531	MVADR	= 000002	G	QMT0	002314	G
L10033	017602		L10124	035166	MSG17	005622				QMT1	002316	G
L10034	017552		L10125	035164	MSG18	005715				QNEK	= 020000	G

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\$NSO	=	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	RPO	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\$SEK0	=	010000	T17.2	024340	T35.1	034772			
QVAD08	=	000400	G	SELUVA	=	000014	G	T\$SIZE	=	000006	T17.3	024414	T36	035226	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	SQMT0	002344	G		T\$TAU	=	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\$HM	=	010000	T20.1	025462	T40MSG	037076	G		
QVDA11	=	004000	G	STATU	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	=	000000			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	THRENB	=	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		

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	WDT0	= 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

ZFPAA0.BIC,CZFPAA0/-SP=SVC/ML,ZFPAA01,ZFPAA02,ZFPAA03,ZFPAA04,ZFPAA05,ZFPAA06