DesignWare Cores LPDDR54 PHY Diagnostic Firmware Application Note

DWC LPDDR54 PHY
Firmware Version: C-2020.11

Firmware version: C-2020.11
Document Version 1.40a
December 7, 2020

Copyright Notice and Proprietary Information Notice

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys, Inc.
690 E. Middlefield Road
Mountain View, CA 94043

www.synopsys.com

1 Contents

2 Revision History

Document Revision	Date	Description
1.40a	December 7, 2020	Updated: Adding Figure 1 Initialization Flow. Updating to C-2020.11
1.30	September 4, 2020	Updated: 5.10 Test 10 : Mode Register Read Updating to A/B-2020-09
1.20	June 10, 2020	Updated
1.10	April 30, 2020	Initial release

3 Reference Documents

The following reference documents are used in this application note:

DesignWare Cores LPDDR54 SDRAM PHY Databook, Synopsys, Inc.

DesignWare Cores LPDDR54 Utility Block (PUB) Release Notes, Synopsys, Inc.

4 Introduction

Note	This application note pertains to A-2019.01 version of the DWC DDRPHY Diagnostic firmware. There may be variations in operation, interface, and usage procedure, depending on the DWC DDRPHY Diagnostic firmware version

4.1 Purpose

The DWC DDRPHY Diagnostic Firmware is a software tool to obtain diagnostic information from a trained DDR system.

This is delivered as firmware image which contains a number of tests that may be used to troubleshoot or measure the PHY-DRAM link.

In addition to the descriptions of the individual tests, the process for loading and running the firmware, including the prerequisite steps are described.

Since the nature of many of the diagnostic tests involve placing the memory system beyond its typical operating limits, the system is not supported to run mission mode traffic after the diagnostics complete. After performing the diagnostics, the system is required to be reset/reinitialize or powered off.

Note	The DWC DDRPHY Diagnostic Firmware is a utility intended for diagnostic debug purposes. The DWC DDRPHY Diagnostic Firmware is not intended for production use

4.2 Overview and Pre-Requisites

The PHY Diagnostic firmware may be executed immediately following PHY training.

In the case of a system with multiple trained pstates – it is up to the user to appropriately choose when to run the diagnostics.

Assuming the example Training + mission mode sequence below:

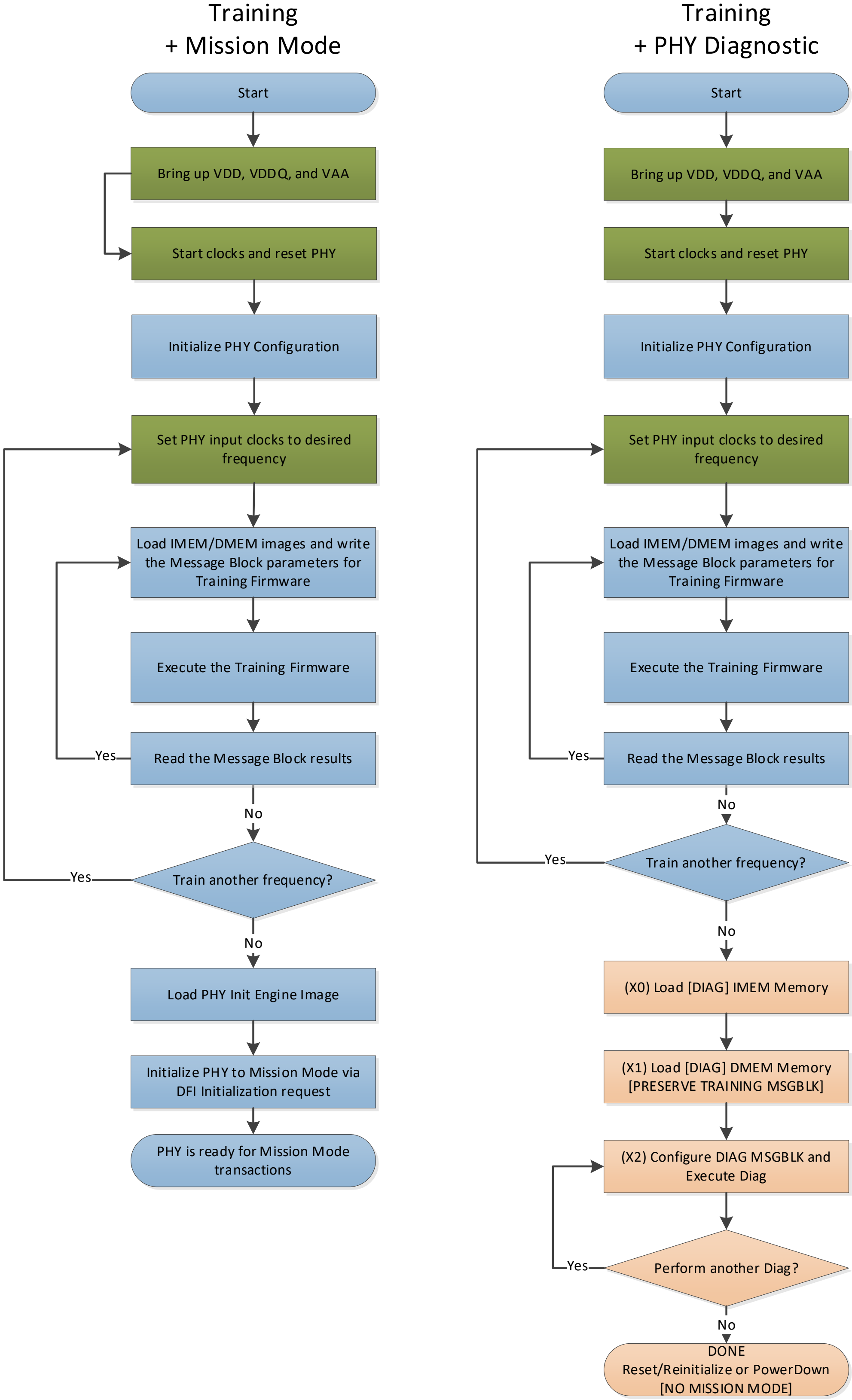
a. Train -P0 @3200
b. DFI Initialization @ P0

The sequence may be interrupted after completion of any of the training steps (a,b,c); and proceed to diagnostics instead of continuing with the complete sequence. The following (Figure 1) shows the differences between the standard training + mission mode flow and the PHY Diagnostic flow.

For the Diagnostic Firmware sequence to execute, the PHY must be in the following state:

- Previous Training step executed as expected and returned Message 0x7 [Firmware complete]
- The complete message block from the previous training step is preserved in the PHY DMEM.
- The DFICLK is stable and operating at the same frequency as the previous training step.
- The PHY Register MicroReset = 0x1
- The PHY Register MicroContMuxSel = 0x0
- The PHY Register UcclkHclkEnables= 0x3

Figure 1: Initialization flow


Training
+ Mission Mode
Start
Bring up VDD, VDDQ, and VAA
Start clocks and reset PHY
Initialize PHY Configuration
Set PHY input clocks to desired frequency
Load IMEM/DMEM images and write the Message Block parameters for Training Firmware
Execute the Training Firmware
Read the Message Block results
Yes
No
Train another frequency?
Yes
No
Load PHY Init Engine Image
Initialize PHY to Mission Mode via DFI Initialization request
PHY is ready for Mission Mode transactions
Training
+ PHY Diagnostic
Start
Bring up VDD, VDDQ, and VAA
Start clocks and reset PHY
Initialize PHY Configuration
Set PHY input clocks to desired frequency
Load IMEM/DMEM images and write the Message Block parameters for Training Firmware
Execute the Training Firmware
Read the Message Block results
Yes
No
Train another frequency?
Yes
No
(X0) Load [DIAG] IMEM Memory
(X1) Load [DIAG] DMEM Memory [PRESERVE TRAINING MSGBLK]
(X2) Configure DIAG MSGBLK and Execute Diag
Perform another Diag?
Yes
No
DONE
Reset/Reinitialize or PowerDown
[NO MISSION MODE]

The Diagnostic Firmware does not alter the mode register state (except where required by the specific test-type), and relies on the DRAM state to be initialized by the preceding training steps.

In LPDDR4 systems the following dram mode register settings must be constrained if Diagnostics Firmware will be run (mode registers not listed may be set as desired). Individual tests may have additional limitations, refer to the descriptions for each test for further details.

LPDDR4 Mode Register #	Constrained Values	Comments
MR13	OP[3] = {1}	VRCG must be in high current mode for Diags
MR16	OP[7:0] = {0,0,0,0,0,0,0,0}	PASR Mode not supported by Diags
MR17	OP[7:0] = {0,0,0,0,0,0,0,0}	PASR Mode not supported by Diags
MR24	OP[7:4] = {0,0,0,0}	TRR Mode not supported by Diags

In DDR4 systems the following dram mode register settings must be constrained if Diagnostics Firmware will be run (mode registers not listed may be set as desired). Individual tests may have additional limitations, refer to the descriptions for each test for further details.

DDR4 Mode Register #	Constrained Values	Comments
MR0	A[1:0] = {0,1} or {0,0}	Fixed BC4 not supported by Diags
MR0	A[7] = {0}	TM not supported by Diags
MR1	A[0] = {1}	DLL-off mode not supported by Diags
MR1	A[12] = {0}	DRAM output buffers must be enabled
MR2	A[12] = {0}	WCRC not supported by Diags
MR2	A[13,8,2,1,0] = {0,0,0,0,0}	TRR Mode not supported by Diags
MR4	A[1] = {0}	MPSM Mode not supported by Diags
MR4	A[8,7,6] = {0,0,0}	CS to CMD Latency not supported by Diags
MR4	A[13,5] = {0,0}	PPR mode not supported by Diags
MR5	A[9,4,3,2,1,0] = {0,0,0,0,0,0}	DRAM CMD Parity not supported by Diags

In DDR4 RDIMM systems, the following Register control word settings must be constrained. These constraints are in addition to the DDR4 dram restrictions listed above.

RCD Control Word #	Constrained Values	Comments
F0RC00	DA[3,2,0] = {0,0,0}	Only normal RCD operation mode supported
F0RC02	DA[3,2] = {0,0}	Only normal RCD operation mode supported
F0RC06	CMD{6,7,8,10} not allowed	RCD Parity and RCD geardown not supported
F0RC09	DA[3] = {1}	Clock Stop Power Down Mode is required
F0RC0A	DA[2:0] != {1,1,1}	RCD PLL Bypass mode not supported
F0RC0A	DA[3] = {0}	Only RCD Context 1 supported
F0RC0C	DA[3:0] = {0,0,0,0}	Only normal RCD operation mode supported
F0RC0D	DA[3] = {1}	Only normal RCD operation mode supported
F0RC0E	DA[0] = {0}	RCD Parity not supported

4.3 Load Diag IMEM and DMEM (Step X0, X1)

Loading the firmware is accomplished by accessing the PMU SRAMs over the APB bus or JTAG. The SRAMs are mapped into the APB address space starting at address 0x50000 for the instruction memory and 0x58000 for the data memory, out of which the initial 1KB is reserved for training firmware message block. The diagnostic firmware message block starts at 0x58200.

The Diagnostic firmware requires up to 32KB of instruction memory and up to 15KB of data memory. The firmware is provided in two formats:

- Binary image (.bin files)
- Text files with APB address/data pairs (.incv files)

The two formats contain identical firmware images, and the user can use whichever format is more convenient. To use the APB address / data pairs file, write each address with the associated data. To use the binary image file, write the binary contents of the file 16 bits at a time to the memory.
There is a separate file for:

- the instruction memory image (**XXX**_diags_imem.bin or **XXX**_diags_imem.incv)
- the data memory image (**XXX**_diags_dmem.bin or **XXX**_diag_dmem.incv)

Where **XXX** is correct protocol for the trained dram.

Both the instruction memory and the data memory image must be loaded. The instruction memory image contains the executable training code. The data memory image contains data structures that the instruction memory image needs to run.

After running the previous training step, the training messageblock is still resident in the PHY SRAM data memory. If the messageblock is not overwritten by the user, it does not need to be reloaded. The full address regions for the Preserved Training Message block, Diag FW IMEM, and Diag FW DMEM is shown in Table 4-1 PHY SRAM Address Map.

Table 4-1 PHY SRAM Address Map

Image	Start Address	End Address	Data Width per address increment
DIAG FW IMEM	0x50000	0x53FFF (32kB)	16
Preserved Training MsgBlk	0x58000	0x581FF (1kB)	16
DIAG FW DMEM	0x58200	0x55FFF (15kB)	16

Note	After loading with the Binary image (.bin files), it is necessary to reload the original content of trainning's message block content, that is from 0x58000 ~ 0x581ff. There is no need to reload training's message block if using Text files with APB address/data pairs (.incv files).

Note	While loading firmware IMEM/DMEM images, the csrMicroReset must be set so that {Reset = 0; stall =1 }.

4.4 Configure Diag Message Block and Execute (Step X2)

To run the desired test, the Diagnostics message block needs to be configured. The exact settings needed will vary by test. Refer to Section 4.3 for information on the size and location of the diagnostics message block and chapter 5 for details regarding the individual diagnostic tests.

The diagnostics execution process is as follows:

1. Configure the Diagnostic Message block with the desired message block
2. Reset 1st Microcontroller Message Protocol Bit [write PHY Register UctWriteProt = 0x1]
3. Reset 2nd Microcontroller Message Protocol Bit [write PHY Register DctWriteProt = 0x1]
4. Reset Microcontroller Message [write PHY Register UctWriteOnly = 0x0]
5. Set control of the internal CSR bus to the PMU [write PHY Register MicroContMuxSel = 0x1]
6. Assert PMU Reset and Stall [write PHY Register MicroReset = 0x9]
7. De-assert PMU Reset only [write PHY Register MicroReset = 0x1]
8. De-assert PMU Stall [write PHY Register MicroReset = 0x0]

9. Wait for the Diagnostics test to finish
   [Poll PHY Register UctWriteOnlyShadow until a completion message is received]
   See Table 4-2 Mailbox Message Values

10. Assert PMU Stall [write PHY Register MicroReset = 0x1]

11. Set control of the internal CSR bus to the APB [write PHY Register MicroContMuxSel = 0x0]
12. Read back diagnostics return data per test.
    Refer to each test for details of the return data format

If additional tests are desired, repeat the above execution process.

Table 4-2 Mailbox Message Values

UctWriteOnlyShadow Value	Message meaning
0x00	Reset Value
0x07	Diagnostics finished successfully (firmware complete)
0xFF	Abnormal exit (firmware complete)

4.5 Diagnostics Message Block

The diagnostics firmware has a message block area to allow for configuring of the test and returning of the data. There are three major sections of the message block, the 1D configuration block, the Diagnostics Message block and the return data.

The beginning of the message block area is the 1D message block which needs to be preserved from 1D training. The location is the same is used in 1D training. The 1D message block area is preserved so the Diagnostics Firmware knows the current state of the PHY and the DRAM so that it can correctly execute the test.

The diagnostics message block follows the 1D message block. It is located 0x400 bytes past the beginning of the 1D message block at CSR Address 0x58200. It contains the information needed to choose and configure the Diagnostic tests. The names of the arguments are described in the header file format below

Table 4-3 Diagnostics Message Block Definition

```
typedef struct _PMU_SMB_DIAG_t {

   uint8_t  DiagTestNum;      // Byte offset 0x400, CSR Addr 0x58200, Direction=N/A
   uint8_t  DiagSubTest;      // Byte offset 0x401, CSR Addr 0x58200, Direction=N/A
   uint8_t  DiagPrbs;         // Byte offset 0x402, CSR Addr 0x58201, Direction=N/A
   uint8_t  DiagRank;         // Byte offset 0x403, CSR Addr 0x58201, Direction=N/A
   uint8_t  DiagChannel;      // Byte offset 0x404, CSR Addr 0x58202, Direction=N/A
   uint8_t  DiagRepeatCount;  // Byte offset 0x405, CSR Addr 0x58202, Direction=N/A
   uint8_t  DiagLoopCount;    // Byte offset 0x406, CSR Addr 0x58203, Direction=N/A
   uint8_t  DiagByte;         // Byte offset 0x407, CSR Addr 0x58203, Direction=N/A
   uint8_t  DiagLane;         // Byte offset 0x408, CSR Addr 0x58204, Direction=N/A
   uint8_t  DiagVrefInc;      // Byte offset 0x409, CSR Addr 0x58204, Direction=N/A
   uint8_t  DiagReserved0A;   // Byte offset 0x40a, CSR Addr 0x58205, Direction=N/A
   uint8_t  DiagXCount;       // Byte offset 0x40b, CSR Addr 0x58205, Direction=N/A
   uint16_t DiagAddrLow;      // Byte offset 0x40c, CSR Addr 0x58206, Direction=N/A
   uint16_t DiagAddrHigh;     // Byte offset 0x40e, CSR Addr 0x58207, Direction=N/A
   uint16_t DiagPatternLow;   // Byte offset 0x410, CSR Addr 0x58208, Direction=N/A
   uint16_t DiagPatternHigh;  // Byte offset 0x412, CSR Addr 0x58209, Direction=N/A
   uint8_t  DiagMisc0;        // Byte offset 0x414, CSR Addr 0x5820a, Direction=N/A
   uint8_t  DiagMisc1;        // Byte offset 0x415, CSR Addr 0x5820a, Direction=N/A
   uint8_t  DiagMisc2;        // Byte offset 0x416, CSR Addr 0x5820b, Direction=N/A
   uint8_t  DiagReserved17;   // Byte offset 0x417, CSR Addr 0x5820b, Direction=N/A


[…]
   uint8_t  DiagReserved3F;   // Byte offset 0x43F, CSR Addr 0x5821F, Direction=N/A


   uint16_t DiagReturnData;   // Byte offset 0x440, CSR Addr 0x58220, Direction=N/A
                               // This byte offset is the pointer to the first address
                               // of the DiagReturnData ; please see individual tests for
                               // details of the data structure

   } __attribute__ ((packed)) PMU_SMB_DIAG_t;
```

The location of the message block areas are shown in Table 4-3 Message Block Address Map.

Table 4-4 Message Block Address Map

Section	CSR Address	DMEM offset (bytes)
1D Message block	0x58000	0x00
Diagnostics Message block	0x58200	0x400
Data Return Data	0x58220	0x440

The last item in the Diagnostics message block is the DiagReturnData field. It points to the beginning of the return data, the format of which varies between tests. The field is defined as two bytes long in the header, but the return data size is defined by the test that is being run.

4.6 Simulating Diagnostics

When running diagnostics in RTL simulation, memory models may flag certain protocol violations. Refresh requirements, such as tRASmax are not always met and some tests (such as the READ test) may use unititialized data. In these cases those errors can be ignored or waived.

5 Diagnostic Test Details

5.1 Diagnostic Functions Overview

The following table describes the tests that are available for each of the supported DRAM protocols. The Diagnostic Firmware supports multiple protocols, not all Diagnostic routines work with all protocols. Only the diagnostic routines supported by the customer PHY should be used

Test Number	Test	LPDDR4	LPDDR5
1	Reserved		
2	Send Burst Writes	SUPPORTED	SUPPORTED
3	Send Burst Reads	SUPPORTED	SUPPORTED
4	Simple Write Read	SUPPORTED	SUPPORTED
5	Write Eye	SUPPORTED	SUPPORTED
6	Read Eye	SUPPORTED	SUPPORTED
7	Reserved		
8	Reserved		
9	Mode register write	SUPPORTED	SUPPORTED
A	Mode register read	SUPPORTED	SUPPORTED
B	Reserved		
C	Reserved		
D	Reserved		
E	Reserved		
F	Reserved		

5.2 Adjusting Test Run Time

The diagnostic tests allow for the number of bursts per run to be adjusted using the messageblock parameters DiagXCnt, DiagLoopCount, and DiagRepeatCount. Tests that support these looping variables have an internal loop following the below pseudocode.

```
For (x=0; x < diagXCnt; x++){
    Activate bank
    For (loop=0; loop <= (diagLoopCount); loop++){
        For (repeat=0; repeat <= diagRepeatCount; repeat++){
            Write 1 burst
        }//for
For (repeat=0; repeat <= diagRepeatCount; repeat++){
                    Read 1 burst
                }//for
}//for
            Precharge bank
}//for
```

This can be summarized in an equation for the number of bursts sent/received per test run.

BurstCount = DiagXCnt * (1+DiagLoopCount)*(1+DiagRepeatCount).

Tests can offset loop variables, such as making DiagLoopCount specify multiples of 255. When calculating the number of bursts a test will run, it's important to take these offsets into account as part of the loop variables they modify.

5.3 PPGC configuration

All the diagnostics tests have either/both of these two DRAM operations:

- WR command to DRAM to send the pattern specified by DiagPrbs
- RD command to receive back pattern specified by DiagPrbs

To make sure that the PPGC sends correct data on the DiagLane during write operation and uses correct data to compare the received data, the hardware mode to generate DBI lane pattern is used. Each DQ lane and DBI lane carries depending upon diagnostics' message block configuration (DiagLane, DiagPrbs) and the feature status of ReadDbi and WriteDbi. In the case of PRBS mode, PRBS is sent and received on every lane. In the case of pattern mode, data is only sent on the lane under test, and the other lanes are static, 0 in the case of LPDDR4 and 1 in the case of DDR4.

5.4 TEST 0x2 : Send Burst Writes

“Send burst writes” test generates large amounts of write traffic on the bus for customers to observe the trained-behavior of their system during writes. During the burst write test, the phy will send continuous write bursts to the row address provided in two different banks (or bank groups for DDR4), nominally generating write data on all lanes using the input data-pattern. There is a break in DQ traffic continuity for every DiagLoopCount iteration.

Field Used	Range	Description
DiagPrbs	[0,2]	0: Reserved 1: Prbs23 mode 2: pattern mode
DiagRepeatCount	[0,255]	Number of bursts per loop
DiagLoopCount	[0,255]	How many loops to run (multiples of 255)
DiagXCnt	[0,255]	How many times to iterate (0 and 1 will iterate once)
DiagRank		Which rank to test
DiagAddrHigh, DiagAddrLow		Row address to activate
DiagPatternHigh, DiagPatternLow		32 bit-time repeating Pattern to send when [DiagPrbs==2 pattern mode]
DiagMisc4[1]	[0,1]	0: finite burst write 1: infinite burst write (DiagLoopCount & DiagXCount are ignored)

Example test pseudocode:

```
For DiagXCnt {
    For (DiagLoopCount*255) {
        Write diagAddr diagRepeatCount times
    }//for
}//for
```

No data is returned by the write memory test and there is no guarantee that the written data will persist in memory once the test has ended.

When in pattern mode, the data pattern restarts at every iteration of the DiagXcnt Loop.

5.5 TEST 0x3 : Send Burst Reads

The send burst read test generates large amounts of read traffic on the bus for customers to observe the trained-behavior of their system during reads.

The burst read test issues DiagRepeatCount times READ commands for alternating banks, to read data from the specified row. There is a break in DQ traffic continuity for every DiagLoopCount iteration.

Field Used	Range	Description
DiagRepeatCount	[0,255]	Number of repeated bursts per loop
DiagLoopCount	[0,255]	How many repeated loops to run (multiples of 255)
DiagXCnt	[0,255]	How many times to iterate (0 and 1 will iterate once)
DiagRank		Which rank to test
DiagAddrHigh, DiagAddrLow		Row address to activate
DiagMisc4[1]	[0,1]	0: finite burst read 1: infinite burst read (DiagLoopCount & DiagXCount are ignored)

Example test pseudocode:

```
For DiagXCnt {
        Write diagAddr diagRepeatCount times
        For (DiagLoopCount*255) {
                  Read diagAddr DiagRepeatCount times
        }//for
}//for
```

Limitations:
The data received on DQ lanes is completely random as there’s no WR operation involved in this test.

5.6 TEST 0x4 : Simple Write Read

Field Used	Range	Description
DiagPrbs	[0,2]	0: Reserved 1: Prbs23 mode 2: pattern mode
DiagRepeatCount	[0, 127] [Don’t care]	Number of extra bursts per loop(DiagMisc1=2) If DiagMisc1=0
DiagLoopCount	[0, 255]	Number of extra bursts per loop
DiagRank		Which rank to test
DiagAddrHigh, DiagAddrLow		Row address to activate
DiagPatternHigh, DiagPatternLow		32 bit-time repeating Pattern to send when [DiagPrbs==2 pattern mode]

The simple write read test quickly proves that writes and reads to the dram devices are working.

Normally, the write read test sends 63 continuous writes to the target row address using the data pattern provided. The write burst is followed by a read burst checking the contents of the memory addresses just written. If DiagLoopCount is non-zero, the target row will then be overwritten and read back another DiagLoopCount times.

The simple read write test returns data with the following encodings.

Table 5-1 Read Write Diag pass/fail data

Value	Description
0x0	Lane passed
0x1	Lane failed
0xff	Lane was not tested

Results are reported beginning at the address tied to the DiagReturnData messageblock field. Each piece of data is one byte in width. The first byte of the return data is a global pass/fail; it will be set to 1 if any lane in the phy saw errors. The rest of the results report on a per-lane basis which lanes saw errors (1) or ran without issue (0).

Current firmware only supports the read fifo option, which results in a fixed number of reads and writes for each loop iteration, which is 8 for LPDDR5 and 4 for LPDDR4.

Table 2-2 Read Write Test Return Data

Byte Offset	Result	Size in Bytes
0	Pass/Fail	1
1	Reserved	1
2	Dbyte 0 Lane 0 pass/fail	1
3	Dbyte 0 Lane 1 pass/fail	1
4	Dbyte 0 Lane 2 pass/fail	1
6	Dbyte 0 Lane 3 pass/fail	1
7	Dbyte 0 Lane 4 pass/fail	1
8	Dbyte 0 Lane 5 pass/fail	1
9	Dbyte 0 Lane 6 pass/fail	1
10	Dbyte 0 Lane 7 pass/fail	1
11	Dbyte 0 Lane 8 pass/fail	1
12	Dbyte 1 Lane 0 pass/fail	1
2+(9*N)+M	Dbyte N Lane M pass fail	1

The sequence for this test is as follows:

1. All DRAMS on all channels and ranks are sent the exit power down command
2. All channels and ranks are sent the exit self-refresh command
3. The row address specified in DiagAddrLow and DiagAddrHigh is activated
4. The test is run on both channels and the selected rank for DiagLoopCount+1 iterations
5. The complete row is written
6. The complete row is read
7. The hardware training registers is checked for errors
8. All lanes are set to either pass fail or not tested
9. All channels and ranks are sent the enter self-refresh command
10. All channels and ranks are sent the power down command
11. The tests exits

5.7 TEST 0x5 : Tx Eye

Field Used	Range	Description
DiagPrbs	[0,2]	0: Reserved 1: Prbs23 mode 2: pattern mode
DiagRepeatCount	[0, 127] [Don’t care]	Number of extra bursts per loop(DiagMisc1=2) If DiagMisc1=0
DiagLoopCount	[0,255]	How many extra loops to run
DiagXCnt	[0,255]	How many times to iterate (0 and 1 will iterate once)
DiagByte		Which byte to measure
DiagLane	[0,8] [9]	Which dq to measure in DiagByte [LPDDR5 only] measure Write Data Link ECC, MR22[6:4]=1 is required.
DiagRank		Which rank to test
DiagAddrHigh, DiagAddrLow		Row address to activate
DiagPatternHigh, DiagPatternLow		Pattern to send when DiagPrbs is 2 (pattern mode)
DiagVrefInc	[1, 127]	Controls granularity of voltage domain. Larger number means less resolution.

DiagMisc0	DiagMisc0 = 0
	DiagMisc0 = 1
	DiagMisc0 = 2
		No filtering
		Ignore odd bits
		Ignore even bits
		All other values reserved

DiagMisc1	DiagMisc1 = 0	Use Write Read FIFO Reserved All other values reserved

The TxEye test collects the transmit eye associated with a specifc byte and bit. The eye is measured by running traffic while varying the DRAM VREF and Phy's TxDq delay settings. The test records how many errors occur at each delay and voltage setting and returns a matrix of encoded error counts through the DiagReturnData messageblock field.

TxEye sends a continuous group of DiagRepeatCount write bursts to the target row address using the data pattern provided. The write burst is followed by a continuous group of DiagRepeatCount read burst checking the contents of the memory addresses just written. If DiagLoopCount or diagXCnt is non-zero, the target row will then be overwritten and read back another DiagLoopCount*DiagXCnt times. Once finished the test reads the errors on the target lane and stores the results. TxEye repeats this set of reads and writes for the powerset of all device voltages and txDqDelay settings, storing values all the while.
Error counts for each (delay,voltage) pair are 16 bit values compressed into a single byte to save space. The compression scheme is as follows.

Compressed Result Value	Actual Error count
X=[0x0,0x7f]	X
X=[0x80,0xfe]	(X&0x7f) <<7 (approximate)
X=0xff	> 16256 (approximate)

DiagReturnData's results for TxEye are mapped as described below, including the dimensions of the error matrix (nVref x nDly) and the center point found by previous training stages in addition to the error matrix itself. Please note that the data starts with the minimum voltage and increases until the maximum voltage.

Byte Offset	Description	Size in Bytes
0	Number of Delays (nDly)	1
1	Number of Vrefs (nVREF)	1
2	Trained DRAM VREF	1
3	Reserved	1
4	Trained TxDqDelay CSR Value	2
6	Error Count (Delay=0, Vref=0)	1
…	…	…
6+nDly	Error Count (Delay=nDly, Vref=0)	1
7+nDly	Error Count (Delay=0, Vref=1)	1
…	…	..
6+(2*nDly)	Error Count (Delay=nDly, Vref=1)	1
…		…
6+(nVref*nDly)	Error Count (Delay=nDly, Vref=nVref)	1

The rows of the matrix (index nVref) correspond to the DRAM Mode Register value for VrefDQ.

The following equation is used to convert the row index to the VrefDQ:

VrefDQ= (row index) * DiagVrefInc

To convert VrefDq to Volts, please see the Jedec specification for the DRAM protocol of interest.

The columns of the matrix are in units of 1/64*UI.

72columns are correspond to the DQ delay (relative to DQS) -4/64*UI: +68/64*UI from the trained position.

Column Index 0 = Trained Position - 4/64*UI
Column Index 36 = Trained Position
Column Index 71 = Trained Position +68/64*UI

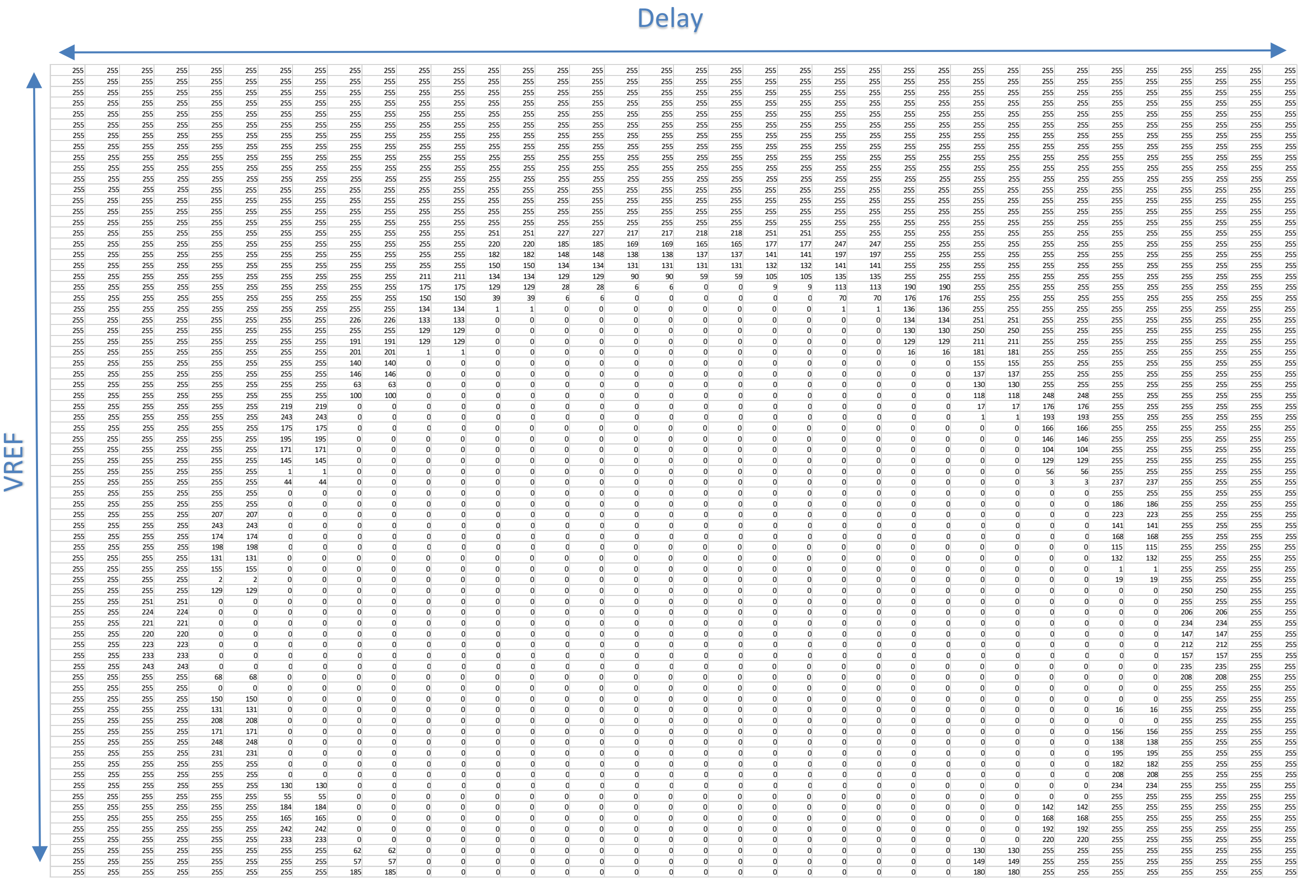
In the case where the Trained position of TxDq is less than 36/64*UI the column indexes correspond to:
Column Index 0 = Trained Position - Trained Position/64* UI

Example test pseudocode:

```
CenterDelay = read (csr_txdqdly)
centerVoltage = read messageblock
For voltage (0 to max in steps DiagVrefInc)
For delays (max(0,centerDelay-36) to centerDelay+35){
For DiagXCnt{
                        For DiagLoopCnt {
                            Write diagAddr DiagRepeatCount times
                            Read diagAddr DiagRepeatCount times
                        }//for
                    }//for
                    Record errors
                }//for
            }//for
```

Example Messageblock ErrorCount Output

TxEye
nVref = 127
nDly = 72



255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	251	251	227	227	217	217	218	218	251	251	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	220	220	185	185	169	169	165	165	177	177	247	247	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	182	182	148	148	138	138	137	137	141	141	197	197	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	150	150	134	134	131	131	131	131	132	132	141	141	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	211	211	134	134	129	129	90	90	59	59	105	105	135	135	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	175	175	129	129	28	28	6	6	0	0	9	9	113	113	190	190	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	150	150	39	39	6	6	0	0	0	0	0	0	70	70	176	176	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	134	134	1	1	0	0	0	0	0	0	0	0	1	1	136	136	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	226	226	133	133	0	0	0	0	0	0	0	0	0	0	0	0	134	134	251	251	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	129	129	0	0	0	0	0	0	0	0	0	0	0	0	130	130	250	250	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	191	191	129	129	0	0	0	0	0	0	0	0	0	0	0	0	129	129	211	211	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	201	201	1	1	0	0	0	0	0	0	0	0	0	0	0	0	16	16	181	181	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	140	140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	155	155	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	146	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137	137	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	63	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	130	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	100	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	118	118	248	248	255	255	255	255	255	255
255	255	255	255	255	255	219	219	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	17	176	176	255	255	255	255	255	255
255	255	255	255	255	255	243	243	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	193	193	255	255	255	255	255	255
255	255	255	255	255	255	175	175	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	166	166	255	255	255	255	255	255
255	255	255	255	255	255	195	195	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	146	146	255	255	255	255	255	255
255	255	255	255	255	255	171	171	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	104	255	255	255	255	255	255
255	255	255	255	255	255	145	145	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	255	255	255	255	255	255
255	255	255	255	255	255	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	56	56	255	255	255	255	255	255
255	255	255	255	255	255	44	44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	237	237	255	255	255	255
255	255	255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255	255	255
255	255	255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	186	186	255	255	255	255
255	255	255	255	207	207	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	223	223	255	255	255	255
255	255	255	255	243	243	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	141	141	255	255	255	255
255	255	255	255	174	174	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	168	255	255	255	255
255	255	255	255	198	198	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	115	115	255	255	255	255
255	255	255	255	131	131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	132	132	255	255	255	255
255	255	255	255	155	155	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	255	255	255	255
255	255	255	255	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	19	255	255	255	255
255	255	255	255	129	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	250	250	255	255
255	255	251	251	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255
255	255	224	224	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	206	206	255	255
255	255	221	221	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	234	234	255	255
255	255	220	220	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	147	147	255	255
255	255	223	223	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	212	212	255	255
255	255	233	233	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	157	157	255	255
255	255	243	243	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	235	235	255	255
255	255	255	255	68	68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	208	208	255	255
255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255
255	255	255	255	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255
255	255	255	255	131	131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	16	255	255	255	255
255	255	255	255	208	208	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255
255	255	255	255	171	171	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	156	156	255	255	255	255
255	255	255	255	248	248	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	138	138	255	255	255	255
255	255	255	255	231	231	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	195	195	255	255	255	255
255	255	255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	182	182	255	255	255	255
255	255	255	255	255	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	208	208	255	255	255	255
255	255	255	255	255	255	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	234	234	255	255	255	255
255	255	255	255	255	255	55	55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255	255	255	255	255	255	255
255	255	255	255	255	255	184	184	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	142	142	255	255	255	255	255	255
255	255	255	255	255	255	165	165	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	168	168	255	255	255	255	255	255
255	255	255	255	255	255	242	242	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	192	192	255	255	255	255	255	255
255	255	255	255	255	255	233	233	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	220	220	255	255	255	255	255	255
255	255	255	255	255	255	255	255	62	62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	130	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	57	57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	149	149	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	185	185	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180	180	255	255	255	255	255	255	255	255

5.8 TEST 0x6 : Rx Eye

Field Used	Range	Description
DiagPrbs	[0,2]	0: Reserved 1: Prbs23 mode 2: pattern mode
DiagRepeatCount	[0, 127] [Don't care]	Number of extra bursts per loop(DiagMisc1=2) If DiagMisc1=0
DiagLoopCount	[0,255]	How many extra loops to run
DiagXCnt	[0,255]	How many times to iterate (0 and 1 will iterate once)
DiagByte		Which byte to measure
DiagLane	[0,8]	Which dq to measure
DiagRank		Which rank to test
DiagAddrHigh, DiagAddrLow		Row address to activate
DiagPatternHigh, DiagPatternLow		32 bit-time repeating Pattern to send when [DiagPrbs==2 pattern mode]
DiagVrefInc	[1, 127]	Controls granularity of voltage domain. Larger number means less resolution.
DiagMisc0	DiagMisc0[3:0] = 0 DiagMisc0[3:0] = 1 DiagMisc0[3:0] = 2	No filtering Ignore Dqs_C bits Ignore Dqs_T bits All other values reserved

DiagMisc1	DiagMisc1 = 0
	DiagMisc1 = 1
		Use Write Read FIFO
		Use Read DQ Calibration
		All other values reserved

DiagMisc2	DiagMisc2 = 0
	DiagMisc2 = 1
		Scan all VREFS
		Scan Only VREF0

	DiagMisc2 = 2	Scan Only VREF1
	DiagMisc2 = 4	Scan Only VREF2
	DiagMisc2 = 8	Scan Only VREF3
		All other values reserved
DiagMisc3	DiagMisc3 = 0	No DFE Filtering
	DiagMisc3 = 1	Filter PrevVal=1
	DiagMisc3 = 2	Filter PrevVal=0
		All other values reserved
DiagMisc4[0]	DiagMisc4[0]=0	Use 2UI fine delay in RxClk dly
	DiagMisc4[0]=1	Use only 1 UI fine delay in RxClk dly

Note	If DiagMisc1=1, Rx test runs with Read DQ Calibration. This will alter some mode register value at the SDRAM. The affected MR register are MR15, MR20, MR32 and MR40 in LPDDR4. The affected MR register are MR31, MR32, MR33 and MR34 in LPDDR5.

The RxEye test collects the receive eye associated with a specifc byte and bit. The eye is measured by running traffic while varying the Phy's VREF and RxClkDly delay settings. The test records how many errors occur at each delay and voltage setting and returns a matrix of encoded error counts through the DiagReturnData messageblock field.

RxEye sends a continuous group of DiagRepeatCount write bursts to the target row address using the data pattern provided. The write burst is followed by a continuous group of DiagRepeatCount read burst checking the contents of the memory addresses just written. If DiagLoopCount or diagXCnt is non-zero, the target row will then be overwritten and read back another DiagLoopCount*DiagXCnt times. Once finished the test reads the errors on the target lane and stores the results. RxExe repeats this set of reads and writes for the powerset of all phy voltages and rxClkDelay settings, storing values all the while.
Current firmware only supports the read fifo option, which results in a fixed number of reads and writes for each loop iteration, which is 8 for LPDDR5 and 4 for LPDDR4.

Error counts for each (delay,voltage) pair are 16 bit values compressed into a single byte to save space. The compression scheme is as follows.

Compressed Result Value	Actual Error count
X=[0x0,0x7f]	X
X=[0x80,0xfe]	(X&0x7f) <<7 (approximate)
X=0xff	> 16256 (approximate)

DiagReturnData's results for RxEye are mapped as described below, including the dimensions of the error matrix (nVref x nDly) and the center point found by previous training stages in addition to the error matrix itself. Please note that the results are reported from minimum voltage to maximum voltage.

Byte Offset	Description		Size in Bytes
0	Number of Delays (nDly)		1
1	Number of Vrefs (nVref)		1
2	Trained phy VrefDAC0		1
3	Trained phy VrefDAC1		1
4	Trained phy VrefDAC2		1
5	Trained phy VrefDAC3		1
6	Trained Rxclkdly CSR Value		2
8	Error Count (Delay=0, Vref=0)		1
…	…		…
8+Dly	Error Count (Delay=Dly, Vref=0)		1
8+(nDly*Vref)+1	Error Count (Delay=1, Vref=1)		1
…	…		..
8+(Vref*nDly)+Dly	Error Count (Delay=Dly, Vref=Vref)		1

The rows of the matrix (index nVref) correspond to the PHY Register value for VrefDAC0.

The following equation is used to convert the row index to the VrefDAC0 setting:

VrefDAC0 = (row index) * DiagVrefInc

To convert VrefDAC0 to Volts, please see the PUB databook description for PHY Register VrefDAC0

The columns of the matrix are in units of 1/64*UI.

38 columns correspond to the DQS delay (relative to DQ).
Since the insertion delay of DQS may be larger than DQ, the eye is mapped starting at -4/64*UI.

Column Index 0   = -4/64*UI
Column Index 4 = 0
Column Index 67 = +63/64*UI

Example test pseudocode:

```
    CenterDelay = read (csr_rxclkdly)
    centerVoltage = read messageblock
    For voltage (0 to 127 in steps DiagVrefInc)
        For delays (-4/64UI to 68/64 UI){
            For DiagXCnt{
                For DiagLoopCnt {
                    Write diagAddr DiagRepeatCount times
                    Read diagAddr DiagRepeatCount times
                }//for
            }//for
        Record errors
        } //for
    }//for
```

VREF:

When sweeping vref, the phy receiver range is limited to maintain trained difference between vrefDAC0, vrefDAC1, vrefDAC2 and vrefDAC3. Returned data is always relative to VREFDAC0

Example Messageblock ErrorCount Output

nVref = 127

nDly = 72

Delay

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	244	244	251	251	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	245	245	230	230	245	245	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	224	224	218	218	238	238	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	237	237	209	209	206	206	229	229	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	213	213	194	194	194	194	218	218	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	238	238	193	193	180	180	182	182	205	205	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	213	213	173	173	167	167	168	168	190	190	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	193	193	160	160	157	157	158	158	176	176	247	247	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	231	231	173	173	148	148	147	147	146	146	161	161	240	240	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	207	207	158	158	140	140	139	139	138	138	150	150	230	230	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	188	188	144	144	133	133	132	132	132	132	140	140	217	217	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	247	247	173	173	137	137	130	130	129	129	129	129	134	134	203	203	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	219	219	157	157	131	131	39	39	23	23	24	24	130	130	188	188	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	200	200	144	144	91	91	1	1	0	0	1	1	113	113	173	173	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	182	182	135	135	4	4	0	0	0	0	0	0	28	28	161	161	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	168	168	130	130	0	0	0	0	0	0	0	0	4	4	151	151	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	247	247	154	154	35	35	0	0	0	0	0	0	0	0	0	0	141	141	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	224	224	143	143	1	1	0	0	0	0	0	0	0	0	0	0	136	136	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	202	202	133	133	0	0	0	0	0	0	0	0	0	0	0	0	132	132	250	250	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	186	186	129	129	0	0	0	0	0	0	0	0	0	0	0	0	130	130	237	237	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	170	170	27	27	0	0	0	0	0	0	0	0	0	0	0	0	129	129	225	225	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	159	159	5	5	0	0	0	0	0	0	0	0	0	0	0	0	56	56	211	211	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	238	238	145	145	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	11	196	196	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	218	218	136	136	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	182	182	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	198	198	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	170	170	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	183	183	94	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	159	159	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	169	169	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	151	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	157	157	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	144	144	252	252	255	255	255	255	255	255
255	255	255	255	255	255	255	255	144	144	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	139	139	237	237	255	255	255	255	255	255
255	255	255	255	255	255	255	255	136	136	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	134	134	221	221	255	255	255	255	255	255
255	255	255	255	255	255	243	243	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	132	132	208	208	255	255	255	255	255	255
255	255	255	255	255	255	226	226	129	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	130	194	194	255	255	255	255	255	255
255	255	255	255	255	255	207	207	18	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	181	181	255	255	255	255	255	255
255	255	255	255	255	255	192	192	14	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	103	103	170	170	255	255	255	255	255	255
255	255	255	255	255	255	176	176	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	26	160	160	255	255	255	255	255	255
255	255	255	255	255	255	163	163	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	14	152	152	255	255	255	255	255	255
255	255	255	255	255	255	150	150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	147	255	255	255	255	255	255
255	255	255	255	255	255	141	141	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	140	140	255	255	255	255	255	255
255	255	255	255	245	245	134	134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	136	136	255	255	255	255	255	255
255	255	255	255	230	230	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	134	134	255	255	255	255	255	255
255	255	255	255	212	212	69	69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	131	131	245	245	255	255	255	255
255	255	255	255	199	199	30	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	130	226	226	255	255	255	255
255	255	255	255	182	182	12	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	212	212	255	255	255	255
255	255	255	255	173	173	12	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	76	76	197	197	255	255	255	255
255	255	255	255	158	158	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	42	42	185	185	255	255	255	255
255	255	255	255	150	150	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	13	174	174	255	255	255	255
255	255	255	255	140	140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	166	166	255	255	255	255
255	255	255	255	135	135	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	159	159	255	255	255	255
255	255	255	255	131	131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	153	255	255	255	255
255	255	255	255	129	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152	152	255	255	255	255
255	255	255	255	64	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	152	152	255	255	255	255
255	255	255	255	71	71	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	155	155	255	255	255	255
255	255	255	255	118	118	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	160	160	255	255	255	255
255	255	255	255	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	11	168	168	255	255	255	255
255	255	255	255	134	134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	178	178	255	255	255	255
255	255	255	255	140	140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53	53	189	189	255	255	255	255
255	255	255	255	146	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100	204	204	255	255	255	255
255	255	255	255	156	156	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	219	219	255	255	255	255
255	255	255	255	164	164	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	130	130	236	236	255	255	255	255
255	255	255	255	180	180	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	132	132	250	250	255	255	255	255
255	255	255	255	194	194	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	134	134	255	255	255	255	255	255
255	255	255	255	212	212	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	137	137	255	255	255	255	255	255
255	255	255	255	226	226	23	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	7	142	142	255	255	255	255	255	255
255	255	255	255	247	247	130	130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25	25	147	147	255	255	255	255	255	255
255	255	255	255	255	255	132	132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	35	154	154	255	255	255	255	255	255
255	255	255	255	255	255	139	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	85	85	165	165	255	255	255	255	255	255
255	255	255	255	255	255	146	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	123	173	173	255	255	255	255	255	255
255	255	255	255	255	255	158	158	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	187	187	255	255	255	255	255	255
255	255	255	255	255	255	170	170	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	131	131	202	202	255	255	255	255	255	255
255	255	255	255	255	255	186	186	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	133	216	216	255	255	255	255	255	255
255	255	255	255	255	255	201	201	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	135	135	231	231	255	255	255	255	255	255
255	255	255	255	255	255	222	222	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	140	140	245	245	255	255	255	255	255	255
255	255	255	255	255	255	235	235	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	143	143	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	129	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	150	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	132	132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	158	158	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	140	140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	167	167	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	148	148	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	180	180	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	161	161	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	195	195	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	175	175	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	17	207	207	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	195	195	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	66	222	222	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	210	210	80	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	236	236	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	232	232	132	132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	129	129	250	250	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	250	250	139	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	131	131	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	151	151	0	0	0	0	0	0	0	0	0	0	0	0	0	0	133	133	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	162	162	0	0	0	0	0	0	0	0	0	0	0	0	0	0	137	137	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	179	179	0	0	0	0	0	0	0	0	0	0	0	0	1	1	141	141	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	197	197	16	16	0	0	0	0	0	0	0	0	0	0	3	3	149	149	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	222	222	130	130	0	0	0	0	0	0	0	0	0	0	13	13	159	159	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	242	242	137	137	0	0	0	0	0	0	0	0	0	0	22	22	170	170	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	150	150	0	0	0	0	0	0	0	0	0	0	59	59	184	184	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	162	162	3	3	0	0	0	0	0	0	0	0	129	129	198	198	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	179	179	127	127	0	0	0	0	0	0	0	0	130	130	213	213	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	198	198	133	133	0	0	0	0	0	0	1	1	133	133	226	226	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	220	220	143	143	2	2	0	0	0	0	5	5	138	138	239	239	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	244	244	157	157	94	94	1	1	0	0	17	17	147	147	249	249	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	175	175	133	133	7	7	7	7	64	64	158	158	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	192	192	141	141	129	129	117	117	130	130	170	170	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	217	217	151	151	136	136	134	134	137	137	186	186	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	238	238	166	166	145	145	143	143	145	145	202	202	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	186	186	155	155	153	153	157	157	217	217	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	205	205	167	167	165	165	170	170	230	230	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	234	234	180	180	178	178	184	184	240	240	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	196	196	190	190	198	198	247	247	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	222	222	207	207	215	215	252	252	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	242	242	218	218	226	226	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	231	231	237	237	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	246	246	245	245	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	251	251	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

VREF

5.9 Test 9 : Mode Register Write

The "Mode Register Write" test issues MRW command to the specified rank and channel in order to program a given mode register (MR).

Mode Register Write test parameters

Field used	Range	Description
DiagRank	[0, 1]	Which rank to test
DiagAddrLow	LPDDR4: [0, 40] DDR5: [0, 63]	Which MR to write
DiagAddrHigh	[0, 255]	Value to be written
DiagChannel	[0, 1]	ChannelA or ChannelB

After the Mode Register Write test, a MRR is issued by the firmware. The returned data of MRR is reported in DiagReturnData for user to compare with the intended written value specified at DiagAddrHigh.

MRR returned data issued after the MRW

Byte offset	Description	Size in bytes
0	Actual mode register value written	1

The sequence for this test is as follows:

LPDDR4/5:

1. All DRAMs on all channels and ranks are sent the power down exit and self-refresh exit commands
2. A MRS command is sent to the specified rank and channel
3. A MRR command is send to the specified rank and channel. Returned data is written to DiagReturnData.
4. All DRAMs on all channels and ranks are sent the self-refresh enter and power down commands
5. The test exits

5.10 Test 10 : Mode Register Read

The "Mode Register Read" test issues a MRR command to the specified rank in order to read a given mode register (MR). This test is only supported for LPDDR5 memories. For LPDDR4, a subset of the mode registers can be read using the MPR Read test described in **Section 5.7**.

Mode Register Read test parameters

Field used	Range	Description
DiagRank	[0, 1]	Which rank to test
DiagAddrLow	LPDDR4: [0,40] DDR5: [0,63]	Which MR to read

Results are reported beginning at the DiagReturnData message block field. MR value of all byte will be reported in DiagReturnData area.

The following table illustrates how the MR values are stored in the DiagReturnData area.

Mode Register Read test return data

Byte offset	Description	Size in bytes
0	Nnmber of bytes return N	2
2	Byte index (eg:byte0)	1
3	MRR value of this byte (eg:byte0)	1
4	Byte index (eg:byte1)	1
5	MRR value of this byte (eg:byte1)	1
…		
(N*2)	Byte index (eg:byteN)	1
(N*2)+1	MRR value of this byte (eg:byteN)	1

The sequence for this test is as follows:

1. All DRAMs on all channels and ranks are sent the power down exit and self-refresh exit commands
2. A MRR command is sent to the specified rank
3. The returned MR values are stored in the return section of the diagnostics message block
4. All DRAMs on all channels and ranks are sent the self-refresh enter and power down commands
5. The test exits