ZYNQMP DRAM DIAGNOSTICS TEST



Updated 15 July 2016

# Introduction

The Zynq MP DRAM diagnostics test is a stand-alone program running on a single Zynq MPSoC Cortex-A53 processor, executing out of OCM. The program uses the UART for interactive operation. A small menu is displayed, and the user may choose to run various memory tests.

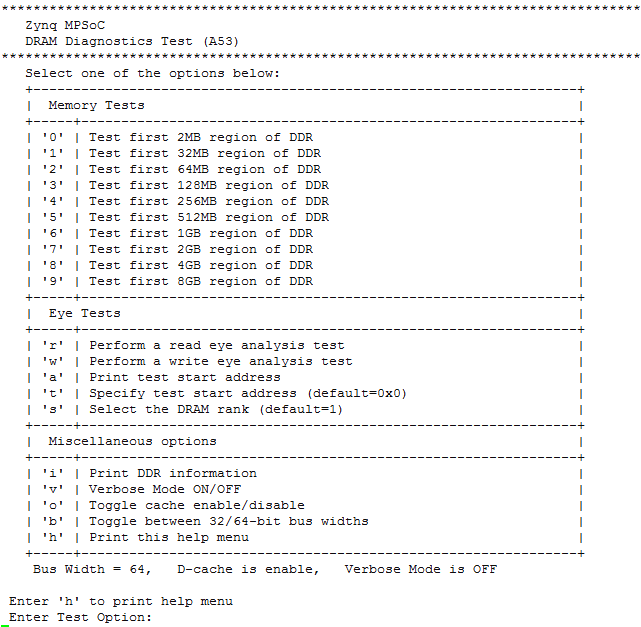
While running the test from SDK, these operations are performed by SDK, during the debug launch.

Do not do DDR remap or RAM remap.

# Running the Test

After connecting a board and launching the test from SDK, in the terminal window you should see the test menu as shown in the screen shot below.

To select a function, hit a single key without <enter>. Information is printed as the test progresses, and the menu is re-printed when the function completes.

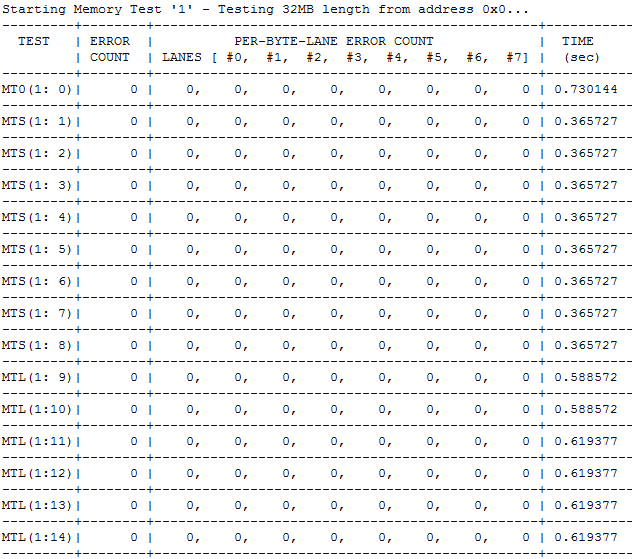


## Memory Tests

The keys 0,1,2,3,4,5,6,7,8,9 select a memory test of varying lengths, as shown below.

|  |  |
| --- | --- |
| Key | Test Length |
| ‘0’ | 2MB |
| ‘1’ | 32MB |
| ‘2’ | 64MB |
| ‘3’ | 128MB |
| ‘4’ | 256MB |
| ‘5’ | 512MB |
| ‘6’ | 1GB |
| ‘7’ | 2GB |
| ‘8’ | 4GB |
| ‘9’ | 8GB |

Below is a screen shot after hitting key ‘1’.



Each memory test consists of 15 sub-tests using different data patterns. In each sub-test, the entire range is first written sequentially, and then read and compared against the expected value. The 15 patterns are:

|  |  |
| --- | --- |
| Sub-test | Description |
| 0 | Incrementing pattern, unique value per memory location (data = address) |
| 1 | All 0 |
| 2 | All 0xffffffff |
| 3 | All 0xAAAAAAAA |
| 4 | All 0x55555555 |
| 5 | Alternating 0x00000000 and 0xFFFFFFFF |
| 6 | Alternating 0xFFFFFFFF and 0x00000000 |
| 7 | Alternating 0x55555555 and 0xAAAAAAAA |
| 8 | Alternating 0xAAAAAAAA and 0x55555555 |
| 9 | Aggressor pattern identical on all 8 bits |
| 10 | Aggressor pattern with one bit inverted, x8 times (1 per bit) |
| 11-14 | Pseudo random patterns with different seeds |

A word error count and per-byte-lane error counts are provided in the format (Lane-0 Lane-1 Lane-2 Lane-3, Lane-4, Lane-5, Lane-6, Lane-7).

**Word Error Count:** No. of words having errors on read back and comparison with written value.  
**Per-Byte-Lane Error Counts:** No. of bytes having errors on read back and comparison with written value.

*Example:*  
Data Written : 0x01010101, Data Read Back: 0x01000001

The per-byte-lane error count reads errors on Lane-1 & Lane-2

The word error count is 1

Errors, if any, are reported for each sub-test. In verbose mode (hit the ‘v’ key), the first 10 errors in each sub-test are printed, for example:

Memtest\_l ERROR: addr=0x107154 rd/ref/xor = 0x478FDCF5 0x478FDDF5 0x00000100

Memtest\_l ERROR: addr=0x10E78C rd/ref/xor = 0x34BBA068 0x34BBA078 0x00000010

Memtest\_l ERROR: addr=0x10F00C rd/ref/xor = 0xFEF17729 0xFEF1F729 0x00008000

Memtest\_l ERROR: addr=0x10F024 rd/ref/xor = 0xBC7DCA6B 0xBC7DCA7B 0x00000010

Memtest\_l ERROR: addr=0x11304C rd/ref/xor = 0x87D46558 0x87D46758 0x00000200

Memtest\_l ERROR: addr=0x115154 rd/ref/xor = 0xD894F5B9 0xD894F5BD 0x00000004

Memtest\_l ERROR: addr=0x12F00C rd/ref/xor = 0xEFE50484 0xEFE58484 0x00008000

Memtest\_l ERROR: addr=0x12F154 rd/ref/xor = 0xA4CBAACE 0xA4CBABCE 0x00000100

Memtest\_l ERROR: addr=0x13D20C rd/ref/xor = 0xC7DF9980 0xC7DF99C0 0x00000040

Memtest\_l ERROR: addr=0x13D5EC rd/ref/xor = 0xF82DDD6F 0xF82DDF6F 0x00000200

Memtest\_l ( 0:14) Done 1 MB starting at 1 MB, 67 errors (20 40 1 6). 0.105578 sec

## Eye Tests

Prior to running the test, users must:

* Set APU frequency at >1GHz
* Initialize the DDR PLL
* Initialize the DDR controller (using FSBL or zDDR TCL scripts)

If this test is being run from SDK, these operations will be performance by SDK before launching the program.

The test runs out of OCM with the processor caches and MMU enabled. The test is agnostic to memory type (DDR3, DDR4, LPDDR4) or to the memory bus width (32 or 64-bits). This test does not measure the eye width for ECC lane. However, to ensure smooth operation of the program, user must write the entire DRAM address space to ensure valid ECC codes before running the test. The program does not use entire DRAM but this is a good practice nonetheless.

A fixed pattern that is pre-computed and is known to cause maximum errors on the data lanes as the eye width shrinks is run. The memory test is run using a single A53 processor running at speed of 1.0GHz+ with caches enabled. The memory test writes to a 1MB block of memory and as per the default behavior of the system, the block will be read in the APU’s L2 cache and it will be written there. Once the 1MB buffer is written completely, L2 cache flush routine is invocated to ensure the entire block of memory is written back to DRAM at full speed.

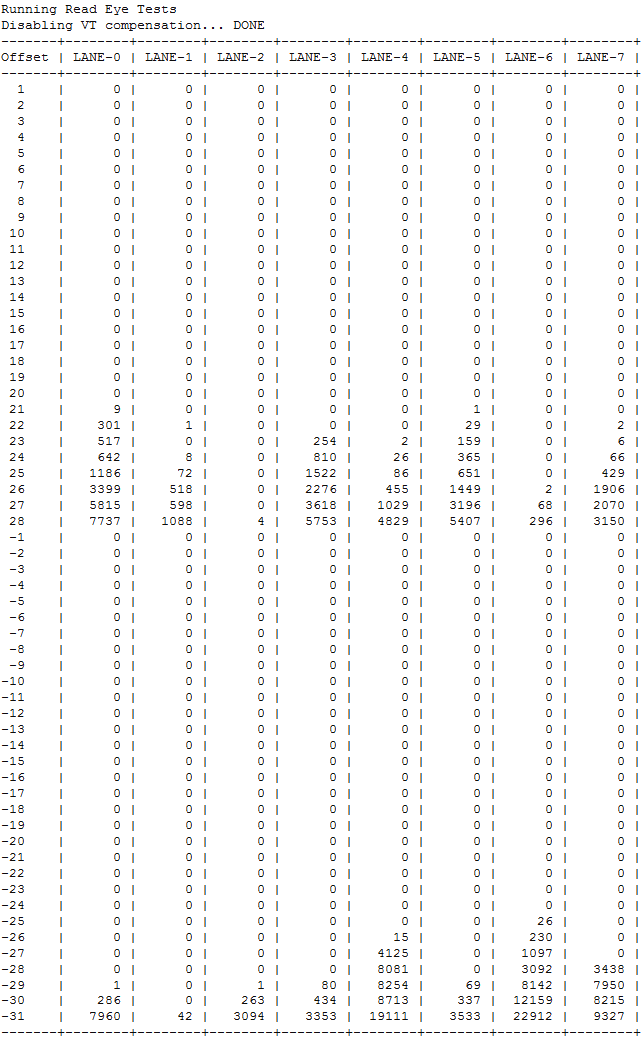
The test then reads back the contents of the memory that was written to using a known pattern and calculated any errors on a per-byte lane basis. This number of errors is display after the test is complete and is used to judge the edge of the eye per lane.

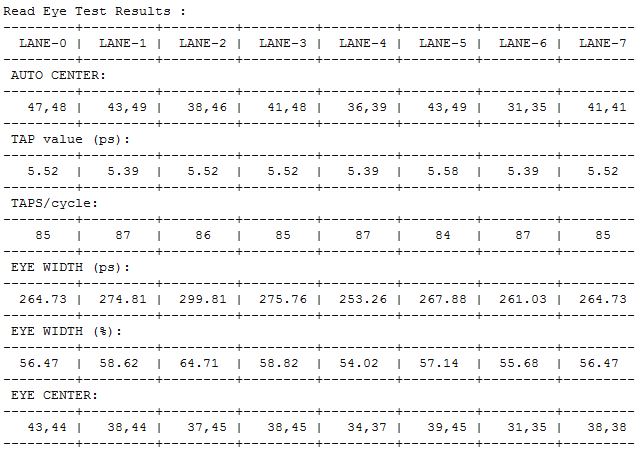
### Read Eye Measurement

This can be invoked by pressing ‘r’ key on the menu. The test will read the eye center as computed after auto training and use that as the baseline to compute the new center by moving the DQS position by one tap count at a time and running the memory test pattern. The resulting output is as displayed in the following picture.

As seen in the displayed output, the test runs for multiple tap counts in each direction to find the edge of the eye per byte lane. Once all lanes have failed the memory test in both directions, the final output is shown (per byte lane) with some key information to judge the width and center of the eye.

After the final results are displayed, the software will restore the original state of any registers. VT compensation is disabled during the test and re-enabled once completed.



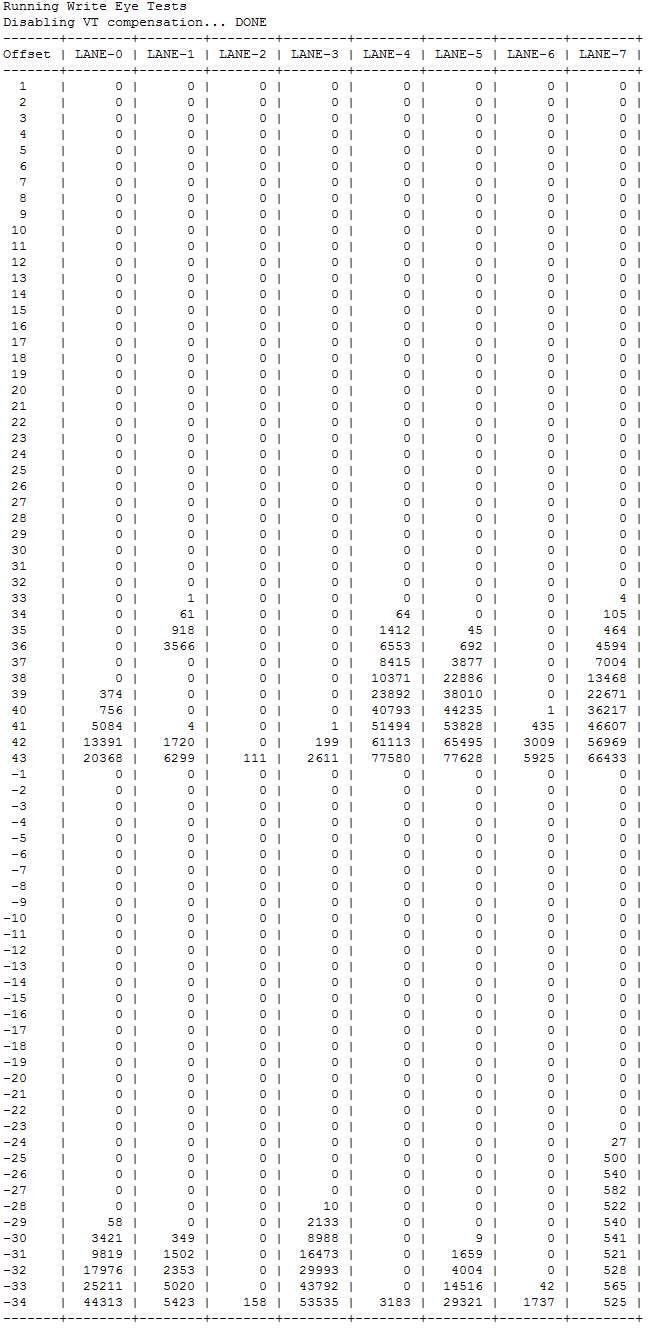


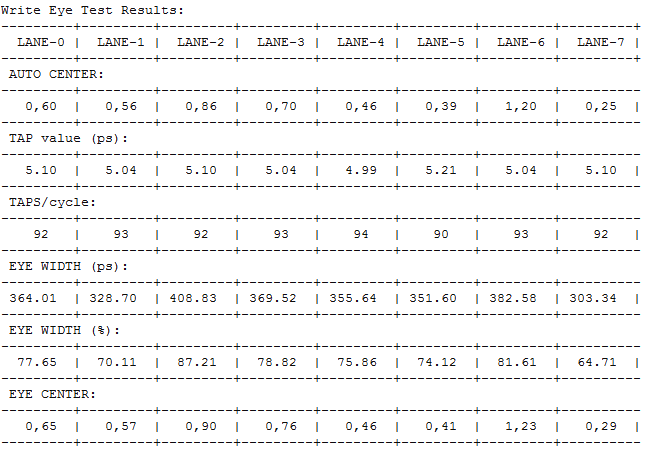
### Write Eye Measurement

This test can be invoked by pressing ‘w’ key on the menu. The test then reads the eye center as computed after auto training and use that as the baseline to compute the new center by moving the DQS position by one tap count at a time and running the memory test pattern. The resulting output is as displayed in the picture above.

Similar to the read eye measurement test, the write eye measurement also runs for multiple tap counts in each direction to find the edge of the eye per byte lane. Once all lanes have failed the memory test in both directions, the final output is shown (per byte lane) with some key information to judge the width and center of the eye.

After the final results are displayed, the software will restore the original state of any registers. VT compensation is disabled during the test and re-enabled once completed.





## Other Menu Items

Other menu items are described below.

|  |  |  |
| --- | --- | --- |
| Key | Name | Description |
| ‘b’ | Bus Width | Toggle between 32/64-bit bus widths |
| ‘v’ | Verbose | Toggle verbose mode on/off. If on and errors occur during a memory test, the first 10 errors in each sub-test are printed. |
| ‘o’ | D-Cache Enable/Disable | Toggle D-cache – enable/disable. |