International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 9 Sep. 2022, pp: 301-305 www.ijaem.net ISSN: 2395-5252

Verification of GDDR6 DRAM Features Using System Verilog and UVM

Amruth N [1] and Dr. Kariyappa B S [2]
[1] Student, RV College of Engineering, Bangalore.
[2] Professor,RV College of Engineering, Bangalore.

Date of Submission: 01-09-2022 Date of Acceptance: 10-09-2022

ABSTRACT: As the performance requirements of digital systems continue to increase, there are increasing requirements to deliver devices that enables reliable operation at higher accessing rates. The digital design of the Graphics Double Data Rate 6 DRAM memory is verified using System Verilog and Universal Verification Methodology to ensure the DRAM works appropriately as instructed by the memory controller. The functional features considered for verification are initialization, command address training, word clock to command clock training. And maximum code coverage and functional coverage is achieved in the verification process of these features. Initialization of GDDR6 is accomplished by providing the power supplies, command, and data path initialization. Command address training is achieved to GDDDR6 to align the command path with the command clock for accurate sampling of command addresses and to ensure that GDDR6will receive the commands properly as sent by the memory controller in real time. The purpose of word clock to command clock (WCK2CK) training is to align the command clock with the internal data word clock of GDDR6 and vendor ID is issued from the GDDR6 to memory controller. Code coverage of 97.42% is achieved for the digital design by covering the blocks, expressions, signal, and port toggling implemented in the design verification process. And the achieved functional coverage is 100% for the cover groups and assertions built for the verification of memory model design.

KEYWORDS:GDDR6, UVM, WCK2CK, DRAM

I. INTRODUCTION

Graphics Double Data Rate 6 memory has got a higher speed and it finds applications in the devices which requires higher bandwidth and used mainly in play stations, graphic consoles, electronic devices, high performance computational devices and gaming laptops with high resolution graphic requirements. GDDR6 is provided by the two differential clocks for command/address sampling and data sampling. Data sampling happens at both rising and falling edges of the data word clock, by which it enhances the data access speed in the memory. The digitally designed GDDR6 memory allows up to16 Gb/sec data rate. GDDR6 being an advanced memory model has got some special features implemented at architecture level and these features help to create difference in speed and other aspects. The memory model designed as per the defined features is supposed to be verified to make sure the design features will function appropriately afterthe silicon validation. The initialization of GDDR6 is verified by building a corresponding sequence which includes its sequential steps such as power supply ramp up, command path and data path initialization. The initialization is ensured by the handshake signals.

Similarly, the command address training provided to the GDDR6 DRAM is verified by sending the commands on the command path from the memory controller and feedback is received to adjust in the memory controller for accurate training. And this training is assured by the alignment of command path with the command clock for synchronous command and address sampling. Data word clock to command clock training is verified by driving the differential clocks of both command and data path and feedback is received for the synchronization and vendor ID information is received from the GDDR6 on the assertion of the control signal.

II. LITERATURE SURVEY

The memory model of GDDR6 is built with prefetch architecture for parallel access with two channels [1][2]. The operation of the DRAM is controlled by mode register commands. The functional verification of the digital design of any

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 9 Sep. 2022, pp: 301-305 www.ijaem.net ISSN: 2395-5252

memory model is a crucial step for the appropriate performance after the manufacture of the model [3].System Verilog is proven to be the best programming language for creating the testbench environment to verify the RTL design with object-oriented programming approach and the features of constraints and randomization [4].

The UVM provides a suitable environment to build thetestbench with multiple hierarchical components in verification [5].The reusability feature is very helpful in instantiating the existing modules and classes. Configuration database of UVM helps in storing the transaction information and can be utilized by other testbench components. This helps in the accessing the information in the different testbench architecture [6].

III. DESIGN METHODOLOGY

The verification environment is built to verify the design features of GDDR6 DRAM based on the universal verification methodology. The standard test bench architecture is created as shown in the Figure 1 with universal verification components to achieve verification for the register transfer level code written and the sequential steps of the methodology are discussed as follows:

Fig. 1 Verification Methodology

Testcase and Sequence Development: The testcases are defined for the features defined to verify and corresponding sequences are built using system Verilog. A top module is created in the sequence where all the verification components, design under test and interfaces are instantiated. The reusable feature of the UVM classes is enclosed in the environment of GDDR6 and the default configuration is defined as per the requirements.

Generation of Transaction Data: The transaction objects for the signals defined in each feature are generated in the transaction class. Signal declaration is accompanied by defined constraints on the values getting generated and randomized.

Driving the Signals: UVM driver drives the signals generated in the transaction class of GDDR6 to design under test through advanced peripheral bus interface. It uses the virtual instance of the interface to drive the transactions to the static design under test.

Result Comparison in Scoreboard: When the transactions are driven to the design under test, the results will get generated. Those results are compared with the expected results based on the logic built in the reference model written in the scoreboard.

Code Coverage and Functional Coverage: Code coverage is a measure to analyse how much percentage of the code has been exercised in the verification. Functional coverage is a measure to analyse the percentage of design features covered in the verification.

Waveform Analysis: The waveforms generated in the dump by running the testcases of the design features are verified accordingly for the signals driven at the design under test level.

IV. VERIFICATION OF GDDR6 FEATURES

The features of GDDR6 are verified sequentially because initialization of GDDR6 is the prerequisite for both command address training and data word clock to command clock training.

Initialization begins by powering up all the voltage pins and hence it will be allowed to function in the reset state. All voltage pins of GDDR6 are active high in the design. And reset pin is not lifted immediately after power supply to ensure all the signals are in high impedance state at the beginning. This guarantees no previous values of any pin of DRAM persisted after initialization. Input output drive voltage is maintained high after the power supply ramp up to protect the input and outputs of DRAM from the current surge. Command path and data path are initialized for both the channels of DRAM after power supply and the pins of path are asserted with default command and data. The initialization of GDDR6 is assured by the handshake signals.

In command address training, the two differential command address clocks are enabled and become stable on the assertion of the clock enablesignal. Commands are sent by the memory controller on theinitialized command path to the GDDR6. The command clock and command path are aligned to reduce the latency between the generation and sampling of the commands. Feedback is received on the training data output pin to make changes in the command generation at the memory controller side.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 9 Sep. 2022, pp: 301-305 www.ijaem.net ISSN: 2395-5252

After command address training is confirmed by the handshake signal, word clock to command clock training is done to DRAM to synchronize the command and data sampling. The clocks should align at a point to ensure there is no latency for the commanded data operation. The position or arrival of the data clock is sent as feedback to the memory controller from the DRAM. If the arrival of data clock is early compared to the address clock, then feedback is sent as early to the memory controller and if command clock comes first before the data clock, then the feedback is sent as late. Error detection code is considered to be the path to send the feedback.

Code coverage is the measure that calculates the amount or percentage of code of the digital design been verified. It covers the block of codes, expressions, statements and toggling of the ports and signals in the design. Functional coverage is metric which assures the percentage of design functionalities been exercised or covered in the verification process. In order to have all the scenarios of the design signals or functionalities to be covered, cover groups and assertions are built to hit the multiple speed bins, commands and addresses.

V. RESULTS AND DISCUSSION

The results are obtained by compiling and running the testcases using Cadence Xceliummain tool in linux environment. The waveforms are analysed using the Cadence Simvision waveform viewer due to good graphical interface. And the code coverage and functional coverage are generated by the Integrated Metrics Center (IMC) tool of Cadence.

The power supply is provided to the voltage pins of GDDR6 by driving the voltage pins high, shown in the Figure 2 and input output drive voltage signal is maintained for some time to protect the DRAM signals from current surge.

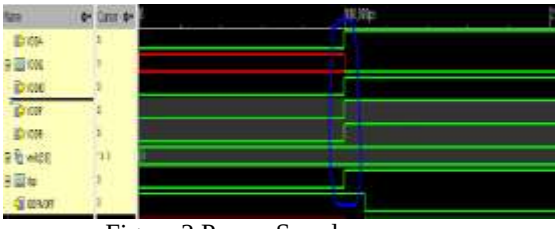

Figure 2 Power Supply ramp up

In Figure 3, initialization of command path is verified for all the pins including the command address bus inversion pin to reduce the power consumption.

Figure 3 Command Path Initialization

The data path is initialized for all the pins to make GDDR6 active in data transmission and sampling, shown in the Figure4 and data bus inversion line is asserted to reduce the DC power consumption.

Figure 4 Data Path Initialization

In command address training, commands are generated from the memory controller and sent on the command path line to GDDR6. Feedback is received on the CATDQ signal back to memory controller and the command path to command clock alignment is seen in the figure 5 and command address training is completed by the assertion of CATCOMPLETE signal.

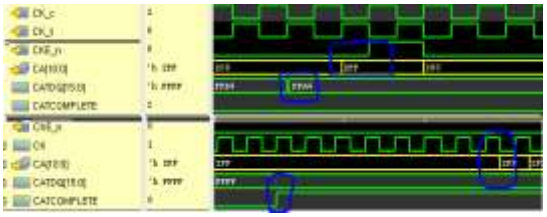

Figure 5 Command Address Training

In WCK2CK training, based on the arrival of the data word clock with respect to the command clock, the feedback is seen in the Error Detection Code (EDC) line as shown in the Figure6. Since data word clock has arrived late than the command clock, the feedback is seen to be low in the waveform.

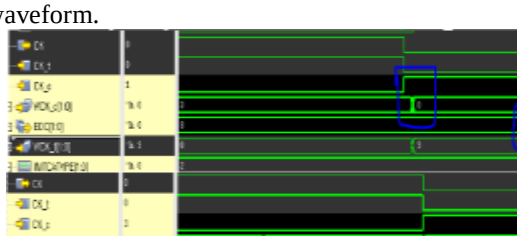

Figure 6 WCK2CK Training

International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 9 Sep. 2022, pp: 301-305 www.ijaem.net ISSN: 2395-5252

Once the WCK2CK training is achieved by the alignment of command clock with the internal data word clock, the GDDR6 can send the vendor ID through the label pin with the assertion of ID pin as shown in the Figure 7, to describe the information of the vendor to the memory controller.

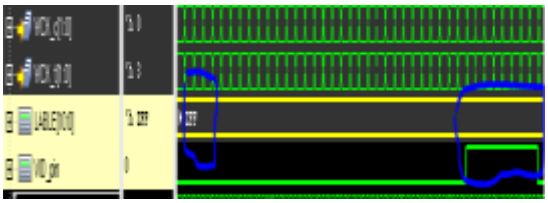

Figure 7 Transfer to vendor ID

The coverage closure report is generated using the Cadence IMC tool and in figure 8 code coverage achieved for the verified features to be 97.42%.

Figure 8 Code Coverage Report

In figure 9, the functional coverage for the cover groups and assertions implemented for the design features in the verification is 100%.

Figure 9 Functional Coverage Report

VI. CONCLUSION

Initialization and training features of GDDR6 DRAM have been verified for the memory model design and the corresponding wave forms are analysed. The challenges faced by traditional verification methodology are clearly outreached by UVM based verification. Architectural modifications are brought into design such as command address bus inversion and data bus inversion. Code coverage of 97.42% is achieved and cover groups help in covering all the randomized signal inputs, thus the functional coverage has reached to be 100%.

REFERENCES

[1]. K. Salah and H. Mostafa, "Constructing effective uvm testbench for dram memory controllers," in 2018 New Generation of CAS (NGCAS), IEEE, 2018, pp. 178–181.

[2]. P. Moorby, "Design for verification with system verilog," in 17th International Conference on VLSI Design. Proceedings., 2004, pp. 378–. doi: 10.1109/ICVD. 2004.1260952.

[3]. M.-K. You and G.-Y. Song, "Systemverilog- based verification environment using systemc custom hierarchical channel," in 2009 IEEE 8th International Conference on ASIC, 2009, pp. 1310–1313. doi:10.1109/ASICON.2009.5351242.

[4]. A. Goel, B. B. T. Sundari, and S. Mathew, "Uvm based controller area network verification ip (vip)," in 2020 International Conference on Smart Electronics and Communication (ICOSEC), IEEE, 2020, pp. 645–652.

[5]. M. Keaveney, A. McMahon, N. O'Keeffe, K. Keane, and J. O'Reilly, "The development of advanced verification environments using system verilog," in IET Irish Signals and Systems Conference (ISSC 2008), 2008, pp. 325–330.doi: 10.1049/cp: 20080683.

[6]. T. Pavithran andR. Bhakthavatchalu, "Uvmbased testbench architecture for logic sub-system verification," in 2017 International Conference on Technological Ad vancements in Power and Energy (TAP Energy), IEEE, 2017, pp. 1–5

[7]. A. T. Vanaraj, M. Raj, and L. Gopalakrishnan, "Functional verification closure using optimal test scenarios for digital designs," in 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), IEEE, 2020, pp.535– 538.

[8]. S. Das, R. Mohanty, P. Dasgupta, and P. Chakrabarti, "Synthesis of system verilog assertions," in Proceedings of the Design Automation Test in Europe Conference, vol. 2, 2006, pp. 1–6. doi: 10.1109/DATE.2006.243776.

[9]. J. Bromley, "If systemverilog is so good, why do we need the uvm? sharing re sponsibilities between libraries and the core language," in Proceedings of the

2013 Forum on specification and Design Languages (FDL), IEEE, 2013, pp. 1–7.

[10]. P. Vitankar and A. Kureshi, "Verification approach using uvm," International Journal of Innovations in Engineering Research and Technology, pp. 1–2,

[11]. K. Salah, "A uvm-based smart functional verification platform: Concepts, pros, cons, and opportunities," in 2014 9th International Design and Test Symposium (IDT), IEEE, 2014, pp. 94–99.

[12]. B. Madiwalar and B. Kariyappa, "Single bit-line 7t sram cell for low power and high snm," in 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), IEEE, 2013, pp. 223–228.

[13]. H. Ke, D. Zhongliang, and S. Qiong, "Verification of amba bus model using sys temverilog," in 2007 8th International Conference on Electronic Measurement and Instruments, 2007, pp. 1-776-1–780. doi: 10.1109/ICEMI.2007.4350567.

[14]. L. Nikitha, N. Bhargavi, and B. Kariyappa, "Design and development of nonvolatile multi-threshold schmitt trigger sram cell," in Emerging Research in Electronics, Computer Science and Technology, Springer, 2019, pp. 877–

[15]. P. S. Vitankar and A. Kureshi, "Uvm architecture for verification," International Journal of Electronics and communications Engineering and Technology, vol. 7, no. 3, pp. 29–37, 2016.

[16]. J. JESD250, "Graphics double data rate 6 (gddr6) sgram standard," JEDEC Solid State TechnologyAssociation, 2017.