esa

DOCUMENT

document title/ titre du document

COBS PATCH FOR UVCS SAFING (OCD 2410)

prepared by/*préparé par*	Ton van Overbeek
reference/*réference*	SOHO/PRG/TN/700
issue/*édition*	1
revision/*révision*	
date of issue/*date d'édition*	2010-02-02
status/*état*	Authorised
Document type/*type de document*	TN
Distribution/*distribution*	

European Space Agency
Agence spatiale européenne

SOHO Project Office
NASA - Goddard Space Flight Center - Mailcode 444 - Greenbelt, MD 20771 - USA
Tel. (1) 301 286 5963 - Fax (1) 301 286 1602 - www.esa.int

CHANGE LOG

reason for change /*raison du changement*	issue/*issue*	revision/*revision*	date/*date*
First draft	1	Draft	2009-10-30
Final issue	1		2010-02-02

CHANGE RECORD

Issue: 1 Revision:

reason for change/*raison du changement*	page(s)/*page(s)*	paragraph(s)/*paragraph(s)*
Updated after initial testing	All	

Table Of Contents

1 INTRODUCTION 4

2 IMPLEMENTATION 4
2.1 Experiment Off-pointing Function 5
2.2 ESR Warning Function 6

3 PATCH PROCEDURE 8

4 MODULE TESTING THE PATCH 11
4.1 Experiment off pointing function 11
4.2 ESR warning function 12

5 TEST ON THE SPACECRAFT SIMULATOR 14

6 APPENDIX 17
6.1 Setup1.cmd 17
6.2 Setup2.cmd 18
6.3 Test1a.cmd 20
6.4 Test1b.cmd 21
6.5 Test1c.cmd 23
6.6 Test2a.cmd 25
6.7 Test2b.cmd 26
6.8 Test2c.cmd 28
6.9 Test1a.log 29
6.10 Test1b.log 39
6.11 Test1c.log 44
6.12 Test2a.log 54
6.13 Test2b.log 61
6.14 Test2c.log 66

1 INTRODUCTION

When the COBS experiment off-pointing function triggers UVCS is sent both the primary and redundant 'Safe mode' relay command (UCMDSFN and UCMDSFR). If the ESR warning function triggers UVCS is sent the 'ESR Warning' OBDH block command.

In OCD 2410 UVCS requests to change this so that for both cases (experiment off-pointing and ESR warning) the following sequence of commands is sent:

- UBMDPRIM Preops (immediate)
- UBFQBAT Flush batch queue
- UBFQINT Flush internal queue
- UBAB Abort
- UCMDPAN Enter patrol mode (nominal)

This will make UVCS thermally more robust in case of ESR or off-pointing during long gaps.

This note describes the patches needed to both COBS functions for this change.

2 IMPLEMENTATION

The specified command sequence is implemented as on-board macro 21. Macro 21 is located in the area 0xACD4 – 0xACDD and consists of six TCM entries:

Address	TCM size and pointer
0xACD4-0xACD5	0x0800D363
0xACD6-0xACD7	0x0A00D3A4
0xACD8-0xACD9	0x0A00D3A9
0xACDA-0xACDB	0x0A00D3AE
0xACDC-0xACDD	0x0A00D3B3
0xACDE-0xACDF	0x0800D3B8

The six TCM definitions are:

Address	TCM definition	Mnemonic
0xD363-0xD366	0x0008,0x8000,0x00E9,0x0000	DZDUMMY
0xD3A4-0xD3A8	0x000A,0x4680,0x3782,0xE300,0x1A82	UBMDPRIM
0xD3A9-0xD3AD	0x000A,0x4680,0x3782,0xEA02,0x2184	UBFQBAT
0xD3AE-0xD3B2	0x000A,0x4680,0x3782,0xEA01,0x2183	UBFQINT
0xD3B3-0xD3B7	0x000A,0x4680,0x3782,0xE700,0x1E82	UBAB
0xD3B8-0xD3BA	0x0008,0x8000,0x0048,0x0239	UCMDPAN

In the initial implementation the DZDUMMY command was missing. This caused problems in the case of executing the experiment off-pointing function with both MDI and UVCS as receivers. This is explained below in section 4.1.

For both the experiment off-pointing function and the ESR warning function we now have to trigger the execution of macro 21 for UVCS.

2.1 Experiment Off-pointing Function

In case of the experiment off-pointing function the warning is sent to UVCS by the procedure WarnUVCS in the expoffpointing module (source file expoffpointing_1.ada of the gyroless software). The assembly code of this function looks as follows:

```
-------------------------------------------------------------------------------
   67 |   195 |     procedure WarnUVCS is
   67 |   196 |
   67 |   197 |         -- Send off-pointing warning to UVCS: Enter safe mode.
   67 |   198 |
-------------------------------------------------------------------------------

       9643   '                      WARNUVCS_$EXS03C
       9643   '  9F 3 3                      PSHM      3,3

-------------------------------------------------------------------------------
   68 |   199 |     begin
   68 |   200 |
   69 |   201 |         GyrolessWires.PutPower (GyrolessWires.UCMDSFN);  -- Safe mode, Prim
-------------------------------------------------------------------------------

69.    9644   '  85 3 0    001A   '          LIM       3,16#1A#
       9646   '  7E F 0    90C8   X          SJS       15,PUTPOWER_$GYT02W

-------------------------------------------------------------------------------
   70 |   202 |         GyrolessWires.PutPower (GyrolessWires.UCMDSFR);  -- Safe mode, Red
   70 |   203 |
-------------------------------------------------------------------------------

70.    9648   '  85 3 0    001B   '          LIM       3,16#1B#
       964A   '  7E F 0    90C8   X          SJS       15,PUTPOWER_$GYT02W

-------------------------------------------------------------------------------
   71 |   204 |     end;
   71 |   205 |
   71 |   206 | ---------------------------------------------------------------------------------
   71 |   207 |
-------------------------------------------------------------------------------

71.    964C   '  8F 3 3              I#9L    POPM      3,3
       964D   '  7F F 0                      URS       15
```

Here the needed patch is simple: replace the first call to GyrolessWires.PutPower with a call to IF_Macro.Trigg(21) and replace the second call to GyrolessWires.PutPower with NOP instructions. The code area 0x9644-0x964B now becomes:

```
9644  8530  LIM 3, 16#15#     ; Macro 21
9645  0015
9646  7EF0  SJS 15, IF_MACRO.TRIGG_$IGE02V
9647  6872
9648  FF00  NOP
9649  FF00  NOP
964A  FF00  NOP
964B  FF00  NOP
```

2.2 ESR Warning Function

When an ESR is detected by the COBS it sends the ESR Warning OBDH block command to the selected experiments. This is done by the function PutWarning in the Exp module (source file exp_4.ada).

```
------------------------------------------------------------------------------
--                                PutWarning                                --
------------------------------------------------------------------------------
--
--
--      Send ESR Warning message to all experiments.
--

    procedure  PutWarning is

        Dum:    IF_OBDH_BlkQ.BlkId_t;           -- Block id of command sent
                                                -- (not used since status never
                                                -- inquired).
    begin

        for  ExpId in ExpId_t range EIT..UVCS
        loop
            if WarRcvrs (ExpId)
            then
                Dum := IF_OBDH_BlkQ.Put (OBDH_BlkQ.ESRW, User(ExpId),
                                         PutWarningCmd,  PutWarningDat);
            end if;
        end loop;

    end PutWarning;
```

The code loops over the experiments and sends the command if the boolean WarRcvrs(ExpId) is true. The type ExpId_t is defined as:

```
type ExpId_t is (None, EIT, CDS, CELIAS, CEPAC, GOLF, LASCO, MDI,
SUMER, SWAN, UVCS, VIRGO, DumC, DumD, DumE, DumF);
```

This corresponds to numerical values 0 for None, 1 for EIT, …, 9 for SWAN, 10 for UVCS, … and 15 for DumF. WarRcvrs is an array of 16 booleans indexed by ExpId. The code ends the loop at UVCS, since VIRGO cannot receive OBDH block commands.
The patch changes the last loop value to 9 (= SWAN) and then treats UVCS separately after the loop by jumping to new code to trigger macro 21 when WarRcvrs(UVCS) is true. The original assembly code looks as follows:

```
52 |   326 | ------------------------------------------------------------------------------
52 |   327 | --                                PutWarning                                --
52 |   328 | ------------------------------------------------------------------------------
52 |   329 | --
52 |   330 | --
52 |   331 | --      Send ESR Warning message to all experiments.
52 |   332 | --
52 |   333 |
52 |   334 |
53 |   335 |     procedure  PutWarning is
53 |   336 |
```

```
   53 |   337 |
-------------------------------------------------------------------------------

53.    0003   '                                ORG       16#0#
       0000   '                        PUTWARNING_$EXQ02Y
       0000   '  9F 2 C                         PSHM      2,12
       0001   '  85 F F    FFFE   *             LIM       15,-16#2#,15

-------------------------------------------------------------------------------
   54 |   338 |         Dum:    IF_OBDH_BlkQ.BlkId_t;           -- Block id of command sent
   54 |   339 |                                                 -- (not used since status never
   54 |   340 |                                                 -- inquired).
   54 |   341 |
   55 |   342 |     begin
   55 |   343 |
   56 |   344 |         for  ExpId in ExpId_t range EIT..UVCS
   56 |   345 |         loop
-------------------------------------------------------------------------------

56.    0003   '  82 C 0                         LISP      12,16#1#

-------------------------------------------------------------------------------
   57 |   346 |             if WarRcvrs (ExpId)
   57 |   347 |             then
-------------------------------------------------------------------------------

57.    0004   '                        I#4L
       0004   '  81 2 C                         LR        2,12
       0005   '  80 B 2    0000   X             L         11,WARRCVRS_$EXQ036,2
       0007   '  75 0B                          BEZ       I#12L

-------------------------------------------------------------------------------
   58 |   348 |                 Dum := IF_OBDH_BlkQ.Put (OBDH_BlkQ.ESRW, User(ExpId),
   58 |   349 |                                           PutWarningCmd,  PutWarningDat);
-------------------------------------------------------------------------------

58.    0008   '  85 6 0    0000   *             LIM       6,AGG$_$EXQ001
       000A   '  80 5 0    0000   X             L         5,PUTWARNINGCMD_$EXQ03A
       000C   '  80 4 2    0000   X             L         4,USER_$EXQ037,2
       000E   '  B1 3 3                         SR        3,3
       000F   '  7E F 0    0000   X             SJS       15,PUT_$OBE02Y
       0011   '  02 3 00                        STB       15,DUM

-------------------------------------------------------------------------------
   59 |   350 |             end if;
   60 |   351 |         end loop;
   60 |   352 |
-------------------------------------------------------------------------------

60.    0012   '  F2 C 9                 I#12L   CISP      12,16#A#
       0013   '  7B 14                          BGE       I#27L
       0014   '  A2 C 0                         AISP      12,16#1#
57.    0015   '  81 2 C                         LR        2,12
       0016   '  80 B 2    0000   X             L         11,WARRCVRS_$EXQ036,2
       0018   '  75 0B                          BEZ       I#23L
58.    0019   '  85 6 0    0000   *             LIM       6,AGG$_$EXQ001
       001B   '  80 5 0    0000   X             L         5,PUTWARNINGCMD_$EXQ03A
       001D   '  80 4 2    0000   X             L         4,USER_$EXQ037,2
       001F   '  B1 3 3                         SR        3,3
       0020   '  7E F 0    0000   X             SJS       15,PUT_$OBE02Y
       0022   '  02 3 00                        STB       15,DUM
60.    0023   '  F2 C 9                 I#23L   CISP      12,16#A#
       0024   '  7B 03                          BGE       I#27L
       0025   '  A2 C 0                         AISP      12,16#1#
       0026   '  74 DE                          BR        I#4L
-------------------------------------------------------------------------------

61.    0027   '  85 F F    0002   *  I#27L     LIM       15,16#2#,15
       0029   '  8F 2 C                         POPM      2,12
```

```
002A   '  7F F 0                        URS       15
002B   '                      I#2BL
                                        END
```

Note that the Ada compiler has done loop unrolling and the code inside the loop is duplicated. The end of loop check (for the value 10 = 0xA = UVCS) is there twice: once at offset 0x0012 and once at offset 0x0023 (labels I#12L and I#23L). The labels I#12L and I#23L correspond to absolute addresses 0x56D4 and 0x56E5 respectively. These two CISP 12,16#A# (= 0xF2C9) have to be replaced by CISP 12,9 (= 0xF2C8).
After the loop the LIM 15,2,15 instruction at offset 0x0027 (= label I#27L = address 0x56E9) has to be overwritten by a jump to the new code.
We locate this code at address 0x5830, which is inside the ESR_GyroSet module (dead code in the gyroless). ESR_GyroSet starts at 0x57E2, but the initial part is reserved for the patch using thermal control circuits for temperature standard monitoring. The new code looks as follows:

```
                            ORG   16#5830#
5830   82 2 9               LISP    2,10                    -- 10 = UVCS
5831   80 B 2  CE5E         L       11,WARNING_RECEIVERS,2  -- UVCS on warning list ?
5833   75 05                BEZ     Out                     -- No
5834   85 3 0  0015         LIM     3,21                    -- Macro parameter for IF_Macro.Trigg
5836   7E F 0  6872         SJS     15, IF_MACRO_TRIGG      -- Trigger Macro 21
5838   85 F F  0002   Out   LIM     15,2,15                 -- Overwritten instruction
583A   8F 2 C               POPM    2,12                    -- Exit Exp.PutWarning here since no
583B   7F F 0               URS     15                      -- space saved when jumping back
```

With this new code the PutWarning function now looks as follows in Ada:

```
procedure  PutWarning is

    Dum:    IF_OBDH_BlkQ.BlkId_t;           -- Block id of command sent
                                            -- (not used since status never
                                            -- inquired).
begin

    for  ExpId in ExpId_t range EIT..SWAN
    loop
        if WarRcvrs (ExpId)
        then
            Dum := IF_OBDH_BlkQ.Put (OBDH_BlkQ.ESRW, User(ExpId),
                                     PutWarningCmd,  PutWarningDat);
        end if;
    end loop;

    if WarRcvrs (UVCS)
    then
        IF_Macro.Trigg(21);
    end if;

end PutWarning;
```

Changed/Added code in ***Italic Bold***.

3 PATCH PROCEDURE

- Verify assumptions: COBS loaded with all patches (Gyroless and TCM in Macros are prerequisites for this patch) and macro 21 defined as described above.

- Disable memory protection of memory areas 0x5xxx and 0x9xxx, `KNK34000,0x01DF,` and verify memory protection disabled in telemetry.
- Select group 4 and set memory dump start address to 0x9644:
  `KNK20000,0x0004,0x0000`
  `KNK2F000,0x9644`.
- Verify existing code in WarnUVCS:

Word in dump	*Address (hex)*	*Value (hex)*
DAT_1	9644	8530
DAT_2	9645	001A
DAT_3	9646	7EF0
DAT_4	9647	90C8
DAT_5	9648	8530
DAT_6	9649	001B
DAT_7	964A	7EF0
DAT_8	964B	90C8

- Replace with new code to trigger Macro 21:
  `KNK2C000,0x9645,0x0015,0x7EF0,0x6872,0xFF00,0xFF00,0xFF00,0xFF00`
- Verify new code:

Word in dump	*Address (hex)*	*Value (hex)*
DAT_1	9644	8530
DAT_2	9645	0015
DAT_3	9646	7EF0
DAT_4	9647	6872
DAT_5	9648	FF00
DAT_6	9649	FF00
DAT_7	964A	FF00
DAT_8	964B	FF00

- Put start address of memory dump at 0x5830: `KNK2F000,0x5830.`
- Verify existing code of the unused function ESR_GyroSet.Process:

Word in dump	*Address (hex)*	*Value (hex)*
DAT_1	5830	F600
DAT_2	5831	CE86
DAT_3	5832	76FC
DAT_4	5833	7EF0
DAT_5	5834	5673
DAT_6	5835	8122
DAT_7	5836	7A0E
DAT_8	5837	B155
DAT_9	5838	8240
DAT_10	5839	8530
DAT_11	583A	799E

DAT_12	583B	7EF0

- Replace with new code for Exp.PutWarning:
  `KNK2C000,0x5830,0x8229,0x80B2,0xCE5E,0x7505,0x8530,0x0015,0x7EF0,0x6872`
  `KNK2C000,0x5838,0x85FF,0x0002,0x8F2C,0x7FF0`
- Verify new code in memory dump:

Word in dump	*Address (hex)*	*Value (hex)*
DAT_1	5830	8229
DAT_2	5831	80B2
DAT_3	5832	CE5E
DAT_4	5833	7505
DAT_5	5834	8530
DAT_6	5835	0015
DAT_7	5836	7EF0
DAT_8	5837	6872
DAT_9	5838	85FF
DAT_10	5839	0002
DAT_11	583A	8F2C
DAT_12	583B	7FF0

- Set memory dump at 0x56D4: `KNK2F000,0x56D4`.
- Verify existing code:

Word in dump	*Address (hex)*	*Value (hex)*
DAT_1	56D4	F2C9
…		
DAT_18	56E5	F2C9
…		
DAT_22	56E9	85FF
DAT_23	56EA	0002

- Replace UVCS (10) with SWAN (9) as loop boundary and jump to new UVCS code after the loop:
  `KNK2C000,0x56D4,0xF2C8`
  `KNK2C000,0x56E5,0xF2C8`
  `KNK2C000,0x56E9,0x7070,0x5830`
- Verify new code:

Word in dump	*Address (hex)*	*Value (hex)*
DAT_1	56D4	F2C8
…		
DAT_18	56E5	F2C8
…		
DAT_22	56E9	7070
DAT_23	56EA	5830

- Restore memory protection, `KNK34000,0x03FF,` and verify.

4 MODULE TESTING THE PATCH

The module test of the patch consists of two independent parts: one for the experiment off-pointing function and one for the ESR warning function. The simulator of the TLD Ada Compilation System (the version used for building the COBS) is used for these tests.

4.1 Experiment Off-pointing function

In this test we run the patched WarnUVCS function and verify that the telecommands defined by macro 21 are put into the immediate telecommand buffer. This is followed by a single run of TC_Exec.Process which executes the first telecommand of the immediate telecommand buffer. TC_Exec.Process is executed later in the same 1 Hz software cycle as WarnUVCS.

In the original implementation of macro 21 the initial DZDUMMY was missing. This caused a conflict when both MDI and UVCS are experiment off-pointing receivers. The WarnMDI function puts the 'limb tracker open loop' OBDH block command in the normal OBDH block command queue. This queue has only one entry and is emptied by the 38 Hz high priority software cycle. When the TC_Exec.Process function tried to put the first UVCS OBDH block command (UBMDPRIM) into the normal queue an 'OBDH block queue full' error was generated and the command was not sent.
Hence the need for the initial DZDUMMY in order to delay the sending of the UBMDPRIM OBDH block command to the next 1 Hz cycle.

To verify and illustrate this we run three test cases:

1a: Only UVCS as receiver, Macro 21 with initial DZDUMMY
1b: All receivers, Macro 21 with initial DZDUMMY
1c: All receivers, Macro 21 without initial DZDUMMY

The test steps are:

- Load the COBS V14 code into the simulator followed by loading the code for the gyroless, intermittent recording and TCM in macros patches.
- Load and verify our patch.
- Set up macro 21 as defined in Section 2 for cases 1a and 1b, without DZDUMMY for case 1c.
- Dump the TCMs for macro 21 (memory area 0xD3A4 .. 0xD3BB for the UVCS commands, 0xD363 .. 0xD366 for DZDUMMY).
- Initialise stackpointer (R15) to 0xFFFF.
- Elaborate OBDH_Blkq to initialize the OBDH block command queues.
- Elaborate Exp to initialize the ESR warning command constants and set the default ESR warning receivers.

- Elaborate TC_Exec and the low level OBDH functions it calls (IL_Pkt and RTU).
- Put an immediate return in OBDH.Put since the OBDH hardware is not present.
- Set break point at end of PutWarnings (0x96CD).
- Set break points at WarnUVCS (0x9643), WarnLASCO (0x964E), WarnMDI (0x9658) and WarnSUMER (0x9663).
- Set break point at PutPower (0x90C8). This is the original function called by WarnUVCS and should not be reached when our patch is present.
- Set break point at ExtMacro.Trigg (0x9F11). This function is called by IF_Macro.Trigg. The break point is used to check that ExtMacro.Trigg is called with the correct parameter (= the macro number to execute). The expected value is 21 in register R3.
- Set the program counter to the start of PutWarnings (0x9672).
- Dump the first free index of the immediate telecommand queue (ImFFR at 0xABFD). Expected value is 0.
- Dump the start of the immediate telecommand queue (0xA621 .. A639). Expected to be all zeros.
- Set off pointing receivers for the specific case
- Run PutWarnings and verify the macro 21 tele commands appear in the immediate tele command queue and the applicable OBDH block commands appear in the ESR warning queue and normal queue.
- Run TC_Exec.Process and verify first command executes (cases 1a and 1b) or generates an error (case 1c).

The tests performed as expected. The telecommands defined by macro 21 were put correctly into the immediate telecommand queue. The ‘OBDH block command queue full’ error occurred in case 1c.

The command files and test logs for the three cases can be found in the Appendix.

4.2 ESR warning function

This test runs the patched ESR warning function (Exp.PutWarning). Also here we run three cases:

2a: Only UVCS as receiver, Macro 21 with initial DZDUMMY
2b: All receivers, Macro 21 with initial DZDUMMY
2c: All receivers, Macro 21 without initial DZDUMMY

The expected result is that the ESR warning OBDH block commands are put into the ESR warning OBDH block command queue for all instruments except UVCS and that the macro 21 telecommands are put into the immediate telecommand queue. A subsequent run of TC_Exec.Process should show no anomalies for all three cases.

The test steps are:

- Load the COBS V14 code into the simulator followed by loading the code for the gyroless, intermittent recording and TCM in macros patches.
- Load and verify our patch.

- Set up macro 21 as defined in Section 2 for cases 1a and 1b, without DZDUMMY for case 1c.
- Dump the TCMs for macro 21 (memory area 0xD3A4 .. 0xD3BB for the UVCS commands, 0xD363 .. 0xD366 for DZDUMMY).
- Initialise stackpointer (R15) to 0xFFFF.
- Elaborate OBDH_Blkq to initialize the OBDH block command queues.
- Elaborate Exp to initialize the ESR warning command constants and set the default ESR warning receivers.
- Elaborate TC_Exec and the low level OBDH functions it calls (IL_Pkt and RTU).
- Put an immediate return in OBDH.Put since the OBDH hardware is not present.
- Set breakpoint at end of original Exp.PutWarning (0x56EC). This breakpoint should not be reached.
- Set breakpoint at end of patched Exp.Putwarning (0x583B). This breakpoint should be reached.
- Set break point at ExtMacro.Trigg (0x9F11). This function is called by IF_Macro.Trigg. The break point is used to check that ExtMacro.Trigg is called with the correct parameter (= the macro number to execute). The expected value is 21 in register R3.
- Set program counter to start of Exp.PutWarning (0x56C2).
- Dump start of ESR warning queue (0xB14B .. 0xB16C) and verify all zeros.
- Dump the first free index of the immediate telecommand queue (ImFFR at 0xABFD). Expected value is 0.
- Dump the start of the immediate telecommand queue (0xA621 .. A639). Expected to be all zeros.
- Set and verify ESR warning receivers (0xCE5E .. 0xCE6D) as required by the test case.
- Run the program. Verify R3 = 21 at first break.
- Continue execution. Verify second break is at the end of our patched Exp.PutWarning (0x583B) and stackpointer has correct value (R15 = 0xFFFF).
- Dump start of ESR warning queue (0xB14B .. 0xB16C) and verify the ESR warning commands for each test case.
- Dump the first free index of the immediate telecommand queue (ImFFR at 0xABFD). Expected value is 24 (0x18) without DZDUMMY, 28 (0x1C) with DZDUMMY.
- Dump the start of the immediate telecommand queue (0xA621 .. 0xA639) and verify the macro 21 tele commands are present.
- Run TC_Exec.Process and verify for all cases that the first tele command executes without anomalies.

The three tests performed as expected. The ESR warning OBDH block commands were put correctly into the ESR warning queue and the five telecommands defined by macro 21 were put correctly into the immediate telecommand queue. The first macro 21 telecommand executed properly for all three cases (Dummy OBDH command for 2a and 2b, OBDH block command for case 2c).

Here there is no conflict with the normal OBDH block command queue since all receivers except UVCS use the ESR warning OBDH block command queue which has 16 entries.

The command files and the test logs for the three cases are in the Appendix.

5 TESTS ON THE SPACECRAFT SIMULATOR

After uploading the patch and macro 21 as defined in section 2 the following test cases were run:
Experiment Off-pointing:

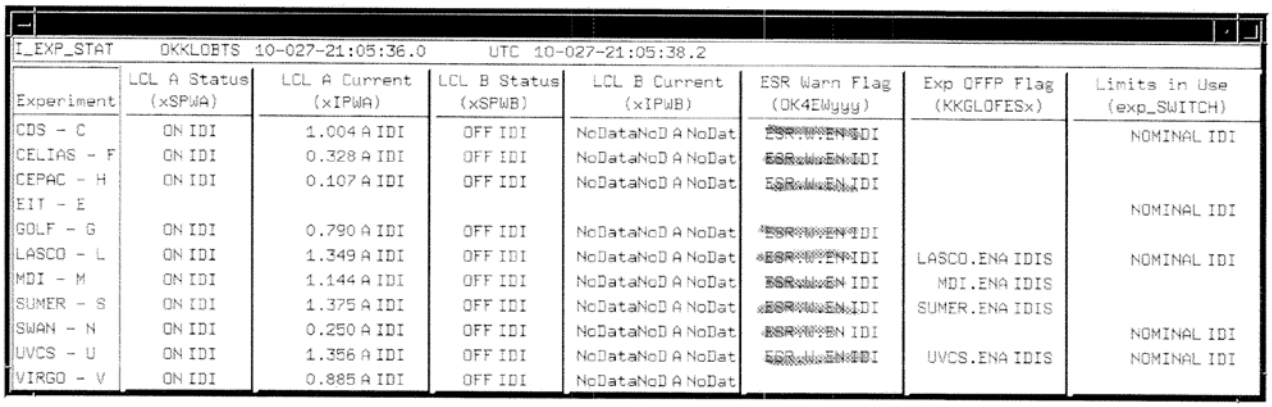
1. Only UVCS as receiver
2. LASCO, MDI and SUMER as receivers
3. UVCS, LASCO, MDI and SUMER as receivers

ESR Warning:

4. Only UVCS as receiver
5. All except UVCS as receivers
6. All receivers

The proper execution of the tele commands and resulting OBDH block commands were verified by checking the OBDH block command counters and dumping group 15 (anomaly table) and group 8 (TC execution acknowledge table).

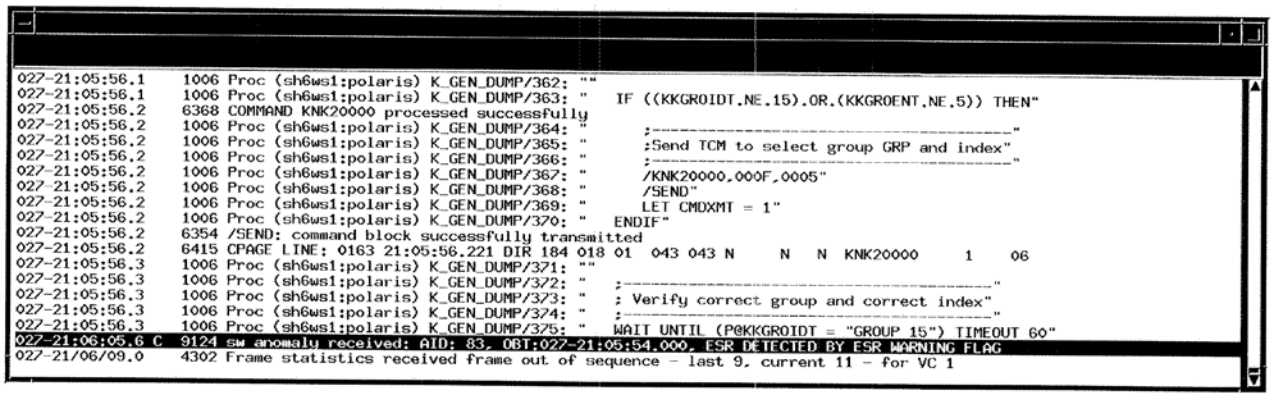
As an example here the screendumps from the last test (ESR Warning to all receivers).
The first one shows all possible ESR warning receivers enabled.



I_EXP_STAT OKKLOBTS 10-027-21:05:36.0 UTC 10-027-21:05:38.2

Experiment	LCL A Status (xSPWA)	LCL A Current (xIPWA)	LCL B Status (xSPWB)	LCL B Current (xIPWB)	ESR Warn Flag (OK4EWyyy)	Exp OFFP Flag (KKGLOFESx)	Limits in Use (exp_SWITCH)
CDS - C	ON IDI	1.004 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI		NOMINAL IDI
CELIAS - F	ON IDI	0.328 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI		
CEPAC - H	ON IDI	0.107 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI		
EIT - E							NOMINAL IDI
GOLF - G	ON IDI	0.790 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI		
LASCO - L	ON IDI	1.349 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI	LASCO.ENA IDIS	NOMINAL IDI
MDI - M	ON IDI	1.144 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI	MDI.ENA IDIS	
SUMER - S	ON IDI	1.375 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI	SUMER.ENA IDIS	
SWAN - N	ON IDI	0.250 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI		NOMINAL IDI
UVCS - U	ON IDI	1.356 A IDI	OFF IDI	NoDataNoD A NoDat	ESR.W.EN IDI	UVCS.ENA IDIS	NOMINAL IDI
VIRGO - V	ON IDI	0.885 A IDI	OFF IDI	NoDataNoD A NoDat			

The second one shows the reporting of the raising of the ESR warning flag at 21:05:54.000.



```
027-21:05:56.1    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/362: ""
027-21:05:56.1    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/363: "   IF ((KKGROIDT.NE.15).OR.(KKGROENT.NE.5)) THEN"
027-21:05:56.2    6368 COMMAND KNK20000 processed successfully
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/364: "      ;-----------------------------------------"
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/365: "      ;Send TCM to select group GRP and index"
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/366: "      ;-----------------------------------------"
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/367: "      /KNK20000,000F,0005"
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/368: "      /SEND"
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/369: "      LET CMDXMT = 1"
027-21:05:56.2    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/370: "   ENDIF"
027-21:05:56.2    6354 /SEND: command block successfully transmitted
027-21:05:56.2    6415 CPAGE LINE: 0163 21:05:56.221 DIR 184 018 01  043 043 N     N   N  KNK20000     1    06
027-21:05:56.3    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/371: ""
027-21:05:56.3    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/372: "   ;-----------------------------------------"
027-21:05:56.3    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/373: "   ; Verify correct group and correct index"
027-21:05:56.3    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/374: "   ;-----------------------------------------"
027-21:05:56.3    1006 Proc (sh6ws1:polaris) K_GEN_DUMP/375: "   WAIT UNTIL (P@KKGROIDT = "GROUP 15") TIMEOUT 60"
027-21:06:05.6 C  9124 SW anomaly received: AID: 83, OBT:027-21:05:54.000, ESR DETECTED BY ESR WARNING FLAG
027-21/06/09.0    4302 Frame statistics received frame out of sequence - last 9, current 11 - for VC 1
```

The third one shows the acknowledgment of the execution of the telecommands: The first command from macro 21 (DZDUMMY) got acknowledged at 21:05:54 (entry 7). The next one 2 seconds later (after the execution of the OBDH block command sequence) followed by commands 3, 4 and 5 each one second later. The last command (an OBDH relay command, entry 2) got acknowledged in the same TC_Exec.Process execution cycle as the previous (entry 3) OBDH block command.

```
K_GRP8_EXETABLE                    PACKET ID = SW
KKLOBTS            61f30979 IDI    KKLOBTF                    b000 IDI
KKGROIDT            GROUP 8 IDI    KKGROENT      0 IDI*
UTC        10-027-21:07:06.4       OKKLOBTS     10-027-21:07:05.0

                    *** EXECUTION ACKNOWLEDGE TABLE ***
```

ENTRY #	EXECUTED MESSAGE NAME				EXECUTED TIME	
0	KK08NA00	d02c IDI	KK08SE00	1643317603 Sec. IDI	OKK08SE00	10-027-21:06:43.0
1	KK08NA01	d02b IDI	KK08SE01	1643317569 Sec. IDI	OKK08SE01	10-027-21:06:09.0
2	KK08NA02	8000 IDI	KK08SE02	1643317559 Sec. IDI	OKK08SE02	10-027-21:05:59.0
3	KK08NA03	4680 IDI	KK08SE03	1643317559 Sec. IDI	OKK08SE03	10-027-21:05:59.0
4	KK08NA04	4680 IDI	KK08SE04	1643317558 Sec. IDI	OKK08SE04	10-027-21:05:58.0
5	KK08NA05	4680 IDI	KK08SE05	1643317557 Sec. IDI	OKK08SE05	10-027-21:05:57.0
6	KK08NA06	4680 IDI	KK08SE06	1643317556 Sec. IDI	OKK08SE06	10-027-21:05:56.0
7	KK08NA07	8000 IDI	KK08SE07	1643317554 Sec. IDI	OKK08SE07	10-027-21:05:54.0
8	KK08NA08	c52a IDI	KK08SE08	1643317552 Sec. IDI	OKK08SE08	10-027-21:05:52.0
9	KK08NA09	f0a9 IDI	KK08SE09	1643317537 Sec. IDI	OKK08SE09	10-027-21:05:37.0
10	KK08NA10	d7a8 IDI	KK08SE10	1643317484 Sec. IDI	OKK08SE10	10-027-21:04:44.0
11	KK08NA11	d027 IDI	KK08SE11	1643317483 Sec. IDI	OKK08SE11	10-027-21:04:43.0
12	KK08NA12	d026 IDI	KK08SE12	1643317370 Sec. IDI	OKK08SE12	10-027-21:02:50.0
13	KK08NA13	c525 IDI	KK08SE13	1643317330 Sec. IDI	OKK08SE13	10-027-21:02:10.0
14	KK08NA14	f0a4 IDI	KK08SE14	1643317275 Sec. IDI	OKK08SE14	10-027-21:01:15.0
15	KK08NA15	d023 IDI	KK08SE15	1643317179 Sec. IDI	OKK08SE15	10-027-20:59:39.0
16	KK08NA16	f0a2 IDI	KK08SE16	1643317153 Sec. IDI	OKK08SE16	10-027-20:59:13.0
17	KK08NA17	d021 IDI	KK08SE17	1643316954 Sec. IDI	OKK08SE17	10-027-20:55:54.0
18	KK08NA18	8000 IDI	KK08SE18	1643316898 Sec. IDI	OKK08SE18	10-027-20:54:58.0
19	KK08NA19	4680 IDI	KK08SE19	1643316898 Sec. IDI	OKK08SE19	10-027-20:54:58.0
20	KK08NA20	4680 IDI	KK08SE20	1643316897 Sec. IDI	OKK08SE20	10-027-20:54:57.0
21	KK08NA21	4680 IDI	KK08SE21	1643316896 Sec. IDI	OKK08SE21	10-027-20:54:56.0
22	KK08NA22	4680 IDI	KK08SE22	1643316895 Sec. IDI	OKK08SE22	10-027-20:54:55.0
23	KK08NA23	8000 IDI	KK08SE23	1643316893 Sec. IDI	OKK08SE23	10-027-20:54:53.0
24	KK08NA24	d620 IDI	KK08SE24	1643316891 Sec. IDI	OKK08SE24	10-027-20:54:51.0
25	KK08NA25	da1f IDI	KK08SE25	1643316859 Sec. IDI	OKK08SE25	10-027-20:54:19.0
26	KK08NA26	d01e IDI	KK08SE26	1643316764 Sec. IDI	OKK08SE26	10-027-20:52:44.0
27	KK08NA27	c51d IDI	KK08SE27	1643316743 Sec. IDI	OKK08SE27	10-027-20:52:23.0
28	KK08NA28	f09c IDI	KK08SE28	1643316689 Sec. IDI	OKK08SE28	10-027-20:51:29.0
29	KK08NA29	d79b IDI	KK08SE29	1643316584 Sec. IDI	OKK08SE29	10-027-20:49:44.0
30	KK08NA30	d01a IDI	KK08SE30	1643316583 Sec. IDI	OKK08SE30	10-027-20:49:43.0
31	KK08NA31	d019 IDI	KK08SE31	1643315590 Sec. IDI	OKK08SE31	10-027-20:33:10.0
32	KK08NA32	d018 IDI	KK08SE32	1643315541 Sec. IDI	OKK08SE32	10-027-20:32:21.0
33	KK08NA33	8000 IDI	KK08SE33	1643315531 Sec. IDI	OKK08SE33	10-027-20:32:11.0

The last screenshot shows the OBDH block command counters after test 6. The count for CDS, CELIAS, CEPAC, GOLF, and SWAN is at 2 (= two times the ESR warning flag, tests 5 and 6). The count for LASCO and SUMER is at 4 (= two times the ESR warning flag from the off-

pointing tests, cases 2 and 3, and two from the ESR warning flag tests, cases 5 and 6). The count for MDI is at 4 (two times the limb tracker open loop command from the off pointing tests, cases 2 and 3, and two from the ESR warning flag tests, cases 5 and 6). The UVCS counter is at 16 (Each macro 21 execution has 4 OBDH block commands. There were four macro 21 executions, cases 1, 3, 4 and 6). Note also that the time of the execution acknowledgment for the last UVCS command is 3 seconds after the last ESR warning command.

```
K_CMD_COUNTER                    Packet ID = SW          UTC   10-027-21:07:13.6
OKKLOBTS  10-027-21:07:05.0      KKLOBTS    61f30979 IDI  KKLOBTF        b000 IDI

      TC Frame Acc/Rej             TC Block Acc/Rej             TC Message Acc/Rej
KKATCCNT        149     IDI   KKATCBKC        185     IDI   KKTCMEXC        239     IDI
KKRTCCNT          0     IDI*  KKRTCBKC          0     IDI*  KKTCMRJC          0     IDI*

OBDH Block Command Status      OBDH BC STATUS (AOCS)        OBDH BC STATUS (CDS)
Acceptance Counter             KKBKAAOC          5     IDI  KKBKACDS          2     IDI
Rejection Counter              KKBKRAOC          0     IDI* KKBKRCDS          0     IDI*
Header of Last Block           KKBKHAOC        424     IDI  KKBKHCDS        10a2     IDI
Time of Last Block             KKBKOAOC   61f2f289     IDI  KKBKOCDS   61f30932     IDI
   OBDH BC STATUS (CELIAS)     OBDH BC STATUS (CEPAC)       OBDH BC STATUS (EIT)
KKBKACEL          2     IDI  KKBKACPC          2     IDI  KKBKAEIT          0     IDI*
KKBKRCEL          0     IDI* KKBKRCPC          0     IDI* KKBKREIT          0     IDI*
KKBKHCEL       14a2     IDI  KKBKHCPC       18a2     IDI  KKBKHEIT          0     IDI*
KKBKOCEL   61f30932     IDI  KKBKOCPC   61f30932     IDI  KKBKOEIT          0     IDI*
   OBDH BC STATUS (GOLF)       OBDH BC STATUS (LASCO)       OBDH BC STATUS (MDI)
KKBKAGLF          2     IDI  KKBKALAS          4     IDI  KKBKAMDI          4     IDI
KKBKRGLF          0     IDI* KKBKRLAS          0     IDI* KKBKRMDI          0     IDI*
KKBKHGLF       20a2     IDI  KKBKHLAS       24a2     IDI  KKBKHMDI       28a2     IDI
KKBKOGLF   61f30932     IDI  KKBKOLAS   61f30932     IDI  KKBKOMDI   61f30933     IDI
   OBDH BC STATUS (SUMER)      OBDH BC STATUS (SWAN)        OBDH BC STATUS (UVCS)
KKBKASUM          4     IDI  KKBKASWN          2     IDI  KKBKAUVC         16     IDI
KKBKRSUM          0     IDI* KKBKRSWN          0     IDI* KKBKRUVC          0     IDI*
KKBKHSUM       2ca2     IDI  KKBKHSWN       30a2     IDI  KKBKHUVC       3782     IDI
KKBKOSUM   61f30933     IDI  KKBKOSWN   61f30933     IDI  KKBKOUVC   61f30936     IDI
```

All simulator tests worked as expected. This in contrast with our first test run on the spacecraft simulator (without the initial DZDUMMY in macro 21) when we received the OBDH block queue full anomaly during the experiment off-pointing tests with both MDI and UVCS as receivers.

6 APPENDIX

6.1 Setup1.cmd

```
! Setup1.cmd: UVCS macro with leading DZDUMMY
attach tldmps
! Load COBS and patches
load [-.cobsv14]start
load [-.gyroless]gyroless_2
load [-.intrec]intrec_3
load [-.tcmmac]tcm_mac_2
! Load our patch
load uvcswarning.ldm
! Reload COBSV14 symbols to avoid warning messages
load/symbols [-.cobsv14]start
exam/ins 16#9643#..16#964D#
exam/ins 16#56D4#..16#56D5#
exam/ins 16#56E5#..16#56E6#
exam/ins 16#56E9#..16#56EC#
exam/ins 16#5830#..16#583B#
! Define macro offset table
mod/dec 16#795f#..+10#15#:=0,4,12,16,20,24,28,32,36,38,40,44,48,50,60,63
mod/dec 16#796f#..+4:=75,78,86,89,99
! Define entry 21 of macro size table
mod/dec 16#af2e#+10#20#:=6
! Define macro 21: DZDUMMY, UBMDPRIM, UBFQBAT, UBFQINT, UBAB, UCMDPAN
mod/hex
16#acd4#..+10#11#:=0800,D363,0A00,D3A4,0A00,D3A9,0A00,D3AE,0A00,D3B3,0800,D3B8
! Define TCMs for macro 21
! DZDUMMY
mod/hex 16#D363#..+3:=0008,8000,00E9,0000
! UBMDPRIM
mod/hex 16#D3A4#..+4:=000A,4680,3782,E300,1A82
! UBFQBAT
mod/hex 16#D3A9#..+4:=000A,4680,3782,EA02,2184
! UBFQINT
mod/hex 16#D3AE#..+4:=000A,4680,3782,EA01,2183
! UBAB
mod/hex 16#D3B3#..+4:=000A,4680,3782,E700,1E82
! UCMDPAN
mod/hex 16#D3B8#..+3:=0008,8000,0048,0239
!
! Dump TCMs
exam/hex 16#D363#..16#D366#
exam/hex 16#D3A4#..16#D3BB#
!
! Init stack pointer
mod/hex \rf:=ffff
!
! Elaborate OBDH_Blkq (Initialize OBDH block command queues)
set break 16#2793#
mod/hex \ic:=2730
```

```
go
wait
cancel break/all
!
! Elaborate Exp (Initialize ESR warning command constants and set default ESR
warning receivers)
set break 16#577c#
mod/hex \ic:=575c
go
wait
cancel break/all
!
! Elaborate TC_Exec
set break 16#48DA#
mod/hex \ic:=48B7
go
wait
cancel break/all
!
! Elaborate IL_Pkt
set break 16#1EF8#
mod/hex \ic:=1EF6
go
wait
cancel break/all
!
! Elaborate RTU
set break 16#162E#
mod \ic:=16#161F#
go
wait
cancel break/all
!
! Put an immediate return in OBDH.Put
mod/hex 16#131D#:=7FF0
!
exit
```

6.2 Setup2.cmd

```
! Setup2.cmd: UVCS macro without leading DZDUMMY
attach tldmps
! Load COBS and patches
load [-.cobsv14]start
load [-.gyroless]gyroless_2
load [-.intrec]intrec_3
load [-.tcmmac]tcm_mac_2
! Load our patch
load uvcswarning.ldm
! Reload COBSV14 symbols to avoid warning messages
load/symbols [-.cobsv14]start
! Verify our patch
exam/ins 16#9643#..16#964D#
exam/ins 16#56D4#..16#56D5#
```

```
exam/ins 16#56E5#..16#56E6#
exam/ins 16#56E9#..16#56EC#
exam/ins 16#5830#..16#583B#
! Define macro offset table
mod/dec 16#795f#..+10#15#:=0,4,12,16,20,24,28,32,36,38,40,44,48,50,60,63
mod/dec 16#796f#..+4:=75,78,86,89,99
! Define entry 21 of macro size table
mod/dec 16#af2e#+10#20#:=5
! Define macro 21: UBMDPRIM, UBFQBAT, UBFQINT, UBAB, UCMDPAN
mod/hex 16#acd4#..+9:=0A00,D3A4,0A00,D3A9,0A00,D3AE,0A00,D3B3,0800,D3B8
! Define TCMs for macro 21
! UBMDPRIM
mod/hex 16#D3A4#..+4:=000A,4680,3782,E300,1A82
! UBFQBAT
mod/hex 16#D3A9#..+4:=000A,4680,3782,EA02,2184
! UBFQINT
mod/hex 16#D3AE#..+4:=000A,4680,3782,EA01,2183
! UBAB
mod/hex 16#D3B3#..+4:=000A,4680,3782,E700,1E82
! UCMDPAN
mod/hex 16#D3B8#..+3:=0008,8000,0048,0239
!
! Dump TCMs
exam/hex 16#D3A4#..16#D3BB#
!
! Init stack pointer
mod/hex \rf:=ffff
!
! Elaborate OBDH_Blkq (Initialize OBDH block command queues)
set break 16#2793#
mod/hex \ic:=2730
go
wait
cancel break/all
!
! Elaborate Exp (Initialize ESR warning command constants and set default ESR
warning receivers)
set break 16#577c#
mod/hex \ic:=575c
go
wait 1
cancel break/all
!
! Elaborate TC_Exec
set break 16#48DA#
mod/hex \ic:=48B7
go
wait
cancel break/all
!
! Elaborate IL_Pkt
set break 16#1EF8#
mod/hex \ic:=1EF6
go
wait 1
```

```
cancel break/all
!
! Elaborate RTU
set break 16#162E#
mod \ic:=16#161F#
go
wait
cancel break/all
!
! Put an immediate return in OBDH.Put
mod/hex 16#131D#:=7FF0
!
exit
```

6.3 Test1a.cmd

```
! Module test for UVCSwarning, Experiment offpointing part
! Test 1a Only UVCS as receiver, UVCS macro with DZDUMMY
!
set log test1a.log
@setup1.cmd
!
! Set break at end of PutWarnings
set break 16#96CD#
! Set break at WarnUVCS, WarnLASCO, WarnMDI and WarnSUMER
set break 16#9643#
set break 16#964E#
set break 16#9658#
set break 16#9663#
! Set break at PutPower (should not be reached)
set break 16#90C8#
! Set break at ExtMacro.Trigg (to verify macro parameter)
set break 16#9F11#
! Set program counter at start of PutWarnings
mod/hex \ic:=9672
! Dump first free index in immediate queue (ImFFR)
exam/hex 16#ABFD#
! Dump start of immediate queue
exam/hex 16#A621#..+18
! Set receivers to UVCS only
mod/hex 16#FC12#:=0020
! Run
go
wait
! Verify break at WarnUVCS
go
wait
! Verify R3 = 21 (0x15) at ExtMacro.Trigg
! Run till end of PutWarnings
go
wait
cancel break/all
! Dump first free index in immediate queue (ImFFR), should now be 28 (0x1C)
exam/hex 16#ABFD#
```

```
! Dump start of immediate queue (should now contain the macro 21 TCMs)
exam/hex 16#A621#..+1c
! Dump OBDH block queues. Should be empty (all zeros)
exam/hex 16#B14B#..16#B170#
!
! Now do one run of TC_Exec.Process (will execute first TC of the UVCS macro)
! Set break at TC_exec.Process return
set break 16#4823#
! Set break at DecodeDst
set break 16#47EC#
! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
set break 16#4780#
set break 16#479B#
! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and OBDH.Put
set break 16#1E89#
set break 16#1515#
set break 16#131D#
! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and Anomaly.Put
set break 16#25E6#
set break 16#10A8#
! Put IC at start of TC_Exec.Process
mod/hex \ic:=4803
! Run till DecodeDst
go
wait
! Run till DecodeOBDH_Msg
go
wait
! Run till IL_Pkt.Put
go
wait
! Run till RTU.PutRaw
go
wait
! Verify R3-R4 contain dummy interrogation 00E9,0000
! Run till OBDH.Put
go
wait
! Run till end of TC_Exec.Process
go
wait
!
exit
```

6.4 Test1b.cmd

```
! Module test for UVCSwarning, Experiment offpointing part
! Test 1b All warning receivers, UVCS macro with DZDUMMY
!
set log test1b.log
@setup1.cmd
!
! Set break at end of PutWarnings
set break 16#96CD#
```

```
! Set break at WarnUVCS, WarnLASCO, WarnMDI and WarnSUMER
set break 16#9643#
set break 16#964E#
set break 16#9658#
set break 16#9663#
! Set break at PutPower (should not be reached)
set break 16#90C8#
! Set break at ExtMacro.Trigg (to verify macro parameter)
set break 16#9F11#
! Set program counter at start of PutWarnings
mod/hex \ic:=9672
! Dump first free index in immediate queue (ImFFR)
exam/hex 16#ABFD#
! Dump start of immediate queue
exam/hex 16#A621#..+18
! Set receivers to UVCS, LASCO, MDI, SUMER
mod/hex 16#FC12#:=03A0
! Run
go
wait
! Verify break at WarnUVCS
go
wait
! Verify R3 = 21 (0x15) at ExtMacro.Trigg
! Verify the other 3 Warnxxx functions are called.
go
wait
go
wait
go
wait
! Run till end of PutWarnings
go
wait
cancel break/all
! Dump first free index in immediate queue (ImFFR), should now be 28 (0x1C)
exam/hex 16#ABFD#
! Dump start of immediate queue (should now contain the macro 21 TCMs)
exam/hex 16#A621#..+1c
! Dump OBDH block queues. Should contain 2 ESR warning commands (LASCO, SUMER)
! and one normal (MDI open loop)
exam/hex 16#B14B#..16#B170#
!
! Now do one run of TC_Exec.Process (will execute first TC of the UVCS macro)
! Set break at TC_exec.Process return
set break 16#4823#
! Set break at DecodeDst
set break 16#47EC#
! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
set break 16#4780#
set break 16#479B#
! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and OBDH.Put
set break 16#1E89#
set break 16#1515#
set break 16#131D#
```

```
! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and Anomaly.Put
set break 16#25E6#
set break 16#10A8#
! Put IC at start of TC_Exec.Process
mod/hex \ic:=4803
! Run till DecodeDst
go
wait
! Run till DecodeOBDH_Msg
go
wait
! Run till IL_Pkt.Put
go
wait
! Run till RTU.PutRaw
go
wait
! Verify R3-R4 contain dummy interrogation 00E9,0000
! Run till OBDH.Put
go
wait
! Run till end of TC_Exec.Process
go
wait
!
exit
```

6.5 Test1c.cmd

```
! Module test for UVCSwarning, Experiment offpointing part
! Test 1c All warning receivers, UVCS macro without DZDUMMY
!
set log test1c.log
@setup2.cmd
!
! Set break at end of PutWarnings
set break 16#96CD#
! Set break at WarnUVCS, WarnLASCO, WarnMDI and WarnSUMER
set break 16#9643#
set break 16#964E#
set break 16#9658#
set break 16#9663#
! Set break at PutPower (should not be reached)
set break 16#90C8#
! Set break at ExtMacro.Trigg (to verify macro parameter)
set break 16#9F11#
! Set program counter at start of PutWarnings
mod/hex \ic:=9672
! Dump first free index in immediate queue (ImFFR)
exam/hex 16#ABFD#
! Dump start of immediate queue
exam/hex 16#A621#..+18
! Set receivers to UVCS, LASCO, MDI, SUMER
mod/hex 16#FC12#:=03A0
```

```
! Run
go
wait
! Verify break at WarnUVCS
go
wait
! Verify R3 = 21 (0x15) at ExtMacro.Trigg
! Verify the other 3 Warnxxx functions are called.
go
wait
go
wait
go
wait
! Run till end of PutWarnings
go
wait
cancel break/all
! Dump first free index in immediate queue (ImFFR), should now be 24 (0x18)
exam/hex 16#ABFD#
! Dump start of immediate queue (should now contain the macro 21 TCMs)
exam/hex 16#A621#..+1c
! Dump OBDH block queues. Should contain 2 ESR warning commands (LASCO, SUMER)
! and one normal (MDI open loop)
exam/hex 16#B14B#..16#B170#
!
! Now do one run of TC_Exec.Process (will execute first TC of the UVCS macro)
! Set break at TC_exec.Process return
set break 16#4823#
! Set break at DecodeDst
set break 16#47EC#
! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
set break 16#4780#
set break 16#479B#
! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and OBDH.Put
set break 16#1E89#
set break 16#1515#
set break 16#131D#
! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and Anomaly.Put
set break 16#25E6#
set break 16#10A8#
! Put IC at start of TC_Exec.Process
mod/hex \ic:=4803
! Run till DecodeDst
go
wait
! Run till DecodeOBDH_Blk
go
wait
! Run till OBDH_BlkQ.PutRaw
go
wait
! Run till Anomaly.Put
go
wait
```

```
! Verify errorcode
! Run till end of TC_Exec.Process
go
wait
!
exit
```

6.6 Test2a.cmd

```
! Module test for UVCSwarning, ESR Warning part
! Test 2a Only UVCS as receiver, UVCS macro with DZDUMMY
!
set log test2a.log
@setup1.cmd
!
! Set break at end of original Exp.Putwarning (should not be reached)
set break 16#56EC#
! Set break at end of patched Exp.PutWarning (should be reached)
set break 16#583B#
! Set break at ExtMacro.Trigg (to verify macro parameter)
set break 16#9F11#
! Set program counter at start of Exp.PutWarning
mod/hex \ic:=56C2
! Dump start of ESR Warning queue data
exam/hex 16#B14B#..+21
! Dump first free index in immediate queue (ImFFR)
exam/hex 16#ABFD#
! Dump start of immediate queue
exam/hex 16#A621#..+18
! Set ESR warning receivers to UVCS only
mod/hex 16#CE5E#..+16#F#:=0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
! Dump ESR warning receivers
exam/hex 16#CE5E#..+16#F#
! Run
go
wait
! Verify R3 = 21 (0x15) at ExtMacro.Trigg
! Run till end of Exp.PutWarning
go
wait
cancel break/all
! Dump start of ESR Warning queue data
exam/hex 16#B14B#..+21
! Dump first free index in immediate queue (ImFFR), should now be 28 (0x1C)
exam/hex 16#ABFD#
! Dump start of immediate queue (should now contain the macro 21 TCMs)
exam/hex 16#A621#..+1c
! Dump OBDH block queues. Should be empty (all zeros)
exam/hex 16#B14B#..16#B170#
!
! Now do one run of TC_Exec.Process (will execute first TC of the UVCS macro)
! Set break at TC_exec.Process return
set break 16#4823#
! Set break at DecodeDst
```

```
set break 16#47EC#
! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
set break 16#4780#
set break 16#479B#
! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and OBDH.Put
set break 16#1E89#
set break 16#1515#
set break 16#131D#
! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and Anomaly.Put
set break 16#25E6#
set break 16#10A8#
! Put IC at start of TC_Exec.Process
mod/hex \ic:=4803
! Run till DecodeDst
go
wait
! Run till DecodeOBDH_Msg
go
wait
! Run till IL_Pkt.Put
go
wait
! Run till RTU.PutRaw
go
wait
! Verify R3-R4 contain dummy interrogation 00E9,0000
! Run till OBDH.Put
go
wait
! Run till end of TC_Exec.Process
go
wait
!
exit
```

6.7 Test2b.cmd

```
! Module test for UVCSwarning, ESR Warning part
! Test 2b All receivers, UVCS macro with DZDUMMY
!
set log test2b.log
@setup1.cmd
!
! Set break at end of original Exp.Putwarning (should not be reached)
set break 16#56EC#
! Set break at end of patched Exp.PutWarning (should be reached)
set break 16#583B#
! Set break at ExtMacro.Trigg (to verify macro parameter)
set break 16#9F11#
! Set program counter at start of Exp.PutWarning
mod/hex \ic:=56C2
! Dump OBDH block queues, start contains ESR Warning queue data
exam/hex 16#B14B#..16#B170#
! Dump first free index in immediate queue (ImFFR)
```

```
exam/hex 16#ABFD#
! Dump start of immediate queue
exam/hex 16#A621#..+18
! Set ESR warning receivers to UVCS only
mod/hex 16#CE5E#..+16#F#:=0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
! Dump ESR warning receivers
exam/hex 16#CE5E#..+16#F#
! Run
go
wait
! Verify R3 = 21 (0x15) at ExtMacro.Trigg
! Run till end of Exp.PutWarning
go
wait
cancel break/all
! Dump OBDH block queues, start contains ESR Warning queue data
exam/hex 16#B14B#..16#B170#
! Dump first free index in immediate queue (ImFFR), should now be 28 (0x1C)
exam/hex 16#ABFD#
! Dump start of immediate queue (should now contain the macro 21 TCMs)
exam/hex 16#A621#..+1c
!
! Now do one run of TC_Exec.Process (will execute first TC of the UVCS macro)
! Set break at TC_exec.Process return
set break 16#4823#
! Set break at DecodeDst
set break 16#47EC#
! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
set break 16#4780#
set break 16#479B#
! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and OBDH.Put
set break 16#1E89#
set break 16#1515#
set break 16#131D#
! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and Anomaly.Put
set break 16#25E6#
set break 16#10A8#
! Put IC at start of TC_Exec.Process
mod/hex \ic:=4803
! Run till DecodeDst
go
wait
! Run till DecodeOBDH_Msg
go
wait
! Run till IL_Pkt.Put
go
wait
! Run till RTU.PutRaw
go
wait
! Verify R3-R4 contain dummy interrogation 00E9,0000
! Run till OBDH.Put
go
wait
```

```
! Run till end of TC_Exec.Process
go
wait
!
exit
```

6.8 Test2c.cmd

```
! Module test for UVCSwarning, ESR Warning part
! Test 2c All receivers, UVCS macro without DZDUMMY
!
set log test2c.log
@setup2.cmd
!
! Set break at end of original Exp.Putwarning (should not be reached)
set break 16#56EC#
! Set break at end of patched Exp.PutWarning (should be reached)
set break 16#583B#
! Set break at ExtMacro.Trigg (to verify macro parameter)
set break 16#9F11#
! Set program counter at start of Exp.PutWarning
mod/hex \ic:=56C2
! Dump OBDH block queues, start contains ESR Warning queue data
exam/hex 16#B14B#..16#B170#
! Dump first free index in immediate queue (ImFFR)
exam/hex 16#ABFD#
! Dump start of immediate queue
exam/hex 16#A621#..+18
! Set ESR warning receivers to UVCS only
mod/hex 16#CE5E#..+16#F#:=0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
! Dump ESR warning receivers
exam/hex 16#CE5E#..+16#F#
! Run
go
wait
! Verify R3 = 21 (0x15) at ExtMacro.Trigg
! Run till end of Exp.PutWarning
go
wait
cancel break/all
! Dump OBDH block queues, start contains ESR Warning queue data
exam/hex 16#B14B#..16#B170#
! Dump first free index in immediate queue (ImFFR), should now be 24 (0x18)
exam/hex 16#ABFD#
! Dump start of immediate queue (should now contain the macro 21 TCMs)
exam/hex 16#A621#..+1c
!
! Now do one run of TC_Exec.Process (will execute first TC of the UVCS macro)
! Set break at TC_exec.Process return
set break 16#4823#
! Set break at DecodeDst
set break 16#47EC#
! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
set break 16#4780#
```

```
set break 16#479B#
! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and OBDH.Put
set break 16#1E89#
set break 16#1515#
set break 16#131D#
! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and Anomaly.Put
set break 16#25E6#
set break 16#10A8#
! Put IC at start of TC_Exec.Process
mod/hex \ic:=4803
! Run till DecodeDst
go
wait
! Run till DecodeOBDH_Blk
go
wait
! Run till OBDH_BlkQ.PutRaw
go
wait
! Run till end of TC_Exec.Process
go
wait
! Dump OBDH block queues, start contains ESR Warning queue data, followed by
first UVCS cmd.
exam/hex 16#B14B#..16#B170#
!
exit
```

6.9 Test1a.log

```
Log file started

TLD Ada Symbolic Debugger  VAX/1750A  V-3.1.H    27-JAN-2010  12:00:58
(c) TLD Systems, Ltd; 1992

1=> ! @setup1.cmd
2=> ! ! Setup1.cmd: UVCS macro with leading DZDUMMY
2=> ! attach tldmps
TLD 1750A Multiple Processor System  v-1.3.3
(c) TLD Systems, Ltd; 1992
2=> ! ! Load COBS and patches
2=> ! load [-.cobsv14]start
Loading program to target
Loading at 16#0051A#
Loading at 16#00000#
Loading at 16#00000#
Loading at 16#073C3#
Loading at 16#07400#
Loading at 16#014AE#
Loading at 16#01811#
Loading at 16#01D5A#
Loading at 16#07906#
Loading at 16#0238B#
Loading at 16#027F6#
```

```
Loading at 16#07D88#
Loading at 16#02D5B#
Loading at 16#081C4#
Loading at 16#03569#
Loading at 16#03A9B#
Loading at 16#03FF3#
Loading at 16#04533#
Loading at 16#049AD#
Loading at 16#04EF2#
Loading at 16#0540D#
Loading at 16#05839#
Loading at 16#08686#
Loading at 16#05E54#
Loading at 16#06362#
Loading at 16#08AB3#
Loading at 16#06BDC#
Loading at 16#070F8#
Loading [-.cobsv14]START.trb
Loading [-.cobsv14]START.dbg
Proc   1  HALTED      Inst Count:*0                  Clock:*0.0
R0:*0000 *0000 *0000 *0000 *0000 *0000 *0000 *0000  IC:*0000 MK:*0000 SW:*0000
R8:*0000 *0000 *0000 *0000 *0000 *0000 *0000 *0000  PI:*0000 SU:*0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! load [-.gyroless]gyroless_2
Loading program to target
Loading at 16#09001#
Loading at 16#09486#
Loading [-.gyroless]GYROLESS_2.trb
Loading [-.gyroless]GYROLESS_2.dbg
Proc   1  HALTED      Inst Count: 0                  Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! load [-.intrec]intrec_3
Loading program to target
Loading at 16#09821#
Loading at 16#09CFB#
Loading [-.intrec]INTREC_3.trb
Loading [-.intrec]INTREC_3.dbg
Proc   1  HALTED      Inst Count: 0                  Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! load [-.tcmmac]tcm_mac_2
Loading program to target
Loading at 16#09E74#
Loading [-.tcmmac]TCM_MAC_2.trb
Loading [-.tcmmac]TCM_MAC_2.dbg
Proc   1  HALTED      Inst Count: 0                  Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! ! Load our patch
2=> ! load uvcswarning.ldm
Loading program to target
```

```
Loading at 16#09644#
Loading UVCSWARNING.trb
Loading UVCSWARNING.dbg
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! ! Reload COBSV14 symbols to avoid warning messages
2=> ! load/symbols [-.cobsv14]start
Loading [-.cobsv14]start.trb
Loading [-.cobsv14]START.dbg
2=> ! exam/ins 16#9643#..16#964D#
16#9643#: PSHM      R3,R3
16#9644#: LIM       R3,16#15#
16#9646#: SJS       R15,IF_MACRO.TRIGG._7
16#9648#: NOP
16#9649#: NOP
16#964A#: NOP
16#964B#: NOP
16#964C#: POPM      R3,R3
16#964D#: URS       R15
16#964E#: PSHM      R3,R6
2=> ! exam/ins 16#56D4#..16#56D5#
16#56D4#: CISP      R12,16#9#
16#56D5#: BGE       EXP.PUTWARNING._61
16#56D6#: AISP      R12,16#1#
2=> ! exam/ins 16#56E5#..16#56E6#
16#56E5#: CISP      R12,16#9#
16#56E6#: BGE       EXP.PUTWARNING._61
16#56E7#: AISP      R12,16#1#
2=> ! exam/ins 16#56E9#..16#56EC#
16#56E9#: JC        16#7#,ESR_GYROSET.PROCESS._44+16#2#
16#56EB#: POPM      R2,R12
16#56EC#: URS       R15
16#56ED#: PSHM      R1,R12
2=> ! exam/ins 16#5830#..16#583B#
16#5830#: LISP      R2,16#A#
16#5831#: L         R11,16#CE5E#,R2
16#5833#: BEZ       ESR_GYROSET.PROCESS._48+16#1#
16#5834#: LIM       R3,16#15#
16#5836#: SJS       R15,IF_MACRO.TRIGG._7
16#5838#: LIM       R15,16#2#,R15
16#583A#: POPM      R2,R12
16#583B#: URS       R15
16#583C#: FAB       R12,16#9#
2=> ! ! Define macro offset table
2=> ! mod/dec 16#795f#..+10#15#:=0,4,12,16,20,24,28,32,36,38,40,44,48,50,60,63
2=> ! mod/dec 16#796f#..+4:=75,78,86,89,99
2=> ! ! Define entry 21 of macro size table
2=> ! mod/dec 16#af2e#+10#20#:=6
2=> ! ! Define macro 21: DZDUMMY, UBMDPRIM, UBFQBAT, UBFQINT, UBAB, UCMDPAN
2=> ! mod/hex
16#acd4#..+10#11#:=0800,D363,0A00,D3A4,0A00,D3A9,0A00,D3AE,0A00,D3B3,0800,D3B8
2=> ! ! Define TCMs for macro 21
2=> ! ! DZDUMMY
```

```
2=> ! mod/hex 16#D363#..+3:=0008,8000,00E9,0000
2=> ! ! UBMDPRIM
2=> ! mod/hex 16#D3A4#..+4:=000A,4680,3782,E300,1A82
2=> ! ! UBFQBAT
2=> ! mod/hex 16#D3A9#..+4:=000A,4680,3782,EA02,2184
2=> ! ! UBFQINT
2=> ! mod/hex 16#D3AE#..+4:=000A,4680,3782,EA01,2183
2=> ! ! UBAB
2=> ! mod/hex 16#D3B3#..+4:=000A,4680,3782,E700,1E82
2=> ! ! UCMDPAN
2=> ! mod/hex 16#D3B8#..+3:=0008,8000,0048,0239
2=> ! !
2=> ! ! Dump TCMs
2=> ! exam/hex 16#D363#..16#D366#
16#D363#: 16#8#
16#D364#: 16#8000#
16#D365#: 16#E9#
16#D366#: 16#0#
2=> ! exam/hex 16#D3A4#..16#D3BB#
16#D3A4#: 16#A#
16#D3A5#: 16#4680#
16#D3A6#: 16#3782#
16#D3A7#: 16#E300#
16#D3A8#: 16#1A82#
16#D3A9#: 16#A#
16#D3AA#: 16#4680#
16#D3AB#: 16#3782#
16#D3AC#: 16#EA02#
16#D3AD#: 16#2184#
16#D3AE#: 16#A#
16#D3AF#: 16#4680#
16#D3B0#: 16#3782#
16#D3B1#: 16#EA01#
16#D3B2#: 16#2183#
16#D3B3#: 16#A#
16#D3B4#: 16#4680#
16#D3B5#: 16#3782#
16#D3B6#: 16#E700#
16#D3B7#: 16#1E82#
16#D3B8#: 16#8#
16#D3B9#: 16#8000#
16#D3BA#: 16#48#
16#D3BB#: 16#239#
2=> ! !
2=> ! ! Init stack pointer
2=> ! mod/hex \rf:=ffff
Proc   1  HALTED      Inst Count: 0                  Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
2=> ! !
2=> ! ! Elaborate OBDH_Blkq (Initialize OBDH block command queues)
2=> ! set break 16#2793#
Breakpoint    1  at  16#2793#
    Action:  HALT execution
2=> ! mod/hex \ic:=2730
```

```
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*2730 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
2=> ! wait
Proc   1  HALTED      Inst Count:*597               Clock:*0.000_296_1
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*2793 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at OBDH_BLKQ.INIT._364+16#3#: URS       R15
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate Exp (Initialize ESR warning command constants and set default
ESR warning receivers)
2=> ! set break 16#577c#
Breakpoint    1  at  16#577C#
    Action:  HALT execution
2=> ! mod/hex \ic:=575c
Proc   1  HALTED      Inst Count: 597               Clock: 0.000_296_1
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*575C MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
Proc   1  HALTED      Inst Count:*616               Clock:*0.000_315_9
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*577C MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at EXP.INIT._99+16#3#: URS       R15
2=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate TC_Exec
2=> ! set break 16#48DA#
Breakpoint    1  at  16#48DA#
    Action:  HALT execution
2=> ! mod/hex \ic:=48B7
Proc   1  HALTED      Inst Count: 616               Clock: 0.000_315_9
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*48B7 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
2=> ! wait
Proc   1  HALTED      Inst Count:*1742              Clock:*0.000_794_8
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*48DA MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.INIT._1515+16#1#: URS       R15
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate IL_Pkt
2=> ! set break 16#1EF8#
Breakpoint    1  at  16#1EF8#
    Action:  HALT execution
2=> ! mod/hex \ic:=1EF6
Proc   1  HALTED      Inst Count: 1742              Clock: 0.000_794_8
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*1EF6 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
Proc   1  HALTED      Inst Count:*1743              Clock:*0.000_795_2
```

```
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*1EF8 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at IL_PKT.INIT._35: URS       R15
2=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate RTU
2=> ! set break 16#162E#
Breakpoint    1  at  16#162E#
    Action:  HALT execution
2=> ! mod \ic:=16#161F#
Proc   1  HALTED      Inst Count: 1743              Clock: 0.000_795_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*161F MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
Proc   1  HALTED      Inst Count:*1769              Clock:*0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*162E MK: 0000 SW:*4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at RTU.INIT._197+16#1#: URS       R15
2=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
2=> ! cancel break/all
2=> ! !
2=> ! ! Put an immediate return in OBDH.Put
2=> ! mod/hex 16#131D#:=7FF0
2=> ! !
2=> ! exit
1=> ! !
1=> ! ! Set break at end of PutWarnings
1=> ! set break 16#96CD#
Breakpoint    1  at  16#96CD#
    Action:  HALT execution
1=> ! ! Set break at WarnUVCS, WarnLASCO, WarnMDI and WarnSUMER
1=> ! set break 16#9643#
Breakpoint    2  at  16#9643#
    Action:  HALT execution
1=> ! set break 16#964E#
Breakpoint    3  at  16#964E#
    Action:  HALT execution
1=> ! set break 16#9658#
Breakpoint    4  at  16#9658#
    Action:  HALT execution
1=> ! set break 16#9663#
Breakpoint    5  at  16#9663#
    Action:  HALT execution
1=> ! ! Set break at PutPower (should not be reached)
1=> ! set break 16#90C8#
Breakpoint    6  at  16#90C8#
    Action:  HALT execution
1=> ! ! Set break at ExtMacro.Trigg (to verify macro parameter)
1=> ! set break 16#9F11#
Breakpoint    7  at  16#9F11#
    Action:  HALT execution
1=> ! ! Set program counter at start of PutWarnings
```

```
1=> ! mod/hex \ic:=9672
Proc   1  HALTED      Inst Count: 1769              Clock: 0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*9672 MK: 0000 SW: 4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Dump first free index in immediate queue (ImFFR)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#0#
1=> ! ! Dump start of immediate queue
1=> ! exam/hex 16#A621#..+18
16#A621#: 16#0#
16#A622#: 16#0#
16#A623#: 16#0#
16#A624#: 16#0#
16#A625#: 16#0#
16#A626#: 16#0#
16#A627#: 16#0#
16#A628#: 16#0#
16#A629#: 16#0#
16#A62A#: 16#0#
16#A62B#: 16#0#
16#A62C#: 16#0#
16#A62D#: 16#0#
16#A62E#: 16#0#
16#A62F#: 16#0#
16#A630#: 16#0#
16#A631#: 16#0#
16#A632#: 16#0#
16#A633#: 16#0#
16#A634#: 16#0#
16#A635#: 16#0#
16#A636#: 16#0#
16#A637#: 16#0#
16#A638#: 16#0#
16#A639#: 16#0#
1=> ! ! Set receivers to UVCS only
1=> ! mod/hex 16#FC12#:=0020
1=> ! ! Run
1=> ! go
Proc   1  HALTED      Inst Count:*1812              Clock:*0.000_827_1
R0: 0000  0000 *FFFF *000A *0007  0000  0000  0000  IC:*9643 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNUVCS/$ISECT$: PSHM      R3,R3
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify break at WarnUVCS
1=> ! go
Proc   1  HALTED      Inst Count:*1816              Clock:*0.000_828_5
R0: 0000  0000  FFFF *0015  0007  0000  0000  0000  IC:*9F11 MK: 0000 SW:*4000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFEE  PI: 0000 SU: 0   TSK:
BP taken at EXTMACRO.TRIGG/$ISECT$: PSHM      R0,R12
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3 = 21 (0x15) at ExtMacro.Trigg
1=> ! ! Run till end of PutWarnings
1=> ! go
```

```
1=> ! wait
Proc   1  HALTED      Inst Count:*2860              Clock:*0.001_254_3
R0: 0000  0000 *0000 *0000 *0000  0000  0000  0000  IC:*96CD MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.PUTWARNINGS/$ISECT$+16#5B#: URS       R15
1=> ! cancel break/all
1=> ! ! Dump first free index in immediate queue (ImFFR), should now be 28
(0x1C)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#1C#
1=> ! ! Dump start of immediate queue (should now contain the macro 21 TCMs)
1=> ! exam/hex 16#A621#..+1c
16#A621#: 16#8#
16#A622#: 16#8000#
16#A623#: 16#E9#
16#A624#: 16#0#
16#A625#: 16#A#
16#A626#: 16#4680#
16#A627#: 16#3782#
16#A628#: 16#E300#
16#A629#: 16#1A82#
16#A62A#: 16#A#
16#A62B#: 16#4680#
16#A62C#: 16#3782#
16#A62D#: 16#EA02#
16#A62E#: 16#2184#
16#A62F#: 16#A#
16#A630#: 16#4680#
16#A631#: 16#3782#
16#A632#: 16#EA01#
16#A633#: 16#2183#
16#A634#: 16#A#
16#A635#: 16#4680#
16#A636#: 16#3782#
16#A637#: 16#E700#
16#A638#: 16#1E82#
16#A639#: 16#8#
16#A63A#: 16#8000#
16#A63B#: 16#48#
16#A63C#: 16#239#
16#A63D#: 16#0#
1=> ! ! Dump OBDH block queues. Should be empty (all zeros)
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#0#
16#B14C#: 16#0#
16#B14D#: 16#0#
16#B14E#: 16#0#
16#B14F#: 16#0#
16#B150#: 16#0#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
```

```
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
16#B16D#: 16#0#
16#B16E#: 16#0#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! !
1=> ! ! Now do one run of TC_Exec.Process (will execute first TC of the UVCS
macro)
1=> ! ! Set break at TC_exec.Process return
1=> ! set break 16#4823#
Breakpoint    1  at  16#4823#
    Action:  HALT execution
1=> ! ! Set break at DecodeDst
1=> ! set break 16#47EC#
Breakpoint    2  at  16#47EC#
    Action:  HALT execution
1=> ! ! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
1=> ! set break 16#4780#
Breakpoint    3  at  16#4780#
    Action:  HALT execution
1=> ! set break 16#479B#
Breakpoint    4  at  16#479B#
    Action:  HALT execution
1=> ! ! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and
OBDH.Put
1=> ! set break 16#1E89#
Breakpoint    5  at  16#1E89#
    Action:  HALT execution
1=> ! set break 16#1515#
Breakpoint    6  at  16#1515#
    Action:  HALT execution
1=> ! set break 16#131D#
Breakpoint    7  at  16#131D#
    Action:  HALT execution
```

```
1=> ! ! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and
Anomaly.Put
1=> ! set break 16#25E6#
Breakpoint    8  at  16#25E6#
    Action:  HALT execution
1=> ! set break 16#10A8#
Breakpoint    9  at  16#10A8#
    Action:  HALT execution
1=> ! ! Put IC at start of TC_Exec.Process
1=> ! mod/hex \ic:=4803
Proc   1  HALTED      Inst Count: 2860              Clock: 0.001_254_3
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*4803 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Run till DecodeDst
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*2982              Clock:*0.001_316_5
R0: 0000  0000 *0001 *CCBC *CCBD *CCBE *CD3C  0000  IC:*47EC MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF9  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEDST._1417: PSHM      R2,R2
1=> ! ! Run till DecodeOBDH_Msg
1=> ! go
Proc   1  HALTED      Inst Count:*2991              Clock:*0.001_320_3
R0: 0000  0000 *0002  CCBC  CCBD  CCBE  CD3C  0000  IC:*4780 MK: 0000 SW:*2000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF7  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEOBDH_MSG._1379: PSHM      R2,R5
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till IL_Pkt.Put
1=> ! go
Proc   1  HALTED      Inst Count:*3018              Clock:*0.001_329_6
R0: 0000  0000 *0001 *CCBE *0001 *0001  CD3C  0000  IC:*1E89 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at IL_PKT.PUT._8: PSHM      R0,R14
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till RTU.PutRaw
1=> ! go
Proc   1  HALTED      Inst Count:*3031              Clock:*0.001_336_2
R0: 0000  0000 *CCBE *00E9 *0000  0001  CD3C  0000  IC:*1515 MK: 0000 SW:*4000
R8: 0000  0000  0000 *CCC0 *0001 *CCBE  0000 *FFDE  PI: 0000 SU: 0   TSK:
BP taken at RTU.PUTRAW._119: PSHM      R1,R6
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3-R4 contain dummy interrogation 00E9,0000
1=> ! ! Run till OBDH.Put
1=> ! go
Proc   1  HALTED      Inst Count:*3037              Clock:*0.001_339_2
R0: 0000  0000 *0001  00E9  0000  0001  CD3C  0000  IC:*131D MK: 0000 SW: 4000
R8: 0000  0000  0000  CCC0  0001  CCBE  0000 *FFD6  PI: 0000 SU: 0   TSK:
BP taken at OBDH.PUT._59: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till end of TC_Exec.Process
1=> ! go
```

```
Proc   1  HALTED      Inst Count:*3191              Clock:*0.001_412_4
R0: 0000  0000 *0000 *0000  0000 *0000 *0000  0000  IC:*4823 MK: 0000 SW:*2000
R8: 0000  0000  0000 *0000 *0000 *0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.PROCESS._1441+16#1#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! !
1=> ! exit
=>
=> exit
Exiting Debugger.
```

6.10 Test1b.log

```
Log file started

TLD Ada Symbolic Debugger  VAX/1750A  V-3.1.H    27-JAN-2010  12:09:17
(c) TLD Systems, Ltd; 1992

1=> ! @setup1.cmd
2=> ! ! Setup1.cmd: UVCS macro with leading DZDUMMY
2=> ! attach tldmps
TLD 1750A Multiple Processor System  v-1.3.3
(c) TLD Systems, Ltd; 1992
2=> ! ! Load COBS and patches
2=> ! load [-.cobsv14]start
```

…

See test1a.log

…

```
2=> ! ! Put an immediate return in OBDH.Put
2=> ! mod/hex 16#131D#:=7FF0
2=> ! !
2=> ! exit
1=> ! !
1=> ! ! Set break at end of PutWarnings
1=> ! set break 16#96CD#
Breakpoint    1  at  16#96CD#
    Action:  HALT execution
1=> ! ! Set break at WarnUVCS, WarnLASCO, WarnMDI and WarnSUMER
1=> ! set break 16#9643#
Breakpoint    2  at  16#9643#
    Action:  HALT execution
1=> ! set break 16#964E#
Breakpoint    3  at  16#964E#
    Action:  HALT execution
1=> ! set break 16#9658#
Breakpoint    4  at  16#9658#
    Action:  HALT execution
1=> ! set break 16#9663#
Breakpoint    5  at  16#9663#
    Action:  HALT execution
1=> ! ! Set break at PutPower (should not be reached)
1=> ! set break 16#90C8#
Breakpoint    6  at  16#90C8#
```

```
    Action:  HALT execution
1=> ! ! Set break at ExtMacro.Trigg (to verify macro parameter)
1=> ! set break 16#9F11#
Breakpoint    7  at  16#9F11#
    Action:  HALT execution
1=> ! ! Set program counter at start of PutWarnings
1=> ! mod/hex \ic:=9672
Proc   1  HALTED      Inst Count: 1769              Clock: 0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*9672 MK: 0000 SW: 4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Dump first free index in immediate queue (ImFFR)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#0#
1=> ! ! Dump start of immediate queue
1=> ! exam/hex 16#A621#..+18
16#A621#: 16#0#
16#A622#: 16#0#
16#A623#: 16#0#
16#A624#: 16#0#
16#A625#: 16#0#
16#A626#: 16#0#
16#A627#: 16#0#
16#A628#: 16#0#
16#A629#: 16#0#
16#A62A#: 16#0#
16#A62B#: 16#0#
16#A62C#: 16#0#
16#A62D#: 16#0#
16#A62E#: 16#0#
16#A62F#: 16#0#
16#A630#: 16#0#
16#A631#: 16#0#
16#A632#: 16#0#
16#A633#: 16#0#
16#A634#: 16#0#
16#A635#: 16#0#
16#A636#: 16#0#
16#A637#: 16#0#
16#A638#: 16#0#
16#A639#: 16#0#
1=> ! ! Set receivers to UVCS, LASCO, MDI, SUMER
1=> ! mod/hex 16#FC12#:=03A0
1=> ! ! Run
1=> ! go
Proc   1  HALTED      Inst Count:*1812              Clock:*0.000_827_1
R0: 0000  0000 *FFFF *000A *0007  0000  0000  0000  IC:*9643 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNUVCS/$ISECT$: PSHM      R3,R3
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify break at WarnUVCS
1=> ! go
Proc   1  HALTED      Inst Count:*1816              Clock:*0.000_828_5
R0: 0000  0000  FFFF *0015  0007  0000  0000  0000  IC:*9F11 MK: 0000 SW:*4000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFEE  PI: 0000 SU: 0   TSK:
```

```
BP taken at EXTMACRO.TRIGG/$ISECT$: PSHM      R0,R12
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3 = 21 (0x15) at ExtMacro.Trigg
1=> ! ! Verify the other 3 Warnxxx functions are called.
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*2618              Clock:*0.001_153_4
R0: 0000  0000  FFFF *0006  0007  0000  0000  0000  IC:*964E MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNLASCO/$ISECT$: PSHM      R3,R6
1=> ! go
Proc   1  HALTED      Inst Count:*2903              Clock:*0.001_281_9
R0: 0000  0000  FFFF *0007  0007  0000  0000  0000  IC:*9658 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNMDI/$ISECT$: PSHM      R3,R6
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*3551              Clock:*0.001_520_4
R0: 0000  0000  FFFF *0008  0007  0000  0000  0000  IC:*9663 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNSUMER/$ISECT$: PSHM      R3,R6
1=> ! ! Run till end of PutWarnings
1=> ! go
Proc   1  HALTED      Inst Count:*4073              Clock:*0.001_741_5
R0: 0000  0000 *0000 *0000 *0000  0000  0000  0000  IC:*96CD MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.PUTWARNINGS/$ISECT$+16#5B#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! cancel break/all
1=> ! ! Dump first free index in immediate queue (ImFFR), should now be 28
(0x1C)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#1C#
1=> ! ! Dump start of immediate queue (should now contain the macro 21 TCMs)
1=> ! exam/hex 16#A621#..+1c
16#A621#: 16#8#
16#A622#: 16#8000#
16#A623#: 16#E9#
16#A624#: 16#0#
16#A625#: 16#A#
16#A626#: 16#4680#
16#A627#: 16#3782#
16#A628#: 16#E300#
16#A629#: 16#1A82#
16#A62A#: 16#A#
16#A62B#: 16#4680#
16#A62C#: 16#3782#
16#A62D#: 16#EA02#
16#A62E#: 16#2184#
16#A62F#: 16#A#
16#A630#: 16#4680#
```

```
16#A631#: 16#3782#
16#A632#: 16#EA01#
16#A633#: 16#2183#
16#A634#: 16#A#
16#A635#: 16#4680#
16#A636#: 16#3782#
16#A637#: 16#E700#
16#A638#: 16#1E82#
16#A639#: 16#8#
16#A63A#: 16#8000#
16#A63B#: 16#48#
16#A63C#: 16#239#
16#A63D#: 16#0#
1=> ! ! Dump OBDH block queues. Should contain 2 ESR warning commands (LASCO,
SUMER)
1=> ! ! and one normal (MDI open loop)
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#24A2#
16#B14C#: 16#CCCC#
16#B14D#: 16#F16E#
16#B14E#: 16#2CA2#
16#B14F#: 16#CCCC#
16#B150#: 16#F96E#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#2B03#
16#B16D#: 16#7#
16#B16E#: 16#1#
16#B16F#: 16#2B0B#
16#B170#: 16#0#
```

```
1=> ! !
1=> ! ! Now do one run of TC_Exec.Process (will execute first TC of the UVCS
macro)
1=> ! ! Set break at TC_exec.Process return
1=> ! set break 16#4823#
Breakpoint    1  at  16#4823#
    Action:  HALT execution
1=> ! ! Set break at DecodeDst
1=> ! set break 16#47EC#
Breakpoint    2  at  16#47EC#
    Action:  HALT execution
1=> ! ! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
1=> ! set break 16#4780#
Breakpoint    3  at  16#4780#
    Action:  HALT execution
1=> ! set break 16#479B#
Breakpoint    4  at  16#479B#
    Action:  HALT execution
1=> ! ! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and
OBDH.Put
1=> ! set break 16#1E89#
Breakpoint    5  at  16#1E89#
    Action:  HALT execution
1=> ! set break 16#1515#
Breakpoint    6  at  16#1515#
    Action:  HALT execution
1=> ! set break 16#131D#
Breakpoint    7  at  16#131D#
    Action:  HALT execution
1=> ! ! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and
Anomaly.Put
1=> ! set break 16#25E6#
Breakpoint    8  at  16#25E6#
    Action:  HALT execution
1=> ! set break 16#10A8#
Breakpoint    9  at  16#10A8#
    Action:  HALT execution
1=> ! ! Put IC at start of TC_Exec.Process
1=> ! mod/hex \ic:=4803
Proc   1  HALTED      Inst Count: 4073              Clock: 0.001_741_5
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*4803 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Run till DecodeDst
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*4195              Clock:*0.001_803_7
R0: 0000  0000 *0001 *CCBC *CCBD *CCBE *CD3C  0000  IC:*47EC MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF9  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEDST._1417: PSHM      R2,R2
1=> ! ! Run till DecodeOBDH_Msg
1=> ! go
Proc   1  HALTED      Inst Count:*4204              Clock:*0.001_807_5
R0: 0000  0000 *0002  CCBC  CCBD  CCBE  CD3C  0000  IC:*4780 MK: 0000 SW:*2000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF7  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEOBDH_MSG._1379: PSHM      R2,R5
```

```
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till IL_Pkt.Put
1=> ! go
Proc   1  HALTED      Inst Count:*4231              Clock:*0.001_816_8
R0: 0000  0000 *0001 *CCBE *0001 *0001  CD3C  0000  IC:*1E89 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at IL_PKT.PUT._8: PSHM      R0,R14
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till RTU.PutRaw
1=> ! go
Proc   1  HALTED      Inst Count:*4244              Clock:*0.001_823_4
R0: 0000  0000 *CCBE *00E9 *0000  0001  CD3C  0000  IC:*1515 MK: 0000 SW:*4000
R8: 0000  0000  0000 *CCC0 *0001 *CCBE  0000 *FFDE  PI: 0000 SU: 0   TSK:
BP taken at RTU.PUTRAW._119: PSHM      R1,R6
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3-R4 contain dummy interrogation 00E9,0000
1=> ! ! Run till OBDH.Put
1=> ! go
Proc   1  HALTED      Inst Count:*4250              Clock:*0.001_826_4
R0: 0000  0000 *0001  00E9  0000  0001  CD3C  0000  IC:*131D MK: 0000 SW: 4000
R8: 0000  0000  0000  CCC0  0001  CCBE  0000 *FFD6  PI: 0000 SU: 0   TSK:
BP taken at OBDH.PUT._59: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till end of TC_Exec.Process
1=> ! go
Proc   1  HALTED      Inst Count:*4404              Clock:*0.001_899_6
R0: 0000  0000 *0000 *0000  0000 *0000 *0000  0000  IC:*4823 MK: 0000 SW:*2000
R8: 0000  0000  0000 *0000 *0000 *0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.PROCESS._1441+16#1#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! !
1=> ! exit
=>
=> exit
Exiting Debugger.
```

6.11 Test1c.log

```
Log file started

TLD Ada Symbolic Debugger  VAX/1750A  V-3.1.H    27-JAN-2010  12:15:20
(c) TLD Systems, Ltd; 1992

1=> ! @setup2.cmd
2=> ! ! Setup2.cmd: UVCS macro without leading DZDUMMY
2=> ! attach tldmps
TLD 1750A Multiple Processor System  v-1.3.3
(c) TLD Systems, Ltd; 1992
2=> ! ! Load COBS and patches
```

```
2=> ! load [-.cobsv14]start
Loading program to target
Loading at 16#0051A#
Loading at 16#00000#
Loading at 16#00000#
Loading at 16#073C3#
Loading at 16#07400#
Loading at 16#014AE#
Loading at 16#01811#
Loading at 16#01D5A#
Loading at 16#07906#
Loading at 16#0238B#
Loading at 16#027F6#
Loading at 16#07D88#
Loading at 16#02D5B#
Loading at 16#081C4#
Loading at 16#03569#
Loading at 16#03A9B#
Loading at 16#03FF3#
Loading at 16#04533#
Loading at 16#049AD#
Loading at 16#04EF2#
Loading at 16#0540D#
Loading at 16#05839#
Loading at 16#08686#
Loading at 16#05E54#
Loading at 16#06362#
Loading at 16#08AB3#
Loading at 16#06BDC#
Loading at 16#070F8#
Loading [-.cobsv14]START.trb
Loading [-.cobsv14]START.dbg
Proc   1  HALTED      Inst Count:*0                 Clock:*0.0
R0:*0000 *0000 *0000 *0000 *0000 *0000 *0000 *0000  IC:*0000 MK:*0000 SW:*0000
R8:*0000 *0000 *0000 *0000 *0000 *0000 *0000 *0000  PI:*0000 SU:*0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! load [-.gyroless]gyroless_2
Loading program to target
Loading at 16#09001#
Loading at 16#09486#
Loading [-.gyroless]GYROLESS_2.trb
Loading [-.gyroless]GYROLESS_2.dbg
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! load [-.intrec]intrec_3
Loading program to target
Loading at 16#09821#
Loading at 16#09CFB#
Loading [-.intrec]INTREC_3.trb
Loading [-.intrec]INTREC_3.dbg
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
```

```
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! load [-.tcmmac]tcm_mac_2
Loading program to target
Loading at 16#09E74#
Loading [-.tcmmac]TCM_MAC_2.trb
Loading [-.tcmmac]TCM_MAC_2.dbg
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! ! Load our patch
2=> ! load uvcswarning.ldm
Loading program to target
Loading at 16#09644#
Loading UVCSWARNING.trb
Loading UVCSWARNING.dbg
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  0000  PI: 0000 SU: 0   TSK:
Start at 16#0.0000#: JC        16#7#,16#30B#
2=> ! ! Reload COBSV14 symbols to avoid warning messages
2=> ! load/symbols [-.cobsv14]start
Loading [-.cobsv14]start.trb
Loading [-.cobsv14]START.dbg
2=> ! ! Verify our patch
2=> ! exam/ins 16#9643#..16#964D#
16#9643#: PSHM      R3,R3
16#9644#: LIM       R3,16#15#
16#9646#: SJS       R15,IF_MACRO.TRIGG._7
16#9648#: NOP
16#9649#: NOP
16#964A#: NOP
16#964B#: NOP
16#964C#: POPM      R3,R3
16#964D#: URS       R15
16#964E#: PSHM      R3,R6
2=> ! exam/ins 16#56D4#..16#56D5#
16#56D4#: CISP      R12,16#9#
16#56D5#: BGE       EXP.PUTWARNING._61
16#56D6#: AISP      R12,16#1#
2=> ! exam/ins 16#56E5#..16#56E6#
16#56E5#: CISP      R12,16#9#
16#56E6#: BGE       EXP.PUTWARNING._61
16#56E7#: AISP      R12,16#1#
2=> ! exam/ins 16#56E9#..16#56EC#
16#56E9#: JC        16#7#,ESR_GYROSET.PROCESS._44+16#2#
16#56EB#: POPM      R2,R12
16#56EC#: URS       R15
16#56ED#: PSHM      R1,R12
2=> ! exam/ins 16#5830#..16#583B#
16#5830#: LISP      R2,16#A#
16#5831#: L         R11,16#CE5E#,R2
16#5833#: BEZ       ESR_GYROSET.PROCESS._48+16#1#
16#5834#: LIM       R3,16#15#
16#5836#: SJS       R15,IF_MACRO.TRIGG._7
```

```
16#5838#: LIM       R15,16#2#,R15
16#583A#: POPM      R2,R12
16#583B#: URS       R15
16#583C#: FAB       R12,16#9#
2=> ! ! Define macro offset table
2=> ! mod/dec 16#795f#..+10#15#:=0,4,12,16,20,24,28,32,36,38,40,44,48,50,60,63
2=> ! mod/dec 16#796f#..+4:=75,78,86,89,99
2=> ! ! Define entry 21 of macro size table
2=> ! mod/dec 16#af2e#+10#20#:=5
2=> ! ! Define macro 21: UBMDPRIM, UBFQBAT, UBFQINT, UBAB, UCMDPAN
2=> ! mod/hex 16#acd4#..+9:=0A00,D3A4,0A00,D3A9,0A00,D3AE,0A00,D3B3,0800,D3B8
2=> ! ! Define TCMs for macro 21
2=> ! ! UBMDPRIM
2=> ! mod/hex 16#D3A4#..+4:=000A,4680,3782,E300,1A82
2=> ! ! UBFQBAT
2=> ! mod/hex 16#D3A9#..+4:=000A,4680,3782,EA02,2184
2=> ! ! UBFQINT
2=> ! mod/hex 16#D3AE#..+4:=000A,4680,3782,EA01,2183
2=> ! ! UBAB
2=> ! mod/hex 16#D3B3#..+4:=000A,4680,3782,E700,1E82
2=> ! ! UCMDPAN
2=> ! mod/hex 16#D3B8#..+3:=0008,8000,0048,0239
2=> ! !
2=> ! ! Dump TCMs
2=> ! exam/hex 16#D3A4#..16#D3BB#
16#D3A4#: 16#A#
16#D3A5#: 16#4680#
16#D3A6#: 16#3782#
16#D3A7#: 16#E300#
16#D3A8#: 16#1A82#
16#D3A9#: 16#A#
16#D3AA#: 16#4680#
16#D3AB#: 16#3782#
16#D3AC#: 16#EA02#
16#D3AD#: 16#2184#
16#D3AE#: 16#A#
16#D3AF#: 16#4680#
16#D3B0#: 16#3782#
16#D3B1#: 16#EA01#
16#D3B2#: 16#2183#
16#D3B3#: 16#A#
16#D3B4#: 16#4680#
16#D3B5#: 16#3782#
16#D3B6#: 16#E700#
16#D3B7#: 16#1E82#
16#D3B8#: 16#8#
16#D3B9#: 16#8000#
16#D3BA#: 16#48#
16#D3BB#: 16#239#
2=> ! !
2=> ! ! Init stack pointer
2=> ! mod/hex \rf:=ffff
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC: 0000 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
```

```
2=> ! !
2=> ! ! Elaborate OBDH_Blkq (Initialize OBDH block command queues)
2=> ! set break 16#2793#
Breakpoint    1  at  16#2793#
    Action:  HALT execution
2=> ! mod/hex \ic:=2730
Proc   1  HALTED      Inst Count: 0                 Clock: 0.0
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*2730 MK: 0000 SW: 0000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
2=> ! wait
Proc   1  HALTED      Inst Count:*597               Clock:*0.000_296_1
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*2793 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at OBDH_BLKQ.INIT._364+16#3#: URS       R15
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate Exp (Initialize ESR warning command constants and set default
ESR warning receivers)
2=> ! set break 16#577c#
Breakpoint    1  at  16#577C#
    Action:  HALT execution
2=> ! mod/hex \ic:=575c
Proc   1  HALTED      Inst Count: 597               Clock: 0.000_296_1
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*575C MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
Proc   1  HALTED      Inst Count:*616               Clock:*0.000_315_9
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*577C MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at EXP.INIT._99+16#3#: URS       R15
2=> ! wait 1
%TLDDBG-E-PROCNORUN - Must run processor first
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate TC_Exec
2=> ! set break 16#48DA#
Breakpoint    1  at  16#48DA#
    Action:  HALT execution
2=> ! mod/hex \ic:=48B7
Proc   1  HALTED      Inst Count: 616               Clock: 0.000_315_9
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*48B7 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
2=> ! wait
Proc   1  HALTED      Inst Count:*1742              Clock:*0.000_794_8
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*48DA MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.INIT._1515+16#1#: URS       R15
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate IL_Pkt
2=> ! set break 16#1EF8#
Breakpoint    1  at  16#1EF8#
    Action:  HALT execution
```

```
2=> ! mod/hex \ic:=1EF6
Proc   1  HALTED      Inst Count: 1742              Clock: 0.000_794_8
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*1EF6 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
Proc   1  HALTED      Inst Count:*1743              Clock:*0.000_795_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*1EF8 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at IL_PKT.INIT._35: URS       R15
2=> ! wait 1
%TLDDBG-E-PROCNORUN - Must run processor first
2=> ! cancel break/all
2=> ! !
2=> ! ! Elaborate RTU
2=> ! set break 16#162E#
Breakpoint    1  at  16#162E#
    Action:  HALT execution
2=> ! mod \ic:=16#161F#
Proc   1  HALTED      Inst Count: 1743              Clock: 0.000_795_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*161F MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
2=> ! go
Proc   1  HALTED      Inst Count:*1769              Clock:*0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*162E MK: 0000 SW:*4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
BP taken at RTU.INIT._197+16#1#: URS       R15
2=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
2=> ! cancel break/all
2=> ! !
2=> ! ! Put an immediate return in OBDH.Put
2=> ! mod/hex 16#131D#:=7FF0
2=> ! !
2=> ! exit
1=> ! !
1=> ! ! Set break at end of PutWarnings
1=> ! set break 16#96CD#
Breakpoint    1  at  16#96CD#
    Action:  HALT execution
1=> ! ! Set break at WarnUVCS, WarnLASCO, WarnMDI and WarnSUMER
1=> ! set break 16#9643#
Breakpoint    2  at  16#9643#
    Action:  HALT execution
1=> ! set break 16#964E#
Breakpoint    3  at  16#964E#
    Action:  HALT execution
1=> ! set break 16#9658#
Breakpoint    4  at  16#9658#
    Action:  HALT execution
1=> ! set break 16#9663#
Breakpoint    5  at  16#9663#
    Action:  HALT execution
1=> ! ! Set break at PutPower (should not be reached)
1=> ! set break 16#90C8#
Breakpoint    6  at  16#90C8#
```

```
    Action:  HALT execution
1=> ! ! Set break at ExtMacro.Trigg (to verify macro parameter)
1=> ! set break 16#9F11#
Breakpoint    7  at  16#9F11#
    Action:  HALT execution
1=> ! ! Set program counter at start of PutWarnings
1=> ! mod/hex \ic:=9672
Proc   1  HALTED      Inst Count: 1769              Clock: 0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*9672 MK: 0000 SW: 4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Dump first free index in immediate queue (ImFFR)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#0#
1=> ! ! Dump start of immediate queue
1=> ! exam/hex 16#A621#..+18
16#A621#: 16#0#
16#A622#: 16#0#
16#A623#: 16#0#
16#A624#: 16#0#
16#A625#: 16#0#
16#A626#: 16#0#
16#A627#: 16#0#
16#A628#: 16#0#
16#A629#: 16#0#
16#A62A#: 16#0#
16#A62B#: 16#0#
16#A62C#: 16#0#
16#A62D#: 16#0#
16#A62E#: 16#0#
16#A62F#: 16#0#
16#A630#: 16#0#
16#A631#: 16#0#
16#A632#: 16#0#
16#A633#: 16#0#
16#A634#: 16#0#
16#A635#: 16#0#
16#A636#: 16#0#
16#A637#: 16#0#
16#A638#: 16#0#
16#A639#: 16#0#
1=> ! ! Set receivers to UVCS, LASCO, MDI, SUMER
1=> ! mod/hex 16#FC12#:=03A0
1=> ! ! Run
1=> ! go
Proc   1  HALTED      Inst Count:*1812              Clock:*0.000_827_1
R0: 0000  0000 *FFFF *000A *0007  0000  0000  0000  IC:*9643 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNUVCS/$ISECT$: PSHM      R3,R3
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify break at WarnUVCS
1=> ! go
Proc   1  HALTED      Inst Count:*1816              Clock:*0.000_828_5
R0: 0000  0000  FFFF *0015  0007  0000  0000  0000  IC:*9F11 MK: 0000 SW:*4000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFEE  PI: 0000 SU: 0   TSK:
```

```
BP taken at EXTMACRO.TRIGG/$ISECT$: PSHM      R0,R12
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3 = 21 (0x15) at ExtMacro.Trigg
1=> ! ! Verify the other 3 Warnxxx functions are called.
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*2502              Clock:*0.001_107_7
R0: 0000  0000  FFFF *0006  0007  0000  0000  0000  IC:*964E MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNLASCO/$ISECT$: PSHM      R3,R6
1=> ! go
Proc   1  HALTED      Inst Count:*2787              Clock:*0.001_236_2
R0: 0000  0000  FFFF *0007  0007  0000  0000  0000  IC:*9658 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNMDI/$ISECT$: PSHM      R3,R6
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*3435              Clock:*0.001_474_7
R0: 0000  0000  FFFF *0008  0007  0000  0000  0000  IC:*9663 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFF0  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.WARNSUMER/$ISECT$: PSHM      R3,R6
1=> ! ! Run till end of PutWarnings
1=> ! go
Proc   1  HALTED      Inst Count:*3957              Clock:*0.001_695_8
R0: 0000  0000 *0000 *0000 *0000  0000  0000  0000  IC:*96CD MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at EXPOFFPOINTING.PUTWARNINGS/$ISECT$+16#5B#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! cancel break/all
1=> ! ! Dump first free index in immediate queue (ImFFR), should now be 24
(0x18)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#18#
1=> ! ! Dump start of immediate queue (should now contain the macro 21 TCMs)
1=> ! exam/hex 16#A621#..+1c
16#A621#: 16#A#
16#A622#: 16#4680#
16#A623#: 16#3782#
16#A624#: 16#E300#
16#A625#: 16#1A82#
16#A626#: 16#A#
16#A627#: 16#4680#
16#A628#: 16#3782#
16#A629#: 16#EA02#
16#A62A#: 16#2184#
16#A62B#: 16#A#
16#A62C#: 16#4680#
16#A62D#: 16#3782#
16#A62E#: 16#EA01#
16#A62F#: 16#2183#
16#A630#: 16#A#
```

```
16#A631#: 16#4680#
16#A632#: 16#3782#
16#A633#: 16#E700#
16#A634#: 16#1E82#
16#A635#: 16#8#
16#A636#: 16#8000#
16#A637#: 16#48#
16#A638#: 16#239#
16#A639#: 16#0#
16#A63A#: 16#0#
16#A63B#: 16#0#
16#A63C#: 16#0#
16#A63D#: 16#0#
1=> ! ! Dump OBDH block queues. Should contain 2 ESR warning commands (LASCO,
SUMER)
1=> ! ! and one normal (MDI open loop)
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#24A2#
16#B14C#: 16#CCCC#
16#B14D#: 16#F16E#
16#B14E#: 16#2CA2#
16#B14F#: 16#CCCC#
16#B150#: 16#F96E#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#2B03#
16#B16D#: 16#7#
16#B16E#: 16#1#
16#B16F#: 16#2B0B#
16#B170#: 16#0#
```

```
1=> ! !
1=> ! ! Now do one run of TC_Exec.Process (will execute first TC of the UVCS
macro)
1=> ! ! Set break at TC_exec.Process return
1=> ! set break 16#4823#
Breakpoint    1  at  16#4823#
    Action:  HALT execution
1=> ! ! Set break at DecodeDst
1=> ! set break 16#47EC#
Breakpoint    2  at  16#47EC#
    Action:  HALT execution
1=> ! ! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
1=> ! set break 16#4780#
Breakpoint    3  at  16#4780#
    Action:  HALT execution
1=> ! set break 16#479B#
Breakpoint    4  at  16#479B#
    Action:  HALT execution
1=> ! ! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and
OBDH.Put
1=> ! set break 16#1E89#
Breakpoint    5  at  16#1E89#
    Action:  HALT execution
1=> ! set break 16#1515#
Breakpoint    6  at  16#1515#
    Action:  HALT execution
1=> ! set break 16#131D#
Breakpoint    7  at  16#131D#
    Action:  HALT execution
1=> ! ! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and
Anomaly.Put
1=> ! set break 16#25E6#
Breakpoint    8  at  16#25E6#
    Action:  HALT execution
1=> ! set break 16#10A8#
Breakpoint    9  at  16#10A8#
    Action:  HALT execution
1=> ! ! Put IC at start of TC_Exec.Process
1=> ! mod/hex \ic:=4803
Proc   1  HALTED      Inst Count: 3957              Clock: 0.001_695_8
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*4803 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Run till DecodeDst
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*4079              Clock:*0.001_756_8
R0: 0000  0000 *0001 *CCBC *CCBD *CCBE *CD3C  0000  IC:*47EC MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF9  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEDST._1417: PSHM      R2,R2
1=> ! ! Run till DecodeOBDH_Blk
1=> ! go
Proc   1  HALTED      Inst Count:*4086              Clock:*0.001_760_2
R0: 0000  0000  0001  CCBC  CCBD  CCBE  CD3C  0000  IC:*479B MK: 0000 SW:*2000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF7  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEOBDH_BLK._1387: PSHM      R1,R12
```

```
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till OBDH_BlkQ.PutRaw
1=> ! go
Proc   1  HALTED      Inst Count:*4155              Clock:*0.001_786_7
R0: 0000 *0003  0001 *0001 *000D *FFE5  CD3C  0000  IC:*25E6 MK: 0000 SW:*4000
R8: 0000  0000  0000 *000D *0001  0000  0000 *FFD2  PI: 0000 SU: 0   TSK:
BP taken at OBDH_BLKQ.PUTRAW._253: PSHM      R0,R13
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till Anomaly.Put
1=> ! go
Proc   1  HALTED      Inst Count:*4601              Clock:*0.001_917_9
R0: 0000 *CCBE *FFBD *002E *FFBD  FFE5  CD3C  0000  IC:*10A8 MK: 0000 SW: 4000
R8: 0000  0000 *000D *0000  0001  0000  0000 *FFBA  PI: 0000 SU: 0   TSK:
BP taken at ANOMALY.PUT._53: PSHM      R0,R14
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify errorcode
1=> ! ! Run till end of TC_Exec.Process
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*4835              Clock:*0.002_027_9
R0: 0000 *0000 *0000 *0000 *0000 *0000 *0000  0000  IC:*4823 MK: 0000 SW:*2000
R8: 0000  0000 *0000  0000 *0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.PROCESS._1441+16#1#: URS       R15
1=> ! !
1=> ! exit
=>
=> ! Dump Anomaly Data
=> exam/hex 16#ffbd#..+2
16#FFBD#: 16#3782#
16#FFBE#: 16#FFFF#
16#FFBF#: 16#D00#
=> exit
Exiting Debugger.
```

6.12 Test2a.log

```
Log file started

TLD Ada Symbolic Debugger  VAX/1750A  V-3.1.H     2-FEB-2010   9:45:15
(c) TLD Systems, Ltd; 1992

1=> ! @setup1.cmd
2=> ! ! Setup1.cmd: UVCS macro with leading DZDUMMY
2=> ! attach tldmps
TLD 1750A Multiple Processor System  v-1.3.3
(c) TLD Systems, Ltd; 1992
2=> ! ! Load COBS and patches
2=> ! load [-.cobsv14]start
```

…

See test1a.log

…

```
2=> ! ! Put an immediate return in OBDH.Put
2=> ! mod/hex 16#131D#:=7FF0
2=> ! !
2=> ! exit
1=> ! !
1=> ! ! Set break at end of original Exp.Putwarning (should not be reached)
1=> ! set break 16#56EC#
Breakpoint    1  at  16#56EC#
    Action:  HALT execution
1=> ! ! Set break at end of patched Exp.PutWarning (should be reached)
1=> ! set break 16#583B#
Breakpoint    2  at  16#583B#
    Action:  HALT execution
1=> ! ! Set break at ExtMacro.Trigg (to verify macro parameter)
1=> ! set break 16#9F11#
Breakpoint    3  at  16#9F11#
    Action:  HALT execution
1=> ! ! Set program counter at start of Exp.PutWarning
1=> ! mod/hex \ic:=56C2
Proc   1  HALTED      Inst Count: 1769              Clock: 0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*56C2 MK: 0000 SW: 4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Dump start of ESR Warning queue data
1=> ! exam/hex 16#B14B#..+21
16#B14B#: 16#0#
16#B14C#: 16#0#
16#B14D#: 16#0#
16#B14E#: 16#0#
16#B14F#: 16#0#
16#B150#: 16#0#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
```

```
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
1=> ! ! Dump first free index in immediate queue (ImFFR)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#0#
1=> ! ! Dump start of immediate queue
1=> ! exam/hex 16#A621#..+18
16#A621#: 16#0#
16#A622#: 16#0#
16#A623#: 16#0#
16#A624#: 16#0#
16#A625#: 16#0#
16#A626#: 16#0#
16#A627#: 16#0#
16#A628#: 16#0#
16#A629#: 16#0#
16#A62A#: 16#0#
16#A62B#: 16#0#
16#A62C#: 16#0#
16#A62D#: 16#0#
16#A62E#: 16#0#
16#A62F#: 16#0#
16#A630#: 16#0#
16#A631#: 16#0#
16#A632#: 16#0#
16#A633#: 16#0#
16#A634#: 16#0#
16#A635#: 16#0#
16#A636#: 16#0#
16#A637#: 16#0#
16#A638#: 16#0#
16#A639#: 16#0#
1=> ! ! Set ESR warning receivers to UVCS only
1=> ! mod/hex 16#CE5E#..+16#F#:=0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
1=> ! ! Dump ESR warning receivers
1=> ! exam/hex 16#CE5E#..+16#F#
16#CE5E#: 16#0#
16#CE5F#: 16#0#
16#CE60#: 16#0#
16#CE61#: 16#0#
16#CE62#: 16#0#
16#CE63#: 16#0#
16#CE64#: 16#0#
16#CE65#: 16#0#
16#CE66#: 16#0#
16#CE67#: 16#0#
16#CE68#: 16#1#
16#CE69#: 16#0#
16#CE6A#: 16#0#
16#CE6B#: 16#0#
16#CE6C#: 16#0#
16#CE6D#: 16#0#
1=> ! ! Run
1=> ! go
```

```
Proc   1  HALTED      Inst Count:*1836              Clock:*0.000_825_4
R0: 0000  0000 *000A *0015  0000  0000  0000  0000  IC:*9F11 MK: 0000 SW: 4000
R8: 0000  0000  0000 *0001 *0009  0000  0000 *FFF1  PI: 0000 SU: 0   TSK:
BP taken at EXTMACRO.TRIGG/$ISECT$: PSHM      R0,R12
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3 = 21 (0x15) at ExtMacro.Trigg
1=> ! ! Run till end of Exp.PutWarning
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*2607              Clock:*0.001_140_2
R0: 0000  0000 *0000 *0000  0000  0000  0000  0000  IC:*583B MK: 0000 SW:*1000
R8: 0000  0000  0000 *0000 *0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at ESR_GYROSET.PROCESS._48+16#4#: URS       R15
1=> ! cancel break/all
1=> ! ! Dump start of ESR Warning queue data
1=> ! exam/hex 16#B14B#..+21
16#B14B#: 16#0#
16#B14C#: 16#0#
16#B14D#: 16#0#
16#B14E#: 16#0#
16#B14F#: 16#0#
16#B150#: 16#0#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
1=> ! ! Dump first free index in immediate queue (ImFFR), should now be 28
(0x1C)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#1C#
```

```
1=> ! ! Dump start of immediate queue (should now contain the macro 21 TCMs)
1=> ! exam/hex 16#A621#..+1c
16#A621#: 16#8#
16#A622#: 16#8000#
16#A623#: 16#E9#
16#A624#: 16#0#
16#A625#: 16#A#
16#A626#: 16#4680#
16#A627#: 16#3782#
16#A628#: 16#E300#
16#A629#: 16#1A82#
16#A62A#: 16#A#
16#A62B#: 16#4680#
16#A62C#: 16#3782#
16#A62D#: 16#EA02#
16#A62E#: 16#2184#
16#A62F#: 16#A#
16#A630#: 16#4680#
16#A631#: 16#3782#
16#A632#: 16#EA01#
16#A633#: 16#2183#
16#A634#: 16#A#
16#A635#: 16#4680#
16#A636#: 16#3782#
16#A637#: 16#E700#
16#A638#: 16#1E82#
16#A639#: 16#8#
16#A63A#: 16#8000#
16#A63B#: 16#48#
16#A63C#: 16#239#
16#A63D#: 16#0#
1=> ! ! Dump OBDH block queues. Should be empty (all zeros)
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#0#
16#B14C#: 16#0#
16#B14D#: 16#0#
16#B14E#: 16#0#
16#B14F#: 16#0#
16#B150#: 16#0#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
```

```
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
16#B16D#: 16#0#
16#B16E#: 16#0#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! !
1=> ! ! Now do one run of TC_Exec.Process (will execute first TC of the UVCS
macro)
1=> ! ! Set break at TC_exec.Process return
1=> ! set break 16#4823#
Breakpoint    1  at  16#4823#
    Action:  HALT execution
1=> ! ! Set break at DecodeDst
1=> ! set break 16#47EC#
Breakpoint    2  at  16#47EC#
    Action:  HALT execution
1=> ! ! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
1=> ! set break 16#4780#
Breakpoint    3  at  16#4780#
    Action:  HALT execution
1=> ! set break 16#479B#
Breakpoint    4  at  16#479B#
    Action:  HALT execution
1=> ! ! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and
OBDH.Put
1=> ! set break 16#1E89#
Breakpoint    5  at  16#1E89#
    Action:  HALT execution
1=> ! set break 16#1515#
Breakpoint    6  at  16#1515#
    Action:  HALT execution
1=> ! set break 16#131D#
Breakpoint    7  at  16#131D#
    Action:  HALT execution
1=> ! ! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and
Anomaly.Put
1=> ! set break 16#25E6#
Breakpoint    8  at  16#25E6#
    Action:  HALT execution
1=> ! set break 16#10A8#
Breakpoint    9  at  16#10A8#
    Action:  HALT execution
1=> ! ! Put IC at start of TC_Exec.Process
1=> ! mod/hex \ic:=4803
```

```
Proc   1  HALTED      Inst Count: 2607              Clock: 0.001_140_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*4803 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Run till DecodeDst
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*2729              Clock:*0.001_202_4
R0: 0000  0000 *0001 *CCBC *CCBD *CCBE *CD3C  0000  IC:*47EC MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF9  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEDST._1417: PSHM      R2,R2
1=> ! ! Run till DecodeOBDH_Msg
1=> ! go
Proc   1  HALTED      Inst Count:*2738              Clock:*0.001_206_2
R0: 0000  0000 *0002  CCBC  CCBD  CCBE  CD3C  0000  IC:*4780 MK: 0000 SW:*2000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF7  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEOBDH_MSG._1379: PSHM      R2,R5
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till IL_Pkt.Put
1=> ! go
Proc   1  HALTED      Inst Count:*2765              Clock:*0.001_215_5
R0: 0000  0000 *0001 *CCBE *0001 *0001  CD3C  0000  IC:*1E89 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at IL_PKT.PUT._8: PSHM      R0,R14
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till RTU.PutRaw
1=> ! go
Proc   1  HALTED      Inst Count:*2778              Clock:*0.001_222_1
R0: 0000  0000 *CCBE *00E9 *0000  0001  CD3C  0000  IC:*1515 MK: 0000 SW:*4000
R8: 0000  0000  0000 *CCC0 *0001 *CCBE  0000 *FFDE  PI: 0000 SU: 0   TSK:
BP taken at RTU.PUTRAW._119: PSHM      R1,R6
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3-R4 contain dummy interrogation 00E9,0000
1=> ! ! Run till OBDH.Put
1=> ! go
Proc   1  HALTED      Inst Count:*2784              Clock:*0.001_225_1
R0: 0000  0000 *0001  00E9  0000  0001  CD3C  0000  IC:*131D MK: 0000 SW: 4000
R8: 0000  0000  0000  CCC0  0001  CCBE  0000 *FFD6  PI: 0000 SU: 0   TSK:
BP taken at OBDH.PUT._59: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till end of TC_Exec.Process
1=> ! go
Proc   1  HALTED      Inst Count:*2938              Clock:*0.001_298_3
R0: 0000  0000 *0000 *0000  0000 *0000 *0000  0000  IC:*4823 MK: 0000 SW:*2000
R8: 0000  0000  0000 *0000 *0000 *0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.PROCESS._1441+16#1#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! !
1=> ! exit
=>
=> exit
```

```
Exiting Debugger.
```

6.13 Test2b.log

```
Log file started

TLD Ada Symbolic Debugger  VAX/1750A  V-3.1.H     2-FEB-2010  10:06:11
(c) TLD Systems, Ltd; 1992

1=> ! @setup1.cmd
2=> ! ! Setup1.cmd: UVCS macro with leading DZDUMMY
2=> ! attach tldmps
TLD 1750A Multiple Processor System  v-1.3.3
(c) TLD Systems, Ltd; 1992
2=> ! ! Load COBS and patches
2=> ! load [-.cobsv14]start
```

…

See test1a.log

…

```
2=> ! ! Put an immediate return in OBDH.Put
2=> ! mod/hex 16#131D#:=7FF0
2=> ! !
2=> ! exit
1=> ! !
1=> ! ! Set break at end of original Exp.Putwarning (should not be reached)
1=> ! set break 16#56EC#
Breakpoint    1  at  16#56EC#
    Action:  HALT execution
1=> ! ! Set break at end of patched Exp.PutWarning (should be reached)
1=> ! set break 16#583B#
Breakpoint    2  at  16#583B#
    Action:  HALT execution
1=> ! ! Set break at ExtMacro.Trigg (to verify macro parameter)
1=> ! set break 16#9F11#
Breakpoint    3  at  16#9F11#
    Action:  HALT execution
1=> ! ! Set program counter at start of Exp.PutWarning
1=> ! mod/hex \ic:=56C2
Proc   1  HALTED      Inst Count: 1769              Clock: 0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*56C2 MK: 0000 SW: 4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Dump OBDH block queues, start contains ESR Warning queue data
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#0#
16#B14C#: 16#0#
16#B14D#: 16#0#
16#B14E#: 16#0#
16#B14F#: 16#0#
16#B150#: 16#0#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
```

```
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
16#B16D#: 16#0#
16#B16E#: 16#0#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! ! Dump first free index in immediate queue (ImFFR)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#0#
1=> ! ! Dump start of immediate queue
1=> ! exam/hex 16#A621#..+18
16#A621#: 16#0#
16#A622#: 16#0#
16#A623#: 16#0#
16#A624#: 16#0#
16#A625#: 16#0#
16#A626#: 16#0#
16#A627#: 16#0#
16#A628#: 16#0#
16#A629#: 16#0#
16#A62A#: 16#0#
16#A62B#: 16#0#
16#A62C#: 16#0#
16#A62D#: 16#0#
16#A62E#: 16#0#
16#A62F#: 16#0#
16#A630#: 16#0#
16#A631#: 16#0#
16#A632#: 16#0#
16#A633#: 16#0#
16#A634#: 16#0#
16#A635#: 16#0#
16#A636#: 16#0#
16#A637#: 16#0#
```

```
16#A638#: 16#0#
16#A639#: 16#0#
1=> ! ! Set ESR warning receivers to UVCS only
1=> ! mod/hex 16#CE5E#..+16#F#:=0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
1=> ! ! Dump ESR warning receivers
1=> ! exam/hex 16#CE5E#..+16#F#
16#CE5E#: 16#0#
16#CE5F#: 16#1#
16#CE60#: 16#1#
16#CE61#: 16#1#
16#CE62#: 16#1#
16#CE63#: 16#1#
16#CE64#: 16#1#
16#CE65#: 16#1#
16#CE66#: 16#1#
16#CE67#: 16#1#
16#CE68#: 16#1#
16#CE69#: 16#0#
16#CE6A#: 16#0#
16#CE6B#: 16#0#
16#CE6C#: 16#0#
16#CE6D#: 16#0#
1=> ! ! Run
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*4729              Clock:*0.001_922_3
R0: 0000  0000 *000A *0015 *000C *0005 *84E0  0000  IC:*9F11 MK: 0000 SW: 4000
R8: 0000  0000  0000 *0001 *0009  0000  0000 *FFF1  PI: 0000 SU: 0   TSK:
BP taken at EXTMACRO.TRIGG/$ISECT$: PSHM      R0,R12
1=> ! ! Verify R3 = 21 (0x15) at ExtMacro.Trigg
1=> ! ! Run till end of Exp.PutWarning
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*5500              Clock:*0.002_237_1
R0: 0000  0000 *0000 *0000 *0000 *0000 *0000  0000  IC:*583B MK: 0000 SW:*1000
R8: 0000  0000  0000 *0000 *0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at ESR_GYROSET.PROCESS._48+16#4#: URS       R15
1=> ! cancel break/all
1=> ! ! Dump OBDH block queues, start contains ESR Warning queue data
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#1CA2#
16#B14C#: 16#CCCC#
16#B14D#: 16#E96E#
16#B14E#: 16#10A2#
16#B14F#: 16#CCCC#
16#B150#: 16#DD6E#
16#B151#: 16#14A2#
16#B152#: 16#CCCC#
16#B153#: 16#E16E#
16#B154#: 16#18A2#
16#B155#: 16#CCCC#
16#B156#: 16#E56E#
16#B157#: 16#20A2#
16#B158#: 16#CCCC#
16#B159#: 16#ED6E#
```

```
16#B15A#: 16#24A2#
16#B15B#: 16#CCCC#
16#B15C#: 16#F16E#
16#B15D#: 16#28A2#
16#B15E#: 16#CCCC#
16#B15F#: 16#F56E#
16#B160#: 16#2CA2#
16#B161#: 16#CCCC#
16#B162#: 16#F96E#
16#B163#: 16#30A2#
16#B164#: 16#CCCC#
16#B165#: 16#FD6E#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
16#B16D#: 16#0#
16#B16E#: 16#0#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! ! Dump first free index in immediate queue (ImFFR), should now be 28
(0x1C)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#1C#
1=> ! ! Dump start of immediate queue (should now contain the macro 21 TCMs)
1=> ! exam/hex 16#A621#..+1c
16#A621#: 16#8#
16#A622#: 16#8000#
16#A623#: 16#E9#
16#A624#: 16#0#
16#A625#: 16#A#
16#A626#: 16#4680#
16#A627#: 16#3782#
16#A628#: 16#E300#
16#A629#: 16#1A82#
16#A62A#: 16#A#
16#A62B#: 16#4680#
16#A62C#: 16#3782#
16#A62D#: 16#EA02#
16#A62E#: 16#2184#
16#A62F#: 16#A#
16#A630#: 16#4680#
16#A631#: 16#3782#
16#A632#: 16#EA01#
16#A633#: 16#2183#
16#A634#: 16#A#
16#A635#: 16#4680#
16#A636#: 16#3782#
16#A637#: 16#E700#
16#A638#: 16#1E82#
16#A639#: 16#8#
16#A63A#: 16#8000#
```

```
16#A63B#: 16#48#
16#A63C#: 16#239#
16#A63D#: 16#0#
1=> ! !
1=> ! ! Now do one run of TC_Exec.Process (will execute first TC of the UVCS
macro)
1=> ! ! Set break at TC_exec.Process return
1=> ! set break 16#4823#
Breakpoint    1  at  16#4823#
    Action:  HALT execution
1=> ! ! Set break at DecodeDst
1=> ! set break 16#47EC#
Breakpoint    2  at  16#47EC#
    Action:  HALT execution
1=> ! ! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
1=> ! set break 16#4780#
Breakpoint    3  at  16#4780#
    Action:  HALT execution
1=> ! set break 16#479B#
Breakpoint    4  at  16#479B#
    Action:  HALT execution
1=> ! ! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and
OBDH.Put
1=> ! set break 16#1E89#
Breakpoint    5  at  16#1E89#
    Action:  HALT execution
1=> ! set break 16#1515#
Breakpoint    6  at  16#1515#
    Action:  HALT execution
1=> ! set break 16#131D#
Breakpoint    7  at  16#131D#
    Action:  HALT execution
1=> ! ! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and
Anomaly.Put
1=> ! set break 16#25E6#
Breakpoint    8  at  16#25E6#
    Action:  HALT execution
1=> ! set break 16#10A8#
Breakpoint    9  at  16#10A8#
    Action:  HALT execution
1=> ! ! Put IC at start of TC_Exec.Process
1=> ! mod/hex \ic:=4803
Proc   1  HALTED      Inst Count: 5500              Clock: 0.002_237_1
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*4803 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Run till DecodeDst
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*5622              Clock:*0.002_299_3
R0: 0000  0000 *0001 *CCBC *CCBD *CCBE *CD3C  0000  IC:*47EC MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF9  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEDST._1417: PSHM      R2,R2
1=> ! ! Run till DecodeOBDH_Msg
1=> ! go
Proc   1  HALTED      Inst Count:*5631              Clock:*0.002_303_1
```

```
R0: 0000  0000 *0002  CCBC  CCBD  CCBE  CD3C  0000  IC:*4780 MK: 0000 SW:*2000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF7  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEOBDH_MSG._1379: PSHM      R2,R5
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till IL_Pkt.Put
1=> ! go
Proc   1  HALTED      Inst Count:*5658              Clock:*0.002_312_4
R0: 0000  0000 *0001 *CCBE *0001 *0001  CD3C  0000  IC:*1E89 MK: 0000 SW:*1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF0  PI: 0000 SU: 0   TSK:
BP taken at IL_PKT.PUT._8: PSHM      R0,R14
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till RTU.PutRaw
1=> ! go
Proc   1  HALTED      Inst Count:*5671              Clock:*0.002_319
R0: 0000  0000 *CCBE *00E9 *0000  0001  CD3C  0000  IC:*1515 MK: 0000 SW:*4000
R8: 0000  0000  0000 *CCC0 *0001 *CCBE  0000 *FFDE  PI: 0000 SU: 0   TSK:
BP taken at RTU.PUTRAW._119: PSHM      R1,R6
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Verify R3-R4 contain dummy interrogation 00E9,0000
1=> ! ! Run till OBDH.Put
1=> ! go
Proc   1  HALTED      Inst Count:*5677              Clock:*0.002_322
R0: 0000  0000 *0001  00E9  0000  0001  CD3C  0000  IC:*131D MK: 0000 SW: 4000
R8: 0000  0000  0000  CCC0  0001  CCBE  0000 *FFD6  PI: 0000 SU: 0   TSK:
BP taken at OBDH.PUT._59: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till end of TC_Exec.Process
1=> ! go
Proc   1  HALTED      Inst Count:*5831              Clock:*0.002_395_2
R0: 0000  0000 *0000 *0000  0000 *0000 *0000  0000  IC:*4823 MK: 0000 SW:*2000
R8: 0000  0000  0000 *0000 *0000 *0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.PROCESS._1441+16#1#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! !
1=> ! exit
=>
=> exit
Exiting Debugger.
```

6.14 Test2c.log

```
Log file started

TLD Ada Symbolic Debugger  VAX/1750A  V-3.1.H     2-FEB-2010  10:32:54
(c) TLD Systems, Ltd; 1992

1=> ! @setup2.cmd
2=> ! ! Setup2.cmd: UVCS macro without leading DZDUMMY
2=> ! attach tldmps
```

```
TLD 1750A Multiple Processor System  v-1.3.3
(c) TLD Systems, Ltd; 1992
2=> ! ! Load COBS and patches
2=> ! load [-.cobsv14]start
```

...

See test1c.log

...

```
2=> ! ! Put an immediate return in OBDH.Put
2=> ! mod/hex 16#131D#:=7FF0
2=> ! !
2=> ! exit
1=> ! !
1=> ! ! Set break at end of original Exp.Putwarning (should not be reached)
1=> ! set break 16#56EC#
Breakpoint    1  at  16#56EC#
    Action:  HALT execution
1=> ! ! Set break at end of patched Exp.PutWarning (should be reached)
1=> ! set break 16#583B#
Breakpoint    2  at  16#583B#
    Action:  HALT execution
1=> ! ! Set break at ExtMacro.Trigg (to verify macro parameter)
1=> ! set break 16#9F11#
Breakpoint    3  at  16#9F11#
    Action:  HALT execution
1=> ! ! Set program counter at start of Exp.PutWarning
1=> ! mod/hex \ic:=56C2
Proc   1  HALTED      Inst Count: 1769              Clock: 0.000_807_2
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*56C2 MK: 0000 SW: 4000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Dump OBDH block queues, start contains ESR Warning queue data
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#0#
16#B14C#: 16#0#
16#B14D#: 16#0#
16#B14E#: 16#0#
16#B14F#: 16#0#
16#B150#: 16#0#
16#B151#: 16#0#
16#B152#: 16#0#
16#B153#: 16#0#
16#B154#: 16#0#
16#B155#: 16#0#
16#B156#: 16#0#
16#B157#: 16#0#
16#B158#: 16#0#
16#B159#: 16#0#
16#B15A#: 16#0#
16#B15B#: 16#0#
16#B15C#: 16#0#
16#B15D#: 16#0#
16#B15E#: 16#0#
16#B15F#: 16#0#
16#B160#: 16#0#
16#B161#: 16#0#
```

```
16#B162#: 16#0#
16#B163#: 16#0#
16#B164#: 16#0#
16#B165#: 16#0#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
16#B16D#: 16#0#
16#B16E#: 16#0#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! ! Dump first free index in immediate queue (ImFFR)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#0#
1=> ! ! Dump start of immediate queue
1=> ! exam/hex 16#A621#..+18
16#A621#: 16#0#
16#A622#: 16#0#
16#A623#: 16#0#
16#A624#: 16#0#
16#A625#: 16#0#
16#A626#: 16#0#
16#A627#: 16#0#
16#A628#: 16#0#
16#A629#: 16#0#
16#A62A#: 16#0#
16#A62B#: 16#0#
16#A62C#: 16#0#
16#A62D#: 16#0#
16#A62E#: 16#0#
16#A62F#: 16#0#
16#A630#: 16#0#
16#A631#: 16#0#
16#A632#: 16#0#
16#A633#: 16#0#
16#A634#: 16#0#
16#A635#: 16#0#
16#A636#: 16#0#
16#A637#: 16#0#
16#A638#: 16#0#
16#A639#: 16#0#
1=> ! ! Set ESR warning receivers to UVCS only
1=> ! mod/hex 16#CE5E#..+16#F#:=0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
1=> ! ! Dump ESR warning receivers
1=> ! exam/hex 16#CE5E#..+16#F#
16#CE5E#: 16#0#
16#CE5F#: 16#1#
16#CE60#: 16#1#
16#CE61#: 16#1#
16#CE62#: 16#1#
16#CE63#: 16#1#
```

```
16#CE64#: 16#1#
16#CE65#: 16#1#
16#CE66#: 16#1#
16#CE67#: 16#1#
16#CE68#: 16#1#
16#CE69#: 16#0#
16#CE6A#: 16#0#
16#CE6B#: 16#0#
16#CE6C#: 16#0#
16#CE6D#: 16#0#
1=> ! ! Run
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*4729              Clock:*0.001_922_3
R0: 0000  0000 *000A *0015 *000C *0005 *84E0  0000  IC:*9F11 MK: 0000 SW: 4000
R8: 0000  0000  0000 *0001 *0009  0000  0000 *FFF1  PI: 0000 SU: 0   TSK:
BP taken at EXTMACRO.TRIGG/$ISECT$: PSHM      R0,R12
1=> ! ! Verify R3 = 21 (0x15) at ExtMacro.Trigg
1=> ! ! Run till end of Exp.PutWarning
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*5384              Clock:*0.002_191_4
R0: 0000  0000 *0000 *0000 *0000 *0000 *0000  0000  IC:*583B MK: 0000 SW:*1000
R8: 0000  0000  0000 *0000 *0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at ESR_GYROSET.PROCESS._48+16#4#: URS       R15
1=> ! cancel break/all
1=> ! ! Dump OBDH block queues, start contains ESR Warning queue data
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#1CA2#
16#B14C#: 16#CCCC#
16#B14D#: 16#E96E#
16#B14E#: 16#10A2#
16#B14F#: 16#CCCC#
16#B150#: 16#DD6E#
16#B151#: 16#14A2#
16#B152#: 16#CCCC#
16#B153#: 16#E16E#
16#B154#: 16#18A2#
16#B155#: 16#CCCC#
16#B156#: 16#E56E#
16#B157#: 16#20A2#
16#B158#: 16#CCCC#
16#B159#: 16#ED6E#
16#B15A#: 16#24A2#
16#B15B#: 16#CCCC#
16#B15C#: 16#F16E#
16#B15D#: 16#28A2#
16#B15E#: 16#CCCC#
16#B15F#: 16#F56E#
16#B160#: 16#2CA2#
16#B161#: 16#CCCC#
16#B162#: 16#F96E#
16#B163#: 16#30A2#
16#B164#: 16#CCCC#
16#B165#: 16#FD6E#
```

```
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#0#
16#B16D#: 16#0#
16#B16E#: 16#0#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! ! Dump first free index in immediate queue (ImFFR), should now be 24
(0x18)
1=> ! exam/hex 16#ABFD#
16#ABFD#: 16#18#
1=> ! ! Dump start of immediate queue (should now contain the macro 21 TCMs)
1=> ! exam/hex 16#A621#..+1c
16#A621#: 16#A#
16#A622#: 16#4680#
16#A623#: 16#3782#
16#A624#: 16#E300#
16#A625#: 16#1A82#
16#A626#: 16#A#
16#A627#: 16#4680#
16#A628#: 16#3782#
16#A629#: 16#EA02#
16#A62A#: 16#2184#
16#A62B#: 16#A#
16#A62C#: 16#4680#
16#A62D#: 16#3782#
16#A62E#: 16#EA01#
16#A62F#: 16#2183#
16#A630#: 16#A#
16#A631#: 16#4680#
16#A632#: 16#3782#
16#A633#: 16#E700#
16#A634#: 16#1E82#
16#A635#: 16#8#
16#A636#: 16#8000#
16#A637#: 16#48#
16#A638#: 16#239#
16#A639#: 16#0#
16#A63A#: 16#0#
16#A63B#: 16#0#
16#A63C#: 16#0#
16#A63D#: 16#0#
1=> ! !
1=> ! ! Now do one run of TC_Exec.Process (will execute first TC of the UVCS
macro)
1=> ! ! Set break at TC_exec.Process return
1=> ! set break 16#4823#
Breakpoint    1  at  16#4823#
    Action:  HALT execution
1=> ! ! Set break at DecodeDst
1=> ! set break 16#47EC#
```

```
Breakpoint    2  at  16#47EC#
    Action:  HALT execution
1=> ! ! Set break at DecodeOBDH_Msg and DecodeOBDH_Blk
1=> ! set break 16#4780#
Breakpoint    3  at  16#4780#
    Action:  HALT execution
1=> ! set break 16#479B#
Breakpoint    4  at  16#479B#
    Action:  HALT execution
1=> ! ! For DZDUMMY (OBDH_Msg) set breaks at IL_Pkt.Put, RTU.PutRaw and
OBDH.Put
1=> ! set break 16#1E89#
Breakpoint    5  at  16#1E89#
    Action:  HALT execution
1=> ! set break 16#1515#
Breakpoint    6  at  16#1515#
    Action:  HALT execution
1=> ! set break 16#131D#
Breakpoint    7  at  16#131D#
    Action:  HALT execution
1=> ! ! For UVCS OBDH block commands set breaks at OBDH_BlkQ.PutRaw and
Anomaly.Put
1=> ! set break 16#25E6#
Breakpoint    8  at  16#25E6#
    Action:  HALT execution
1=> ! set break 16#10A8#
Breakpoint    9  at  16#10A8#
    Action:  HALT execution
1=> ! ! Put IC at start of TC_Exec.Process
1=> ! mod/hex \ic:=4803
Proc   1  HALTED      Inst Count: 5384              Clock: 0.002_191_4
R0: 0000  0000  0000  0000  0000  0000  0000  0000  IC:*4803 MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000  FFFF  PI: 0000 SU: 0   TSK:
1=> ! ! Run till DecodeDst
1=> ! go
1=> ! wait
Proc   1  HALTED      Inst Count:*5506              Clock:*0.002_252_4
R0: 0000  0000 *0001 *CCBC *CCBD *CCBE *CD3C  0000  IC:*47EC MK: 0000 SW: 1000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF9  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEDST._1417: PSHM      R2,R2
1=> ! ! Run till DecodeOBDH_Blk
1=> ! go
Proc   1  HALTED      Inst Count:*5513              Clock:*0.002_255_8
R0: 0000  0000  0001  CCBC  CCBD  CCBE  CD3C  0000  IC:*479B MK: 0000 SW:*2000
R8: 0000  0000  0000  0000  0000  0000  0000 *FFF7  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.DECODEOBDH_BLK._1387: PSHM      R1,R12
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till OBDH_BlkQ.PutRaw
1=> ! go
Proc   1  HALTED      Inst Count:*5582              Clock:*0.002_282_3
R0: 0000 *0003  0001 *0001 *000D *FFE5  CD3C  0000  IC:*25E6 MK: 0000 SW:*4000
R8: 0000  0000  0000 *000D *0001  0000  0000 *FFD2  PI: 0000 SU: 0   TSK:
BP taken at OBDH_BLKQ.PUTRAW._253: PSHM      R0,R13
1=> ! wait
```

```
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Run till end of TC_Exec.Process
1=> ! go
Proc   1  HALTED      Inst Count:*6046              Clock:*0.002_437_9
R0: 0000 *0000 *0000 *0000 *0000 *0000 *0000  0000  IC:*4823 MK: 0000 SW:*2000
R8: 0000  0000  0000 *0000 *0000  0000  0000 *FFFF  PI: 0000 SU: 0   TSK:
BP taken at TC_EXEC.PROCESS._1441+16#1#: URS       R15
1=> ! wait
%TLDDBG-E-PROCNORUN - Must run processor first
1=> ! ! Dump OBDH block queues, start contains ESR Warning queue data, followed
by first UVCS cmd.
1=> ! exam/hex 16#B14B#..16#B170#
16#B14B#: 16#1CA2#
16#B14C#: 16#CCCC#
16#B14D#: 16#E96E#
16#B14E#: 16#10A2#
16#B14F#: 16#CCCC#
16#B150#: 16#DD6E#
16#B151#: 16#14A2#
16#B152#: 16#CCCC#
16#B153#: 16#E16E#
16#B154#: 16#18A2#
16#B155#: 16#CCCC#
16#B156#: 16#E56E#
16#B157#: 16#20A2#
16#B158#: 16#CCCC#
16#B159#: 16#ED6E#
16#B15A#: 16#24A2#
16#B15B#: 16#CCCC#
16#B15C#: 16#F16E#
16#B15D#: 16#28A2#
16#B15E#: 16#CCCC#
16#B15F#: 16#F56E#
16#B160#: 16#2CA2#
16#B161#: 16#CCCC#
16#B162#: 16#F96E#
16#B163#: 16#30A2#
16#B164#: 16#CCCC#
16#B165#: 16#FD6E#
16#B166#: 16#0#
16#B167#: 16#0#
16#B168#: 16#0#
16#B169#: 16#0#
16#B16A#: 16#0#
16#B16B#: 16#0#
16#B16C#: 16#3782#
16#B16D#: 16#E300#
16#B16E#: 16#1A82#
16#B16F#: 16#0#
16#B170#: 16#0#
1=> ! !
1=> ! exit
=>
=> exit
Exiting Debugger.
```