International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 10 (2015) pp. 25143-25153

http://www.ripublication.com

Functional Coverage For DDR4 Memory Controller

M.A.Srivatsav*[1], Dr.Kakarla Hari Kishore[*2]
[1] *Student of VLSI Systems Research Group, Department of Electronics and Communication Engineering, K L University, AP – INDIA*
Email: vatsav@outlook.com
[2]*Asso. Professor VLSI Systems Research Group, Department of Electronics and Communication Engineering, KL University, AP – INDIA*
Email: kakarla.harikishore@kluniversity.in

Abstract

The main objective of this paper is to verify the performance of DDR4 memory controller using the functional coverage. Functional coverage is user defined process of how much of design specification is exercised. Functional coverage is mainly based on the design specification and it is independent of actual design code. In this paper Functional coverage for DDR4 commands and all types of different configurations of bank groups is exercised. Simulation results are performed through the UVM 1.2 Library in Modelsim 10.2 version.

Keywords: Memory controller, DDR, Interface Bus Techniques

Introduction

DDR4 SDRAM is Double Data Rate Fourth generation Synchronous Dynamic Random Access Memory. DDR4 has many advanced features than its predecessor DDR3. DDR4 SDRAM comes with X4, X8 and X16 configurations. It operates at 1.2V between 1600 to 3200 MHz, firstly the JEDEC standard DDR4 specification document is taken and with reference of this document the verification plan is prepared. The verification plan is nothing but a standard verification procedure in verification domain. The aim of verification plan is to prepare a hierarchical process of complete verification procedure. In the verification plan functional coverage plays a major role. In the verification plan with reference to the specifications one analyses what to verify and where to start the verification plan. Once the specification reading is finished, the verification plan for DDR4 Memory controller will be created and according to the verification plan the functional coverage will be done.

The below figure represents block diagram for DDR4 SDRAM memory controller core. Northwest Logic's Double Data Rate 4 (DDR4) SDRAM Controller Core is

designed for use in applications requiring high memory throughput, high clock rates and full programmability. The core accepts commands using a simple local interface and translates them to the command sequences required by DDR4 SDRAM devices. The core also performs all initialization, refresh and power-down functions.

The core uses bank management modules to monitor the status of each SDRAM bank. Banks are only opened or closed when necessary, minimizing access delays. Up to 32 banks can be managed at one time. The core queues up multiple commands in the command queue. This enables optimal bandwidth utilization for both short transfers to highly random address locations as well as longer transfers to contiguous address space. The command queue is also used to opportunistically perform look-ahead activates, pre-charges and auto-pre-charges further improving overall throughput.

The core supports all new DDR4 features, including: 3DS device configurations, write CRC, data bus inversion (DBI), fine granularity refresh, additive latency, per-DRAM addressability, and temperature controlled refresh. Add-On Cores such as a Multi-Port Front-End and Reorder Core can be optionally delivered with the core. The core is delivered fully integrated and verified with the target DDR PHY. Northwest Logic supports a broad range of third party and its own soft DDR PHY.

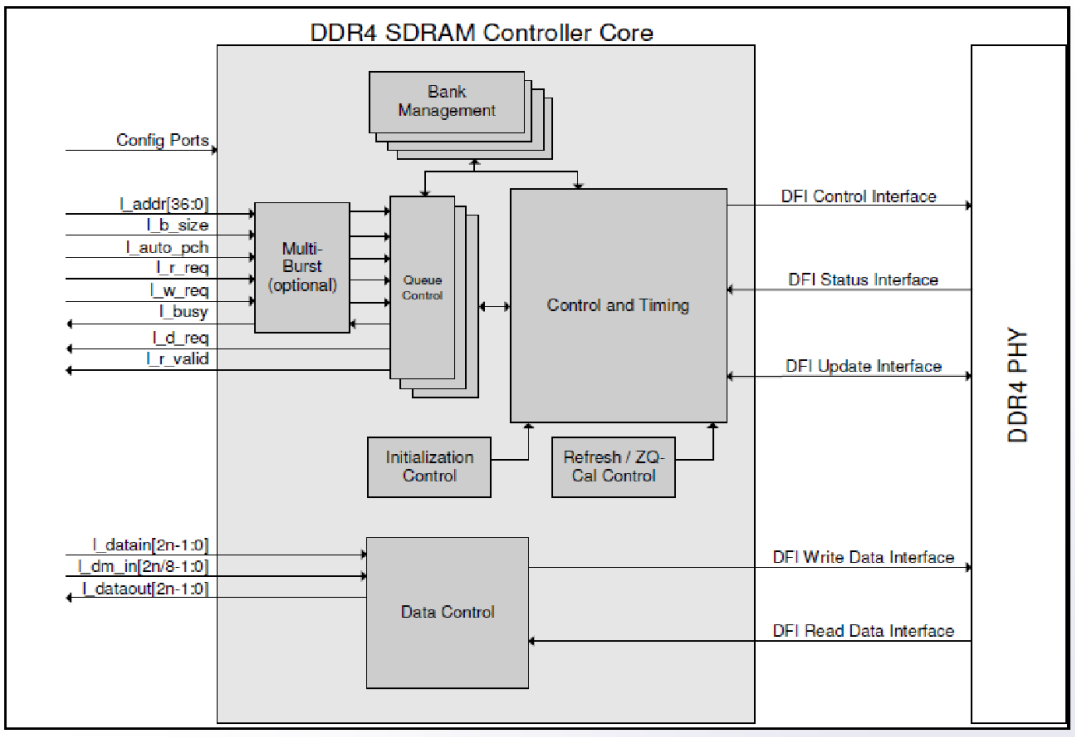


Figure 1: Block Diagram for DDR4 SDRAM Controller Core

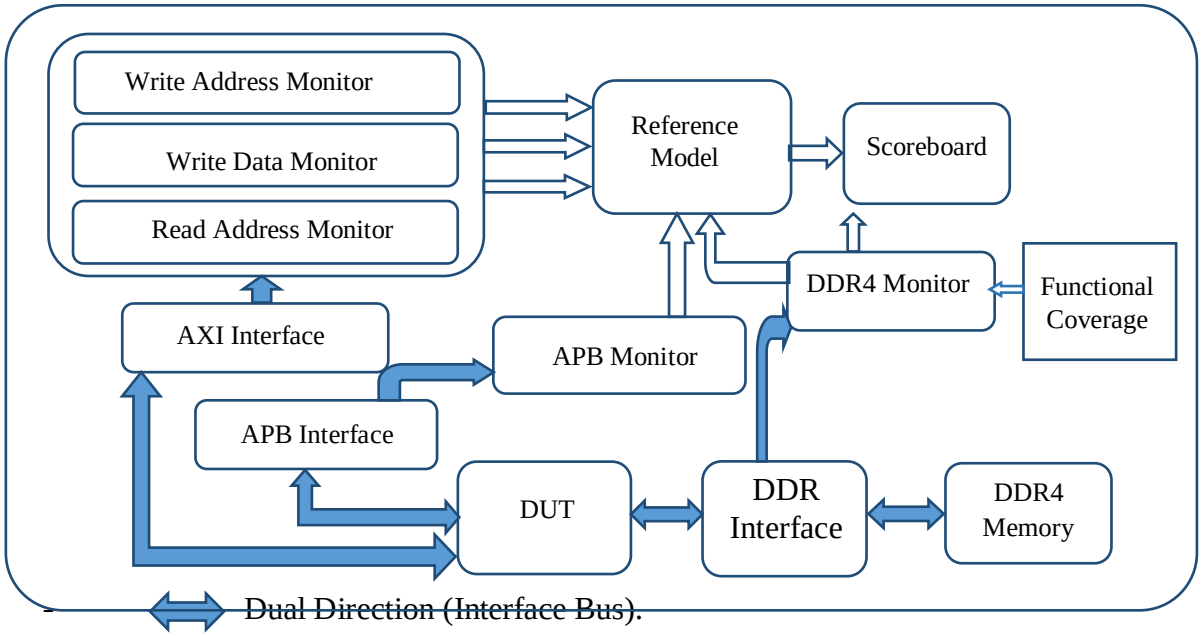
The highlights of block diagram are given below.

- Maximizes bus efficiency via Look-Ahead command processing, Bank Management, Auto-Pre-charge and Additive Latency support

- Minimal latency achieved via parameterized pipelining Achieves high clock rates with minimal routing constraints
- Supports half-rate and quarter rate clock operation
- Supports DDR4 3DS device configurations
- Multi-mode controller support
- Full run-time configurable timing parameters and memory settings
- DFI 3.1 Compatible including full PHY training support
- Full set of Add-On Cores available
- Delivered fully integrated and verified with target DDR PHY
- RDIMM and LRDIMM support
- Minimal ASIC gate count
- Broad range of ASIC and FPGA platforms supported
- Source code available
- Customization and Integration services available

Section II represents the Block Diagram for DDR4 Memory Controller and Section III represents Description of Functional Coverage. Finally Simulation Results analysis on Section IV

Block Diagram Ddr4 Memory Controller



- Dual Direction (Interface Bus).
- Uni-Directional (Interface-Monitor Connection).
- Uni-directional (TLM Port Connection).

Figure 2: Block Diagram of DDR4 Memory Controller

Monitor
A monitor extracts signal information from a bus and translates the information into a transaction that can be made available to other components like the scoreboard. The monitor just takes the transaction items from the interface and passes to the other components like scoreboard. The monitor and the score board communication is can be done by UVM analysis port. Once after the data is available in the monitor, with the analysis port the data can be transferred to the scoreboard.

Reference Model (RM):
It imitates the work done by the Memory Controller. There are five different monitors connected to the RM, three of which captures from the Write Address Channel, Write Data Channel, and Read Address Channel from the AXI Interface. The fourth monitor captures data from the APB Interface and the fifth monitor capture from the DDR4 Interface. These connections are made using UVM analysis ports.

There are two different functionalities of the RM:

1. Verifying the commands generated by the DUT.
2. Verifying the Data read out from the memory module.

Scoreboard
The scoreboard main objective is to compare the data and show the result. It takes the RM data and the DUT data through monitor and compares the data and displays error if any mismatches or else it shows success.

AXI Interface
The AXI Interface drives data, address through the Read Channel and the Write Channel into the DUT. The key features of the AXI protocol being:

1. Separate address/control and data phases.
2. Supports unaligned data transfers, using byte strobes and address.
3. It supports Increment burst-based transactions.
4. It is having separate read and write data channels.
5. It can issue multiple outstanding addresses.
6. Supports out-of-order transaction completion.

APB Interface
The APB Interface drives the control and status registers inside the DUT which in turn helps in timing control and the MRS command generation. Hence, the APB monitor is connected to the RM so as to implement the same behavior as the DUT and to generate of the MRS commands.

Also, the RM is connected to the DDR4 monitor who captures from the DDR4 interface. The commands generated by the DUT, the data read out from the DDR4 memory is captured by the DDR4 monitor as it is driven to the DDR4 memory. This DDR4 monitor contains one analysis port using which the commands generated by the DUT are sent to the RM and the data read out from the DDR4 memory is sent to the scoreboard.

Functional Coverage

The functional coverage covers the set of signal transitions from one to another like zero to one. In the functional coverage the set of signals were taken by creating the cover groups. Once after the cover group a cover point will be created for each and every signal or command. Bins will be created for each and every cover point according to the necessity of the range of the given signal bins range can also be limited. Soon after the bins were created we can sample the coverage. We can even go for the cross coverage for one or more cover points or variables.

Description on Functional Coverage

In this paper, according to the verification plan a UVM subscriber class is created. In this subscriber class coverage will be generated. A different cover group will be created for the signals and the bank group. In this cover group the cover point will be created for the each signal and soon after bins will be created for each cover point. Once the bins created, a cross coverage can be done and automatic bins will be created for the cross coverage. After this, according to the specification we combine the all signals so that it becomes a command. Once commands are generated, we create the object for the cover group and sampled in the top class. Another cover group will be created for the bank groups. DDR4 memory controller has the X4, X8 and X16 configurations so that according to the verification plan cover group is created. Cover points will be created for the bank address and bank groups and all types of addressing modes like 2GB, 4GB, 8GB and 16GB. Once after the cover group created, an object will be created for the cover group and sampled in the top class. Once the bins get 100% covered, coverage will be satisfied.

Simulation Results

ModelSim Covergroup Coverage Report

Scope: /ddr4_top_sv_unit/ddr4_coverage

Show All | Show Covered | Show Missing

Covergroups / Instances	Total Bins	Covered Bins (Hits)	% Hit	Goal	Coverage	% of Goal
Covergroup cmd_group	155	155	100.00%	100.00%	100.00%	**100.00%**
Instance \/ddr4_top_sv_unit::ddr4_coverage::cmd_group per_instance true	155	155	100.00%	100.00%	100.00%	**100.00%**
Covergroup bank_group merge_instance auto(true)	22	22	100.00%	100.00%	100.00%	**100.00%**
Instance \/ddr4_top_sv_unit::ddr4_coverage::bank_group per_instance true	22	22	100.00%	100.00%	100.00%	**100.00%**

Covergroup type:cmd_group				100.00%	100.00%	100.00%
Coverpoints / Bins	**At Least**	**Total Bins**	**Covered Bins (Hits)**	**% Hit**	**Goal**	**Coverage**	**% of Goal**
Coverpoint: ddr4_cke		4	4	100.00%	100.00%	100.00%	100.00%
Coverpoint: ddr4_cs_n		2	2	100.00%	100.00%	100.00%	100.00%
Coverpoint: ddr4_act_n		2	2	100.00%	100.00%	100.00%	100.00%
Coverpoint: ddr4_ras_n_A16		2	2	100.00%	100.00%	100.00%	100.00%
Coverpoint: ddr4_cas_n_A15		2	2	100.00%	100.00%	100.00%	100.00%
Coverpoint: ddr4_we_n_A14		2	2	100.00%	100.00%	100.00%	100.00%
Cross: ddr4_cross <ddr4_cke,ddr4_cs_n,ddr4_act_n,ddr4_ras_n_A16,ddr4_cas_n_A15,ddr4_we_n_A14>		141	141	100.00%	100.00%	100.00%	100.00%

Covergroup instance:\/ddr4_top_sv_unit::ddr4_coverage::cmd_group							
				100.00%	100.00%	100.00%	**100.00%**
Coverpoints / Bins	**At Least**	**Total Bins**	**Covered Bins (Hits)**	**% Hit**	**Goal**	**Coverage**	**% of Goal**
Coverpoint: ddr4_cke (covered 4 of 4 bins) (missing 0 of 4 bins)		4	4	100.00%	100.00%	100.00%	100.00%
bin lowtohigh	1		230			Covered	--
bin hightolow	1		231			Covered	--
bin hightohigh	1		246			Covered	--
bin lowtolow	1		292			Covered	--
Coverpoint: ddr4_cs_n (covered 2 of 2 bins) (missing 0 of 2 bins)		2	2	100.00%	100.00%	100.00%	100.00%
bin low	1		517			Covered	--
bin high	1		483			Covered	--
Coverpoint: ddr4_act_n (covered 2 of 2 bins) (missing 0 of 2 bins)		2	2	100.00%	100.00%	100.00%	100.00%
bin low	1		491			Covered	--
bin high	1		509			Covered	--

Coverpoint: ddr4_ras_n_A16 (covered 2 of 2 bins) (missing 0 of 2 bins)		2	2	100.00%	100.00%	100.00%	100.00%
bin low	1		481			Covered	--
bin high	1		519			Covered	--
Coverpoint: ddr4_cas_n_A15 (covered 2 of 2 bins) (missing 0 of 2 bins)		2	2	100.00%	100.00%	100.00%	100.00%
bin low	1		503			Covered	--
bin high	1		497			Covered	--
Coverpoint: ddr4_we_n_A14 (covered 2 of 2 bins) (missing 0 of 2 bins)		2	2	100.00%	100.00%	100.00%	100.00%
bin low	1		512			Covered	--
bin high	1		488			Covered	--

Cross: ddr4_cross (covered 141 of 141 bins) (missing 0 of 141 bins) <ddr4_cke,ddr4_cs_n,ddr4_act_n,ddr4_ras_n_A16,ddr4_cas_n_A15,ddr4_we_n_A14>		141	141	100.00%	100.00%	100.00%	100.00%
bin <lowtohigh,low,low,low,low,low>	1		8			Covered	--
bin <hightohigh,low,low,low,low,low>	1		8			Covered	--
bin <hightolow,low,low,low,low,low>	1		5			Covered	--
bin <lowtolow,low,low,low,low,low>	1		9			Covered	--
bin <hightolow,high,low,low,low,low>	1		5			Covered	--
bin <hightohigh,high,low,low,low,low>	1		6			Covered	--
bin <lowtolow,high,low,low,low,low>	1		5			Covered	--
bin <hightolow,low,high,low,low,low>	1		10			Covered	--
bin <hightohigh,low,high,low,low,low>	1		6			Covered	--
bin <lowtolow,low,high,low,low,low>	1		7			Covered	--
bin <lowtohigh,high,high,low,low,low>	1		4			Covered	--
bin <hightohigh,high,high,low,low,low>	1		1			Covered	--
bin <hightolow,high,high,low,low,low>	1		8			Covered	--
bin <lowtolow,high,high,low,low,low>	1		9			Covered	--
bin <lowtohigh,low,low,high,low,low>	1		11			Covered	--
bin <lowtohigh,high,low,high,low,low>	1		8			Covered	--
bin <hightohigh,low,low,high,low,low>	1		8			Covered	--
bin <hightohigh,high,low,high,low,low>	1		12			Covered	--
bin <hightolow,low,low,high,low,low>	1		10			Covered	--
bin <hightolow,high,low,high,low,low>	1		7			Covered	--
bin <lowtolow,low,low,high,low,low>	1		9			Covered	--
bin <lowtolow,high,low,high,low,low>	1		10			Covered	--
bin <hightolow,low,high,high,low,low>	1		4			Covered	--
bin <hightohigh,low,high,high,low,low>	1		10			Covered	--

bin <lowtolow,low,high,high,low,low>	1		15			Covered	--
bin <lowtohigh,high,high,high,low,low>	1		6			Covered	--
bin <hightohigh,high,high,high,low,low>	1		7			Covered	--
bin <hightolow,high,high,high,low,low>	1		5			Covered	--
bin <lowtolow,high,high,high,low,low>	1		11			Covered	--
bin <lowtohigh,low,low,low,high,low>	1		10			Covered	--
bin <lowtohigh,low,low,high,high,low>	1		9			Covered	--
bin <lowtohigh,high,low,low,high,low>	1		6			Covered	--
bin <lowtohigh,high,low,high,high,low>	1		8			Covered	--
bin <hightohigh,low,low,low,high,low>	1		6			Covered	--
bin <hightohigh,low,low,high,high,low>	1		4			Covered	--
bin <hightohigh,high,low,low,high,low>	1		5			Covered	--
bin <hightohigh,high,low,high,high,low>	1		7			Covered	--
bin <hightolow,low,low,low,high,low>	1		7			Covered	--
bin <hightolow,low,low,high,high,low>	1		9			Covered	--
bin <hightolow,high,low,low,high,low>	1		4			Covered	--
bin <hightolow,high,low,high,high,low>	1		7			Covered	--
bin <lowtolow,low,low,low,high,low>	1		7			Covered	--
bin <lowtolow,low,low,high,high,low>	1		8			Covered	--
bin <lowtolow,high,low,low,high,low>	1		10			Covered	--
bin <lowtolow,high,low,high,high,low>	1		7			Covered	--
bin <hightolow,low,high,low,high,low>	1		6			Covered	--
bin <hightolow,low,high,high,high,low>	1		9			Covered	--
bin <hightohigh,low,high,low,high,low>	1		12			Covered	--
bin <hightohigh,low,high,high,high,low>	1		11			Covered	--
bin <lowtolow,low,high,low,high,low>	1		9			Covered	--

bin <lowtolow,low,high,high,high,low>	1		21			Covered	--
bin <lowtohigh,high,high,low,high,low>	1		9			Covered	--
bin <lowtohigh,high,high,high,high,low>	1		8			Covered	--
bin <hightohigh,high,high,low,high,low>	1		9			Covered	--
bin <hightohigh,high,high,high,high,low>	1		9			Covered	--
bin <hightolow,high,high,low,high,low>	1		6			Covered	--
bin <hightolow,high,high,high,high,low>	1		10			Covered	--
bin <lowtolow,high,high,low,high,low>	1		8			Covered	--
bin <lowtolow,high,high,high,high,low>	1		10			Covered	--
bin <lowtohigh,low,low,low,low,high>	1		7			Covered	--
bin <lowtohigh,low,high,low,low,high>	1		12			Covered	--
bin <lowtohigh,high,low,low,low,high>	1		10			Covered	--
bin <lowtohigh,high,high,low,low,high>	1		8			Covered	--
bin <hightohigh,low,low,low,low,high>	1		10			Covered	--
bin <hightohigh,low,high,low,low,high>	1		8			Covered	--
bin <hightohigh,high,low,low,low,high>	1		8			Covered	--
bin <hightohigh,high,high,low,low,high>	1		12			Covered	--
bin <hightolow,low,low,low,low,high>	1		10			Covered	--
bin <hightolow,low,high,low,low,high>	1		7			Covered	--
bin <hightolow,high,low,low,low,high>	1		6			Covered	--
bin <hightolow,high,high,low,low,high>	1		5			Covered	--
bin <lowtolow,low,low,low,low,high>	1		9			Covered	--
bin <lowtolow,low,high,low,low,high>	1		6			Covered	--
bin <lowtolow,high,low,low,low,high>	1		7			Covered	--
bin <lowtolow,high,high,low,low,high>	1		15			Covered	--
bin <lowtohigh,low,low,high,low,high>	1		4			Covered	--
bin <lowtohigh,high,low,high,low,high>	1		5			Covered	--

bin <hightohigh,low,low,high,low,high>	1		8			Covered	--
bin <hightohigh,high,low,high,low,high>	1		9			Covered	--
bin <hightolow,low,low,high,low,high>	1		9			Covered	--
bin <hightolow,high,low,high,low,high>	1		7			Covered	--
bin <lowtolow,low,low,high,low,high>	1		10			Covered	--
bin <lowtolow,high,low,high,low,high>	1		11			Covered	--
bin <hightolow,low,high,high,low,high>	1		8			Covered	--
bin <hightohigh,low,high,high,low,high>	1		5			Covered	--
bin <lowtolow,low,high,high,low,high>	1		9			Covered	--
bin <lowtohigh,high,high,high,low,high>	1		4			Covered	--
bin <hightohigh,high,high,high,low,high>	1		8			Covered	--
bin <hightolow,high,high,high,low,high>	1		7			Covered	--
bin <lowtolow,high,high,high,low,high>	1		6			Covered	--
bin <lowtohigh,low,low,low,high,high>	1		5			Covered	--
bin <lowtohigh,low,high,low,high,high>	1		3			Covered	--
bin <lowtohigh,high,low,low,high,high>	1		6			Covered	--
bin <lowtohigh,high,high,low,high,high>	1		6			Covered	--
bin <hightohigh,low,low,low,high,high>	1		9			Covered	--
bin <hightohigh,low,high,low,high,high>	1		1			Covered	--
bin <hightohigh,high,low,low,high,high>	1		7			Covered	--
bin <hightohigh,high,high,low,high,high>	1		6			Covered	--
bin <hightolow,low,low,low,high,high>	1		10			Covered	--
bin <hightolow,low,high,low,high,high>	1		10			Covered	--
bin <hightolow,high,low,low,high,high>	1		7			Covered	--
bin <hightolow,high,high,low,high,high>	1		7			Covered	--
bin <lowtolow,low,low,low,high,high>	1		7			Covered	--
bin <lowtolow,low,high,low,high,high>	1		7			Covered	--

bin <lowtolow,low,low,low,high,high>	1		7			Covered	--
bin <lowtolow,low,high,low,high,high>	1		7			Covered	--
bin <lowtolow,high,low,low,high,high>	1		11			Covered	--
bin <lowtolow,high,high,low,high,high>	1		12			Covered	--
bin <hightolow,low,low,high,high,high>	1		8			Covered	--
bin <hightohigh,low,low,high,high,high>	1		10			Covered	--
bin <lowtolow,low,low,high,high,high>	1		6			Covered	--
bin <lowtohigh,high,low,high,high,high>	1		3			Covered	--
bin <hightohigh,high,low,high,high,high>	1		8			Covered	--
bin <hightolow,high,low,high,high,high>	1		2			Covered	--
bin <lowtolow,high,low,high,high,high>	1		8			Covered	--
bin <hightolow,low,high,high,high,high>	1		7			Covered	--
bin <hightohigh,low,high,high,high,high>	1		7			Covered	--
bin <lowtolow,low,high,high,high,high>	1		4			Covered	--
bin <hightohigh,high,high,high,high,high>	1		9			Covered	--
bin <lowtolow,high,high,high,high,high>	1		9			Covered	--

bin sre	1		7			Covered	--
bin srx	1		5			Covered	--
bin pre	1		7			Covered	--
bin prea	1		7			Covered	--
bin act	1		11			Covered	--
bin wr	1		6			Covered	--
bin wrs4	1		6			Covered	--
bin wrs8	1		6			Covered	--
bin wra	1		6			Covered	--
bin wras4	1		6			Covered	--
bin wras8	1		6			Covered	--
bin rd	1		8			Covered	--
bin rds4	1		8			Covered	--
bin rds8	1		8			Covered	--
bin rda	1		8			Covered	--
bin rdas4	1		8			Covered	--
bin rdas8	1		8			Covered	--
bin nop	1		5			Covered	--
bin des	1		8			Covered	--
bin pde	1		9			Covered	--
bin pdx	1		10			Covered	--
bin zqcl	1		8			Covered	--
bin zqcs	1		8			Covered	--

Bank Groups Coverage

Covergroup type:bank_group				100.00%	100.00%	100.00%
Coverpoints / Bins	**At Least**	**Total Bins**	**Covered Bins (Hits)**	**% Hit**	**Goal**	**Coverage**	**% of Goal**
Coverpoint: DDR4_bg0 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin a[1]	1		494			Covered	--
Coverpoint: DDR4_bg1 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin a[1]	1		515			Covered	--
Coverpoint: DDR4_Addr0 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		488			Covered	--
Coverpoint: DDR4_Addr1 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		488			Covered	--
Coverpoint: DDR4_Addr2 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		515			Covered	--
Coverpoint: DDR4_Addr3 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		507			Covered	--

Coverpoint: DDR4_Addr4 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		486			Covered	--
Coverpoint: DDR4_Addr5 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		495			Covered	--
Coverpoint: DDR4_Addr6 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		497			Covered	--
Coverpoint: DDR4_Addr7 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		485			Covered	--
Coverpoint: DDR4_Addr8 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		487			Covered	--
Coverpoint: DDR4_Addr9 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		499			Covered	--

Coverpoint: DDR4_Addr10 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		497			Covered	--
Coverpoint: DDR4_Addr11 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		495			Covered	--
Coverpoint: DDR4_Addr12 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		524			Covered	--
Coverpoint: DDR4_Addr13 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		500			Covered	--
Coverpoint: DDR4_Addr14 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		488			Covered	--
Coverpoint: DDR4_Addr15 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		497			Covered	--

Coverpoint: DDR4_Addr16 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		519			Covered	--
Coverpoint: DDR4_Addr17 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		518			Covered	--
Coverpoint: DDR4_ba0 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin c[1]	1		481			Covered	--
Coverpoint: DDR4_ba1 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin c[1]	1		490			Covered	--

Covergroup instance:\/ddr4_top_sv_unit::ddr4_coverage::bank_group				100.00%	100.00%	**100.00%**	
Coverpoints / Bins	**At Least**	**Total Bins**	**Covered Bins (Hits)**	**% Hit**	**Goal**	**Coverage**	**% of Goal**
Coverpoint: DDR4_bg0 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin a[1]	1		494			Covered	--
Coverpoint: DDR4_bg1 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin a[1]	1		515			Covered	--
Coverpoint: DDR4_Addr0 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		488			Covered	--
Coverpoint: DDR4_Addr1 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		488			Covered	--
Coverpoint: DDR4_Addr2 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		515			Covered	--
Coverpoint: DDR4_Addr3 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		507			Covered	--
Coverpoint: DDR4_Addr4 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		486			Covered	--
Coverpoint: DDR4_Addr5 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		495			Covered	--
Coverpoint: DDR4_Addr6 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		497			Covered	--
Coverpoint: DDR4_Addr7 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		485			Covered	--
Coverpoint: DDR4_Addr8 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		487			Covered	--
Coverpoint: DDR4_Addr9 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		499			Covered	--
Coverpoint: DDR4_Addr10 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		497			Covered	--
Coverpoint: DDR4_Addr11 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		495			Covered	--
Coverpoint: DDR4_Addr12 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		524			Covered	--
Coverpoint: DDR4_Addr13 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		500			Covered	--
Coverpoint: DDR4_Addr14 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		488			Covered	--
Coverpoint: DDR4_Addr15 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		497			Covered	--

Coverpoint: DDR4_Addr16 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		519			Covered	--
Coverpoint: DDR4_Addr17 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin b	1		518			Covered	--
Coverpoint: DDR4_ba0 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin c[1]	1		481			Covered	--
Coverpoint: DDR4_ba1 (covered 1 of 1 bins) (missing 0 of 1 bins)		1	1	100.00%	100.00%	100.00%	100.00%
bin c[1]	1		490			Covered	--

Conclusion

In DDR4 memory controller the functional coverage through the UVM library 1.2 in Modelsim 10.2 version tool and the Bus group coverage analysis successfully completed with accurate approximations.

References

[1] DDR4 JEDEC Specification.
[2] UVM Class Reference Manual.
[3] https://www.altera.com/en_US/pdfs/literature/hb/external-memory/emi_plan_board_ddr2.pdf
[4] http://www.memcon.com/pdfs/proceedings2013/track2/Designing_for_DDR4_Power_and_Performance.pdf