Ultra96: Test Applications

Overview

With the Hello World application operational, we will now move on to more advanced test applications. Xilinx provides a Memory Test as well as a Peripherals Test in the built-in templates for example applications.

This Tutorial assumes that you have already completed the Hardware Platform and Hello World tutorials. Your starting point will be the SDK project after the Hello World tutorial is complete.

Objectives

When this tutorial is complete, you will be able to:

- Add the Memory Test application
- Add the Peripherals Test application
- Run both test applications
- Edit the memory test to increase the test range

Experiment Setup

Software

The software used to test this reference design is:

- Windows-7 64-bit
- Xilinx SDK 2018.2
- USB-JTAG and USB-UART drivers

Hardware

The hardware setup used to test this reference design includes:

- Win-7 PC with the following recommended memory[1]
    - 4 GB Typical and 5 GB Peak RAM available for the Xilinx tools to complete a XCZU3EG design
- Ultra96
- 96Boards Power Supply
- USB-JTAG (any compatible Xilinx USB-JTAG)
    - Avnet USB-to-JTAG/UART Pod (available September 2018)
    - Digilent HS3 (http://avnet.me/jtaghs3) with flyleads
- USB-UART
    - Avnet USB-to-JTAG/UART Pod (available September 2018)
    - Any other USB-UART dongle

---

[1] Refer to https://www.xilinx.com/products/design-tools/vivado/memory.html

Experiment 1: Create Memory and Peripherals Test Applications

Similar to Hello World, use templates to create two very useful test applications.

1. Launch SDK and open the workspace from the Hello World project.
2. In SDK, select **File → New → Application Project**.
3. In the **Project Name** field type in `Mem_Test`. Change the **BSP** to the existing StandAlone BSP. Click **Next >**.

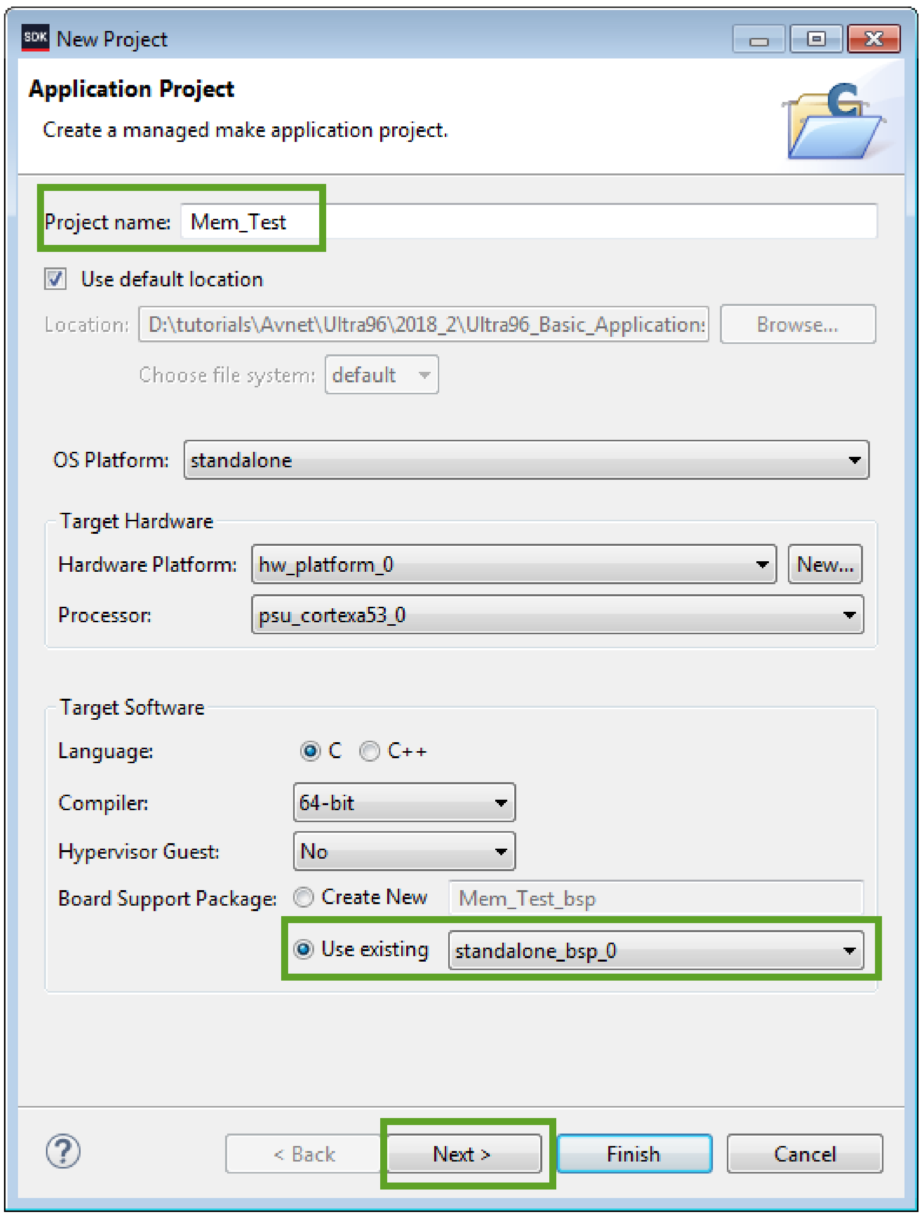

Figure 1 - New Application Wizard

4. Select **Memory Tests** from the *Available Templates* field. Click **Finish**.

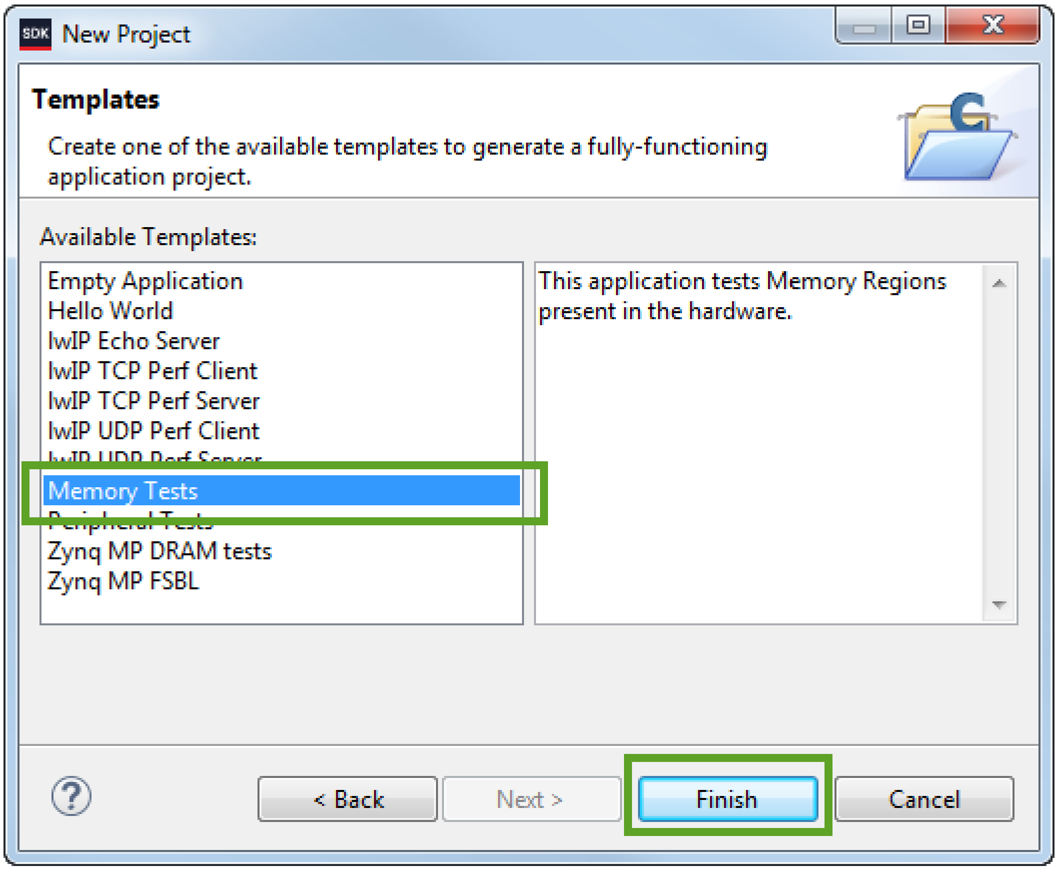

Figure 2 – New Application Project: Hello World

5. Repeat steps 2 through 4 with the following options:
    a. Project Name = **Periph_Test**
    b. BSP = **standalone_bsp_0**
    c. Template = **Peripheral Tests**

6. Repeat steps 2 through 4 with the following options:
    a. Project Name = **ZynqMP_DRAM_Test**
    b. BSP = **standalone_bsp_0**
    c. Template = **Zynq MP DRAM tests**

When complete, *Project Explorer* should look similar to below.

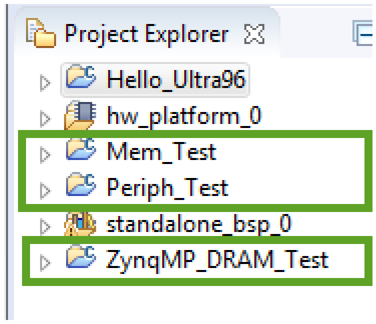

Figure 3 – Project Explorer with New Apps Highlighted

Experiment 2: Run the Applications

1. Follow the instructions in the Hello World tutorial to configure the Ultra96 for JTAG Boot and plug in the USB-JTAG and USB-UART cables or pod.

2. Press the POWER button (SW3) to turn the board on if necessary.

3. If you are continuing from the Tutorial 02 and the board is still on and configured, you will need to manually reset the board by pressing the POR_B button (SW1).

4. Program the bitstream so that the Green DONE LED is lit.

5. Launch a terminal with 115200 baud rate as in Tutorial 02.

6. Continue by right-clicking on the Mem_Test application, selecting **Run As**..., as previously shown in the Hello World tutorial.

7. When asked to terminate the old configuration, select **Yes**.

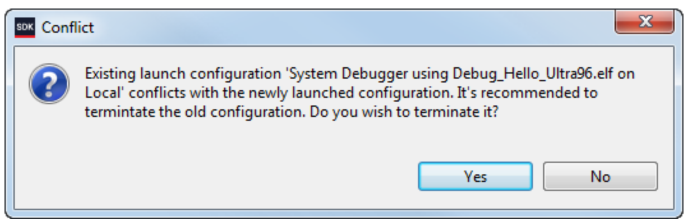

Figure 4 – Terminate Old Configuration

When done you should see this terminal message for Mem_Test.

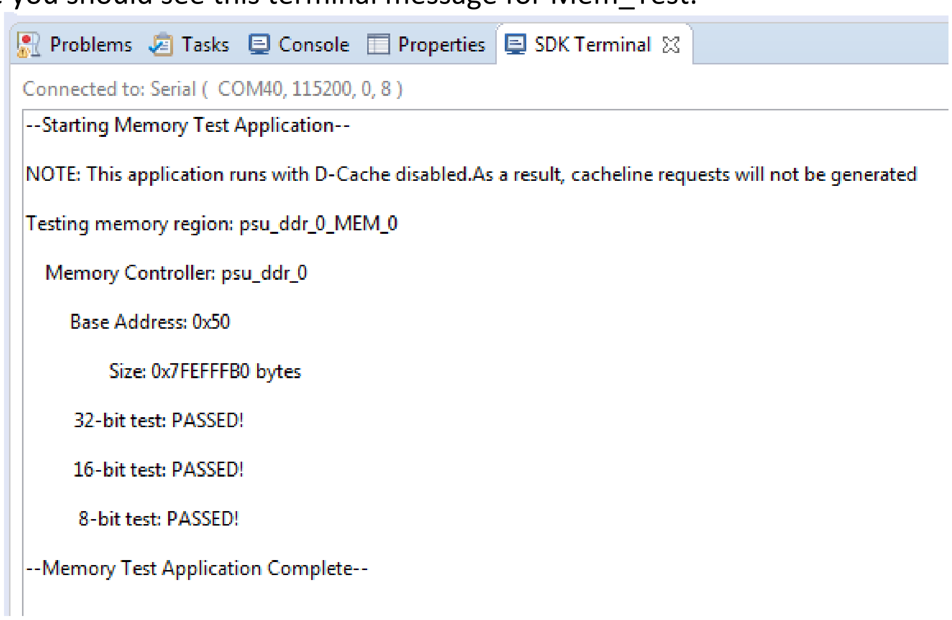

Figure 5 – Memory Test Console

The Periph_Test will require a little manual editing. The default Hardware Platform Preset connects the PSU UART0, which is connected to the Bluetooth Radio. This radio is not going to be responsive without some configuring, which usually takes place with Linux or other driver. Therefore, for this simple bare metal test, we will simply comment out the UART0 tests.

8. In SDK, go to **Project Explorer → Periph_Test → src** and double-click `testperiph.c` to open it.

To make it easier to reference code, we'll turn on line numbers now.

9. Turn on line numbers by right-clicking in the left-hand column, or use the **Window → Preferences** dialog. Go to **General → Editors → Text Editors** and then check the box for *Show line numbers*. Click **OK**.

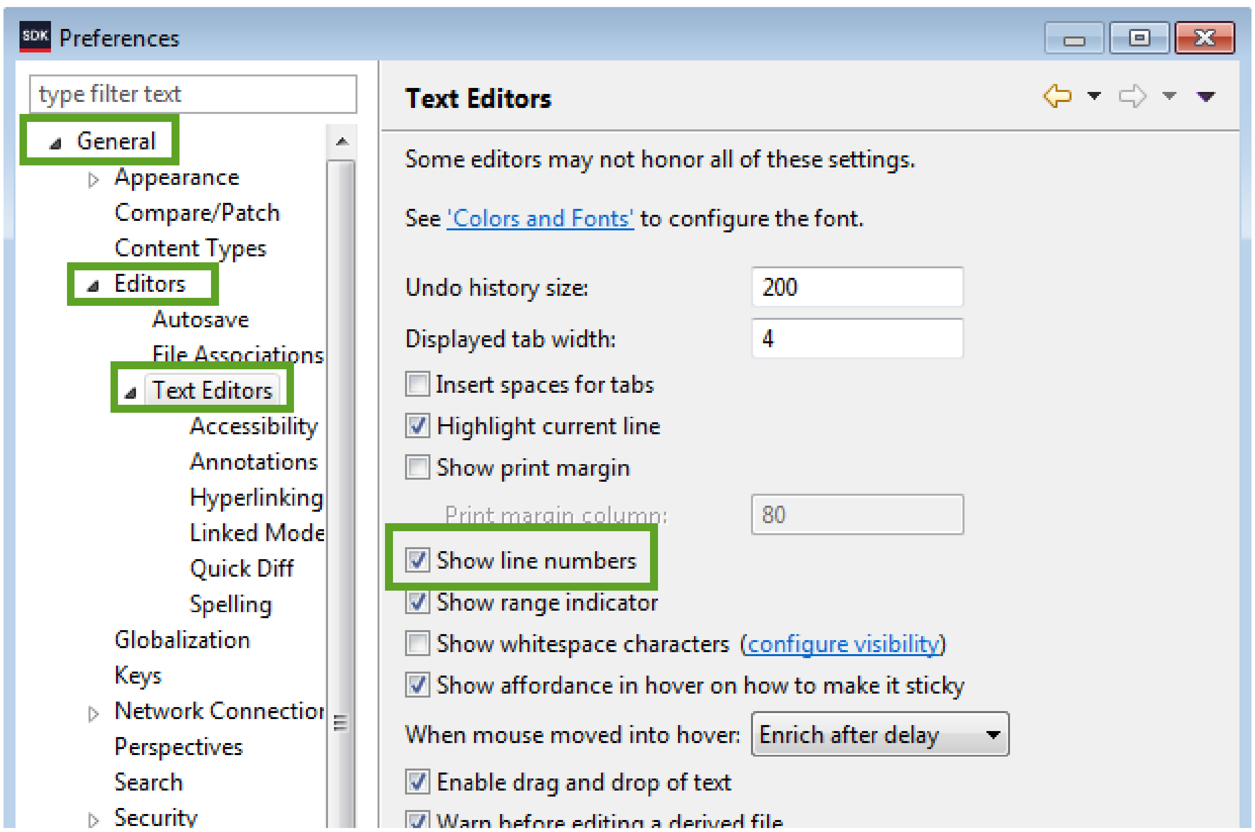

Figure 6 – Show Line Numbers

10. Comment out the psu_uart_0 code from lines 852 to 880.

system.hdf | system.mss | *testperiph.c

```
851
852 //    {
853 //       int Status;
854 //
855 //       print("\r\nRunning UartPsPolledExample() for psu_uart_0...\r\n");
856 //       Status = UartPsPolledExample(XPAR_PSU_UART_0_DEVICE_ID);
857 //       if (Status == 0) {
858 //          print("UartPsPolledExample PASSED\r\n");
859 //       }
860 //       else {
861 //          print("UartPsPolledExample FAILED\r\n");
862 //       }
863 //    }
864 //    {
865 //       int Status;
866 //
867 //       print("\r\n Running Interrupt Test for psu_uart_0...\r\n");
868 //
869 //       Status = UartPsIntrExample(&intc, &psu_uart_0, \
870 //                                  XPAR_PSU_UART_0_DEVICE_ID, \
871 //                                  XPAR_PSU_UART_0_INTR);
872 //
873 //       if (Status == 0) {
874 //          print("UartPsIntrExample PASSED\r\n");
875 //       }
876 //       else {
877 //          print("UartPsIntrExample FAILED\r\n");
878 //       }
879 //
880 //    }
881
```

Figure 7 – psu_uart_0 Peripheral Tests Commented Out

11. Save the file with the Save icon or Ctrl-S.
12. Now repeat Steps 3 through 7 for and Periph_Test to program and run this application.

```
---Entering main---

 Running ScuGicSelfTestExample() for psu_acpu_gic...
ScuGicSelfTestExample PASSED
ScuGic Interrupt Setup PASSED

 Running XZDma_SelfTestExample() for psu_adma_1...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_1...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_adma_2...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_2...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_adma_3...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_3...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_adma_4...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_4...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_adma_5...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_5...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_adma_6...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_6...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_adma_7...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_7...
ZDMA Simple Example PASSED

 Running SysMonPsuPolledPrintfExample() for psu_ams...

Entering the SysMon Polled Example.

The Current Temperature is 46.825 Centigrades.
The Maximum Temperature is 47.066 Centigrades.
The Minimum Temperature is 46.149 Centigrades.

The Current VCCINT is 0.856 Volts.
The Maximum VCCINT is 0.861 Volts.
The Minimum VCCINT is 0.854 Volts.

The Current VCCAUX is 1.813 Volts.
The Maximum VCCAUX is 1.814 Volts.
The Minimum VCCAUX is 1.809 Volts.

Exiting the SysMon Polled Example.
SysMonPsuPolledPrintfExample PASSED

 Running SysMonPsuIntrExample()  for psu_ams...

Entering the SysMonPsu Interrupt Example.

The Current Temperature is 46.514 Centigrade.

The Current VCCINT is 0.855 Volts.

The Current VCCAUX is 1.810 Volts.

Temperature Alarm(0) HIGH Threshold is 36.512 Centigrade.
Temperature Alarm(0) LOW Threshold is 26.510 Centigrade.
VCCINT Alarm(1) HIGH Threshold is 0.655 Volts.
VCCINT Alarm(1) LOW Threshold is 1.055 Volts.
VCCAUX Alarm(3) HIGH Threshold is 1.610 Volts.
VCCAUX Alarm(3) LOW Threshold is 2.010 Volts.

Alarm 0 - Temperature alarm has occured

The Current Temperature is 46.234 Centigrade.
The Maximum Temperature is 47.105 Centigrade.
The Minimum Temperature is 46.219 Centigrade.

The Current VCCINT is 0.859 Volts.
The Maximum VCCINT is 0.861 Volts.
The Minimum VCCINT is 0.853 Volts.

The Current VCCAUX is 1.810 Volts.
The Maximum VCCAUX is 1.813 Volts.
The Minimum VCCAUX is 1.809 Volts.

Exiting the SysMon Interrupt Example.
SysMonPsu IntrExample PASSED

 Running XCsuDma_SelfTestExample() for psu_csudma...
XCsuDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_csudma...
CSUDMA Interrupt Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_0...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_0...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_1...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_1...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_2...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_2...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_3...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_3...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_4...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_4...
ZDMA Simple Example PASSED


 Running XZDma_SelfTestExample() for psu_gdma_5...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_5...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_6...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_6...
ZDMA Simple Example PASSED

 Running XZDma_SelfTestExample() for psu_gdma_7...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_gdma_7...
ZDMA Simple Example PASSED

 Running IicPsSelfTestExample() for psu_i2c_1...
IicPsSelfTestExample PASSED

 Running XZDma_SelfTestExample() for psu_adma_0...
XZDma_SelfTestExample PASSED

 Running Interrupt Test  for psu_adma_0...
ZDMA Simple Example PASSED

 Running SpiPsSelfTestExample() for psu_spi_0...
SpiPsSelfTestExample PASSED

 Running SpiPsSelfTestExample() for psu_spi_1...
SpiPsSelfTestExample PASSED

 Running Interrupt Test  for psu_ttc_0...
TtcIntrExample PASSED

 Running Interrupt Test  for psu_ttc_1...
TtcIntrExample PASSED

 Running Interrupt Test  for psu_ttc_2...
TtcIntrExample PASSED

 Running Interrupt Test  for psu_ttc_3...
TtcIntrExample PASSED

 Running WdtPsSelfTestExample() for psu_wdt_0...
WdtPsSelfTestExample PASSED

 Running WdtPsSelfTestExample() for psu_wdt_1...
WdtPsSelfTestExample PASSED
---Exiting main---
```

Figure 8 – Periphal Tests Console

The Zynq MP DRAM Test is a bit more complex. It is explained in detail in the **`ZYNQMP_DRAM_DIAGNOSTICS_TEST.docx`** document that is included in the `ZynqMPDRAM_Test\src` directory of your SDK Workspace.

A couple of the test outputs are shown below for Ultra96.

```
*****************************************************************************
 Zynq MPSoC
 DRAM Diagnostics Test (A53)
*****************************************************************************
 Select one of the options below:

 +-------------------------------------------------------------------+
 |  Memory Tests                                |

 +-----+-------------------------------------------------------------+
 | '0' | Test first 2MB region of DDR              |

 | '1' | Test first 32MB region of DDR             |

 | '2' | Test first 64MB region of DDR             |

 | '3' | Test first 128MB region of DDR            |

 | '4' | Test first 256MB region of DDR            |

 | '5' | Test first 512MB region of DDR            |

 | '6' | Test first 1GB region of DDR              |

 | '7' | Test first 2GB region of DDR              |

 | '8' | Test first 4GB region of DDR              |

 | '9' | Test first 8GB region of DDR              |

 +-----+-------------------------------------------------------------+
 |  Eye Tests                                   |

 +-----+-------------------------------------------------------------+
 | 'r' | Perform a read eye analysis test          |
 | 'w' | Perform a write eye analysis test         |
 | 'a' | Print test start address                  |
 | 't' | Specify test start address (default=0x0)  |
 | 's' | Select the DRAM rank (default=1)          |
 +-----+-------------------------------------------------------------+
 |  Miscellaneous options                       |
 +-----+-------------------------------------------------------------+
 | 'i' | Print DDR information                     |
 | 'v' | Verbose Mode ON/OFF                       |
 | 'o' | Toggle cache enable/disable               |
 | 'b' | Toggle between 32/64-bit bus widths       |
 | 'h' | Print this help menu                      |
 +-----+-------------------------------------------------------------+
  Bus Width = 32,   D-cache is enable,   Verbose Mode is OFF

DDR ECC is DISABLED
Enter 'h' to print help menu
Enter Test Option:


  Bus Width = 32,   D-cache is enable,   Verbose Mode is OFF

DDR ECC is DISABLED
Enter 'h' to print help menu
Enter Test Option:
```

Figure 9 – ZynqMP DRAM Diagnostics Test Menu

```
Starting Memory Test '1' - Testing 32MB length from address 0x0...

---------+--------+------------------------------------------------+-----------
 TEST   | ERROR |        PER-BYTE-LANE ERROR COUNT          | TIME
        | COUNT  | LANES [ #0,  #1,  #2,  #3,  #4,  #5,  #6,  #7] |  (sec)
---------+--------+------------------------------------------------+-----------

MT0(1: 0)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.670433

---------+--------+------------------------------------------------+-----------

MTS(1: 1)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.33751

---------+--------+------------------------------------------------+-----------

MTS(1: 2)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTS(1: 3)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTS(1: 4)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTS(1: 5)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTS(1: 6)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTS(1: 7)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTS(1: 8)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.338166

---------+--------+------------------------------------------------+-----------

MTP(1: 9)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.540672

---------+--------+------------------------------------------------+-----------

MTP(1:10)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.540672

---------+--------+------------------------------------------------+-----------

MTL(1:11)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.582615

---------+--------+------------------------------------------------+-----------

MTL(1:12)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.582615

---------+--------+------------------------------------------------+-----------

MTL(1:13)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.582615

---------+--------+------------------------------------------------+-----------

MTL(1:14)|     0 |   0,   0,   0,   0,   0,   0,   0,    0 | 0.582615

---------+--------+------------------------------------------------+-----------
```

Figure 10 – Test #1, 32MB test

```
Running Read Eye Tests
Disabling VT compensation... DONE
--------------------------------------------------------------------------------
Offset  [LANE-0] [LANE-1] [LANE-2] [LANE-3]
--------------------------------------------------------------------------------
1  |   0|   0|   0|   0|
2  |   0|   0|   0|   0|
3  |   0|   0|   0|   0|
4  |   0|   0|   0|   0|
5  |   0|   0|   0|   0|
6  |   0|   0|   0|   0|
7  |   0|   0|   0|   0|
8  |   0|   0|   0|   0|
9  |   0|   0|   0|   0|
10  |   0|   0|   0|   0|
11  |   0|   0|   0|   0|
12  |   0|   0|   0|   0|
13  |   0|   0|   0|   0|
14  |   0|   0|   0|   0|
15  |   0|   0|   0|   0|
16  |   0|   0|   0|   0|
17  |   0|   0|   0|   0|
18  |   0|   0|   0|   0|
19  |   0|   0|   0|   0|
20  |   0|   0|   0|   0|
21  |   0|   0|   0|   0|
22  |   0|   0|   0|   0|
23  |   0|   0|   0|   0|
24  |   0|   0|   0|   0|
25  |   0|   0|   0|   0|
26  |   0|   0|   0|   0|
27  |   0|   0|   0|   0|
28  |   0|   0|   0|   0|
29  |   0|   0|   0|   0|
30  |   0|   0|   0|   0|
31  |   0|   0|   0|   0|
32  |   0|   0|   0|   0|
33  |   0|   0|   0|   0|
34  |   0|   0|   0|   0|
35  |   0|   0|   0|   0|
36  |   0|   0|   0|   0|
37  |   0|   0|   0|   0|
38  |   0|   0|   0|   0|
39  |   0|   0|   0|   0|
40  |   0|   0|   0|   0|
41  |   0|   0|   0|   0|
42  |   0|   0|   0|   0|
43  |   0|   0|   0|   0|
44  |   0|   0|   0|   0|
45  |   0|   0|   0|   0|
46  |   0|   0|   0|   0|
47  |   0|   0|   0|   0|
48  |   0|   0|   0|   0|
49  |   0|   0|   0|   0|
50  |   0|   0|   0|   0|
51  |   0|   0|   0|   0|
52  |   0|   0|   0|   0|
53  |   0|   0|   0|   0|
54  |   0|   0|   0|   0|
55  |   0|   0|   0|   0|
56  |   0|   0|   0|   0|
57  |   0|   0|   0|   0|
58  |   0|   0|   0|   0|
59  |   0|   0|   0|   0|
60  |   0|   0|   0|   0|
61  |   0|   0|   0|   0|
62  |   0|   0|   0|   0|
63  |   0|   0|   0|   0|
64  |   0|   0|   0|   0|
65  |   0|   0|   0|   0|
66  |   0|   0|   0|   0|
67  |   0|   0|   0|   0|
68  |   0|   0|   0|   0|
69  |   0|   0|   0|   0|
70  |   0|   0|   0|   0|
71  |   0|   0|   0|   0|
72  |   0|   0|   0|   0|
73  |   0|   0|   0|   0|
74  |   0|   0|   0|   0|
75  |  46|   0|   0|   1|
76  | 1642|  15|   2|  76
-1  |   0|   0|   0|   0|
-2  |   0|   0|   0|   0|
-3  |   0|   0|   0|   0|
-4  |   0|   0|   0|   0|
-5  |   0|   0|   0|   0|
-6  |   0|   0|   0|   0|
-7  |   0|   0|   0|   0|
-8  |   0|   0|   0|   0|
-9  |   0|   0|   0|   0|
-10  |   0|   0|   0|   0|
-11  |   0|   0|   0|   0|
-12  |   0|   0|   0|   0|
-13  |   0|   0|   0|   0|
-14  |   0|   0|   0|   0|
-15  |   0|   0|   0|   0|
-16  |   0|   0|   0|   0|
-17  |   0|   0|   0|   0|
-18  |   0|   0|   0|   0|
-19  |   0|   0|   0|   0|
-20  |   0|   0|   0|   0|
-21  |   0|   0|   0|   0|
-22  |   0|   0|   0|   0|
-23  |   0|   0|   0|   0|
-24  |   0|   0|   0|   0|
-25  |   0|   0|   0|   0|
-26  |   0|   0|   0|   0|
-27  |   0|   0|   0|   0|
-28  |   0|   0|   0|   0|
-29  |   0|   0|   0|   0|
-30  |   0|   0|   0|   0|
-31  |   0|   0|   0|   0|
-32  |   0|   0|   0|   0|
-33  |   0|   0|   0|   0|
-34  |   0|   0|   0|   0|
-35  |   0|   0|   0|   0|
-36  |   0|   0|   0|   0|
-37  |   0|   0|   0|   0|
-38  |   0|   0|   0|   0|
-39  |   0|   0|   0|   0|
-40  |   0|   0|   0|   0|
-41  |   0|   0|   0|   0|
-42  |   0|   0|   0|   0|
-43  |   0|   0|   0|   0|
-44  |   0|   0|   0|   0|
-45  |   0|   0|   0|   0|
-46  |   0|   0|   0|   0|
-47  |   0|   0|   0|   0|
-48  |   0|   0|   0|   0|
-49  |   0|   0|   0|   0|
-50  |   0|   0|   0|   0|
-51  |   0|   0|   0|   0|
-52  |   0|   0|   0|   0|
-53  |   0|   0|   0|   0|
-54  |   0|   0|   0|   0|
-55  |   0|   0|   0|   0|
-56  |   0|   0|   0|   0|
-57  |   0|   0|   0|   0|
-58  |   0|   0|   0|   0|
-59  |   0|   0|   0|   0|
-60  |   0|   0|   0|   0|
-61  |   0|   0|   0|   0|
-62  |   0|   0|   0|   0|
-63  |   0|   0|   0|   0|
-64  |   0|   0|   0|   0|
-65  |   0|   0|   0|   0|
-66  |   0|   0|   0|   0|
-67  |   0|   0|   0|   0|
-68  |   0|   0|   0|   0|
-69  |   0|   0|   0|   0|
-70  |   0|   0|   0|   0|
-71  |   0|   0|   0|   0|
-72  |   0|   0|   0|  74|
-73  |   0|   0|   0| 3086|
-74  |   0|   0| 103| 7408|
-75  |   0|   0| 7818| 18385|
-76  |   0|  82| 23608| 33386|
-77  |   0| 386| 38604| 40861|
-78  | 1592| 4142| 98122| 64366|
-------+--------+--------+--------+--------+--------+--------+--------+--------+

Read Eye Test Results :
--------------------------------------------------------------------------------
   [LANE-0] [LANE-1] [LANE-2] [LANE-3]
--------------------------------------------------------------------------------
AUTO CENTER:
--------+---------+---------+---------+---------+---------+---------+---------
 83,86 |  79,81 |  77,80 |  79,81 |
--------+---------+---------+---------+---------+---------+---------+---------
TAP value (ps):
--------+---------+---------+---------+---------+---------+---------+---------

 5.58 |  5.58 |  5.58 |  5.55 |
--------+---------+---------+---------+---------+---------+---------+---------
TAPS/cycle:
--------+---------+---------+---------+---------+---------+---------+---------
  168  |  168  |  168  |  168  |
--------+---------+---------+---------+---------+---------+---------+---------
EYE WIDTH (ps):
--------+---------+---------+---------+---------+---------+---------+---------
842.63 | 837.05 | 825.89 | 804.36 |
--------+---------+---------+---------+---------+---------+---------+---------
EYE WIDTH (%):
--------+---------+---------+---------+---------+---------+---------+---------
89.88  | 89.29  | 88.10  | 86.31  |
--------+---------+---------+---------+---------+---------+---------+---------
EYE CENTER:
--------+---------+---------+---------+---------+---------+---------+---------
 81,84 |  79,81 |  78,81 |  80,82 |
--------+---------+---------+---------+---------+---------+---------+---------
```

Figure 11 – Measure Read Data Eye

Experiment 3: Edit Memory Test to Expand the Range

Ultra96 contains 2 GB of LPDDR4 RAM. You may have noticed that the Memory Test application actually runs three different memory tests – 32-bit, 16-bit, and 8-bit. These tests completed very quickly, which should be an indication that the entire memory range was not tested.

13. Open the system.hdf in the hw_platform_0 to investigate the memory map for the DDR.

Cell	Base Addr	High Addr	Slave I/f	Mem/Reg	Segment	TrustZone	AccessType
psu_ddr_0	0x00000000	0x7fefffff		MEMORY	MEM_0	NonSecure	Read/Write

Figure 12 – LPDDR4 Memory Map

Notice that the address range is 0x00000000 to 0x7fefffff, which is 0x7FF00000 or 2,146,435,072 bytes. (For an explanation on where the highest 1 MB of LPDDR4 went, see the ZU+ MP TRM chapter on the PMU.

14. Browse to the C source code for the Memory Test application in the *Project Explorer* at Mem_Test → src

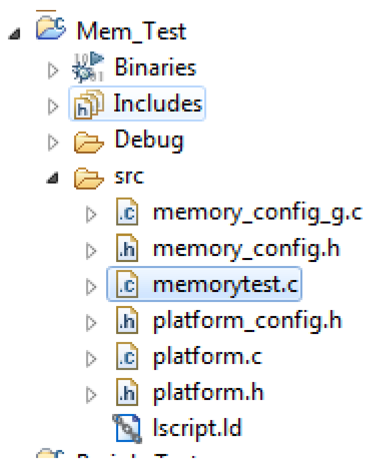

Figure 13 – Memory Test C Sources

15. The main() function is located in memorytest.c. Open that source by double-clicking it.

In main(), a for loop iterates on a variable n_memory_ranges to run function test_memory_range. The n_memory_ranges allows this application to test multiple memory stores, if they exist. However, for Ultra96, the LPDDR4 is the only valid memory store that can be tested.

Looking up further in the file, you will notice the test_memory_range() function.

16. Find lines 79, 82, and 85. You will see that the default function only tests the first 4K bytes:
    - 1024 locations in the 32-bit (4-bytes) test
    - 2048 locations in the 16-bit (2-bytes) test
    - 4096 locations in the 8-bit (1 byte) test

We will change this to test the accessible LPDDR4. The original Mem_Test provided a clue as to what value to use for the maximum range:

```
Base Address: 0x50

Size: 0x7FEFFFB0 bytes
```

17. Return to memorytest.c. Make the following edits:
    - Line 79: replace 1024 with 0x7FEFFFB0/4
    - Line 82: replace 2048 with 0x7FEFFFB0/2
    - Line 85: replace 4096 with 0x7FEFFFB0

```
79      status = Xil_TestMem32((u32*)range->base, 0x7FEFFFB0/4, 0xAAAA5555, XIL_TESTMEM_ALLMEMTESTS);
80      print("          32-bit test: "); print(status == XST_SUCCESS? "PASSED!":"FAILED!"); print("\n\r");
81
82      status = Xil_TestMem16((u16*)range->base, 0x7FEFFFB0/2, 0xAA55, XIL_TESTMEM_ALLMEMTESTS);
83      print("          16-bit test: "); print(status == XST_SUCCESS? "PASSED!":"FAILED!"); print("\n\r");
84
85      status = Xil_TestMem8((u8*)range->base, 0x7FEFFFB0, 0xA5, XIL_TESTMEM_ALLMEMTESTS);
86      print("           8-bit test: "); print(status == XST_SUCCESS? "PASSED!":"FAILED!"); print("\n\r");
```

Figure 14 – Modified Memory Test

18. Save the file, which will cause a re-build.
19. Re-run this edited and newly built Mem_Test. Be patient as the test times are longer.
    - 32-bit test: ~0:09
    - 16-bit test: ~0:18
    - 8-bit test: ~0:36

Total elapsed time will be about 1 minute.

Revision History

Date	Version	Revision
02 Jul 2018	2018_2.01	Initial Avnet release for Vivado 2018.2