ZYNQ DRAM DIAGNOSTICS TEST



Updated 19 February 2015

# Introduction

The Zynq DRAM diagnostics test is a stand-alone program running on a single Zynq Cortex-A9 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 functions, including:

* Memory test
* Read eye measurement
* Write eye measurement

Prior to running the test, users must do the following:

* Initialize the DDR PLL
* Initialize the DDR Controller + ddriob (e.g. zddr initialization)

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

The test uses the default OCM memory mapping (192KB in low memory, 64KB in high memory), and runs with L1+L2 caches and MMU enabled. The test is agnostic to memory type (DDR2, DDR3, LPDDR2), or to memory bus width (16 or 32 bits). However, in 16-bit+ECC mode, the entire DRAM must be written at least once to ensure valid ECC codes before running this test. The test sets the CPU PLL for 667MHz operation.

Do not do DDR remap or RAM remap.

# Running the Test

After connecting a board and launching the test from SDK, in the com0 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 Test

The keys s,1,2,3,4,5,6 select a memory test of varying lengths, as shown below. Note that the first 1MB of the DRAM is not accessible in this memory mapping mode.

|  |  |  |
| --- | --- | --- |
| Key | Test Start Address | Test Length |
| ‘s’ (“short”) | 0x100000 | 1MB |
| ‘1’ | 0x100000 | 32MB |
| ‘2’ | 0x100000 | 64MB |
| ‘3’ | 0x100000 | 128MB |
| ‘4’ | 0x100000 | 255MB |
| ‘5’ | 0x100000 | 511MB |
| ‘6’ | 0x100000 | 1023MB |

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



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).

**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

## Read Data Eye Measurement

To measure the read data eye hit the ‘r’ key. A screen shot is provided below.



The read data eye width (per byte lane) is measured by shifting the read DQS position relative to the read data eye in both forward and backward directions, and finding the range in which a memory test is successful. The DQS offset is measured in 1/256 clock cycle units (256 units = 1 clock cycle). The default relative position of read DQS relative to the read data eye is a ¼ cycle shift (64 units). The test scans the DQS offset value from 64 upwards until a memory test fails on all lanes, and then from 60 downwards until a memory test fails on all lanes. Upon completion, the width and offset of the read data eye per byte lane is displayed. The state of all registers is then restored.

## Write Data Eye Measurement

To measure the write data eye hit the ‘i’ key. A screen shot is provided below.



The write data eye width (per byte lane) is measured by shifting the write DQS position relative to the write data eye, and finding the range in which a memory test is successful. The DQS offset is measured in 1/256 clock cycle units (256 units = 1 clock cycle). The default relative position of write DQS relative to the write data eye is a ¼ cycle shift (64 units). The test scans the DQS offset value from a min to a max value (default 16-112) and marks the range in which a memory test is successful. Upon completion, the width and offset of the write data eye per byte lane is displayed.

### Adjusting the Write Eye

By default, the write eye measurement restores the original state or all registers. To adjust the write eye for optimum position based on the write eye measurement, hit the ‘c’ key prior to running the write eye measurement. Upon test completion, the optimal write eye position will be set. Note that Zynq does not support write data eye training, so the default or manual settings for the write eye may not be optimal, unlike read data eye training, which is supported where available (DDR3, LPDDR2).

## Other Menu Items

Other menu items are described below.

|  |  |  |
| --- | --- | --- |
| Key | Name | Description |
| ‘f’ | Fast | Toggle ‘fast’ mode on/off. In fast mode, the memory test used during eye measurements uses less sub-tests and therefore runs about twice as fast, at the cost of being slightly more optimistic. By default ‘fast’ is **on**. |
| ‘c’ | Center | Toggle ‘center’ mode on/off. When enabled, the write eye measurement result is immediately programmed into the DDR controller. |
| ‘e’ | Eye test size | Vary the size of the memory test used at each step of a read/write eye measurement functions. The default value is 1MB, resulting in fastest speed at the cost of producing slightly optimistic results. Hitting this key repeatedly varies the test size circularly between the values 1,2,4,8,16,32MB. Note that using the value 4 will quadruple the test run time. |
| ‘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. |
| ‘z’ | D-Cache Enable/Disable | Toggle D-cache – enable/disable. |

## Other Limitations

This test does not support ECC mode, and the read/write eye tests must always be run with ECC mode disabled.

As for training purposes it does not really matter whether a particular byte lane is used for data or ECC syndrome, so running the eye test in 32-bit mode with ECC disabled is a viable option to train all the four lanes.

## Example Outputs

Typical Xilinx demonstrated Zynq boards Read/Write Eye diagrams

Data is read as [Eye Width, Offset from ideal eye center (64)]



Example: Lane-0 [80, -4] shows   
 Eye width = 8  
 Eye center = 60 (Offset from ideal center 64 is -4)

Another example timing diagram for the figures calculated for a Xilinx Demonstrated board is.

**--------------------------------------------------------**

**Description LANE-0 LANE-1 LANE-2 LANE-3**

**--------------------------------------------------------**

**EYE [MIN-MAX] : [16, 96] [16,100] [16,100] [12, 96]**

**EYE CENTER : 56/128 58/128 58/128 54/128**

**EYE WIDTH : 62.50% 65.62% 65.62% 65.62%**

****