

A system verilog approach for verification of memory controller

Sowmya K B, Gagana P

Department of Electronics and Communication Engineering, RV College of Engineering, India

Article Info

Article history:

Received Mar 3, 2020

Revised Mar 19, 2020

Accepted Apr 11, 2020

Keywords:

Configuration registers
Coverage report
RAM
ROM
Verification environment
Wishbone interface

ABSTRACT

Memory performance has become the major bottleneck to improve the overall performance of the computer system. By using memory controller, there is effective control of data between processor and memory. In this paper, a memory controller for interfacing Synchronous Static Random Access Memory (SSRAM), Synchronous Dynamic Random Access Memory (SDRAM), Read Only Memory (ROM) and FLASH which is Electrically Erasable Programmable Read-Only Memory is designed and a coverage driven Constraint random verification environment is built for the designed memory controller. Verification plays an important role in any design flow as it is done before silicon development. It is done at time of product development for quality checking and bug fixing in design.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Sowmya K B,
Department of Electronics and Communication Engineering,
RV College of Engineering,
Bengaluru, Karnataka, India.
Email: kb.sowmya@gmail.com

1. INTRODUCTION

The memory controller is a digital circuit that manages the flow of data going to and from the computer's main memory. A memory controller can be a separate chip or integrated into another chip. If it is an integral part of a microprocessor it is usually called an integrated memory controller (IMC). It acts as an interface between the processor and external peripherals for accessing memory.

Literature review is done, In [1] the author specifies about the specifications required for the design of memory controller like the configuration of registers and the address range for each of them. It also gave the information about how to connect various memory devices to the Memory Controller. It also specified the clock frequency and other constraints pertaining to memory, for example the refresh cycle time period for SDRAM (synchronous dynamic random-access memory). In [2] the author gave an insight of Integrated Memory Controller (IMC) which supports DDR3 protocols with two independent, 64-bit wide channels each accessing one or two DIMMs. The type of memory supported by the processor is dependent on the PCH SKU in the target platform. The memory controller has an advanced command scheduler where all pending requests are examined simultaneously to determine the most efficient request to be issued next. In [3] the author explains the basic functionality of SDR SDRAM (standard single data rate-synchronous dynamic random-access memory) devices, such as the command structure. In addition to this, DDR (double data rate) SDRAMs contain several enhancements like Double data rate, Source synchronous operation, Low voltage signaling and Differential clocks. In [5] in this constrained random verification, which is created by means of verification methodology manual (VMM) for system verilog and classification trees, is reusable, scalable, configurable and can reduce time of verification.

2. VERIFICATION ENVIRONMENT FOR MEMORY CONTROLLER

The functionality of the blocks are expressed as:

Generator: This generates scenarios or test cases such as register reset, register read write, sram read write, sdram read write, flash read write, synchronous memory(rom) read write, all chip selects connected with different memories, suspend and resume test. Figure 1 presents the block diagram of memory controller.

Driver (BFM): The only block which is driving the scenario to the DUT is the bus functionality model which models the wishbone bus. Figure 2 shows verification environment for memory controller.

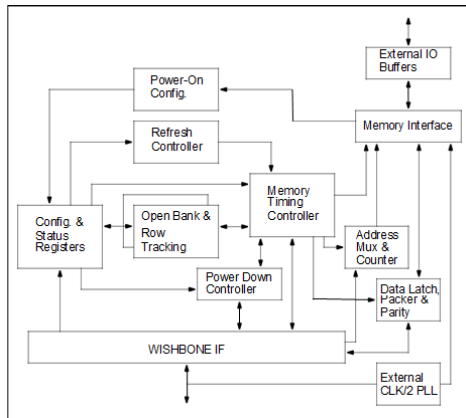


Figure 1. Block diagram of memory controller [1]

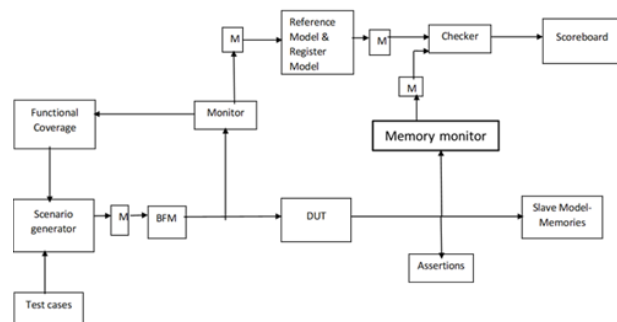


Figure 2. Verification environment for memory controller

Monitor: This is monitoring the signal level information that is the wishbone signals

Coverage (Functional coverage): check if all the functionalities of the design are covered or not. This includes individual signal coverage, register field coverage and scenario coverage

Register model: This model is the test bench equivalent representation of design registers by their individual fields.

Reference model: It is a model conceived in an early phase, before the RTL implementation, assumed as ideal. It simulates the DUT at a high level of abstraction, that is, creating reference output using register model.

Checker: This does the comparing of the actual output of DUT with the expected output given the reference model.

Scoreboard: This is responsible for keeping a track of pass/fail transaction counts, that is, if all the fields in the transaction matches with the expected transaction, then only the testcase is said to be passed. This is implemented as a part of checker.

Assertions: This is for checking signal level protocol behavior. There are two things in checking the signal behavior, that is, a positive check where it's checking if DUT is doing what it is expected to do and a negative check where it's checking if DUT is doing anything that it is not expected to do or for any unintended side effects after doing what it's expected to do.

3. TEST PLAN

In The following are the list of testcases implemented:

3.1. Test_register_reset

In this testcase, the reset signal is applied and released so as to verify if the register contents are actually reset on applying a synchronous reset signal as shown in Figure 3.

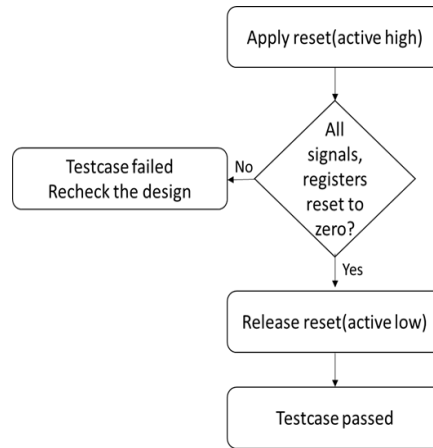


Figure 3. Register reset testcase

3.2. Test_register_write_read

In this testcase, write and read operations are performed to the registers to check for data corruption, that is, if the data written to a particular address is the same when read back from the same location as shown in Figure 4.

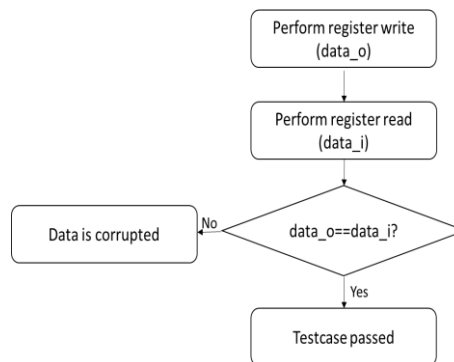


Figure 4. Register write-read testcase

3.3. Test_all_chipselects_sram

In this testcase, write and read operations are performed to the SRAM chips connected so as to check for data corruption only after configuring the registers to operate for SRAM as shown in Figure 5.

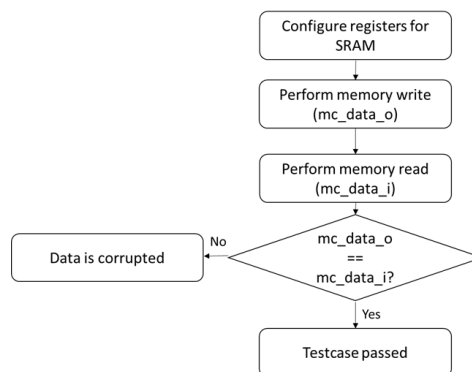


Figure 5. SRAM write-read testcase

3.4. Test_all_chipselects_sdram

In this testcase, write and read operations are performed to the SDRAM chips connected so as to check for data corruption only after configuring the registers to operate for SDRAM as shown in Figure 6.

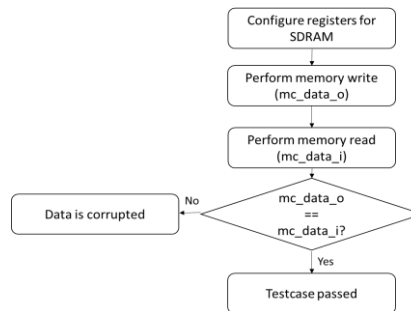


Figure 6. SDRAM write-read testcase

3.5. Test_all_chipselects_flash

In this testcase, write and read operations are performed to the FLASH to check for data corruption after configuring the registers to operate for FLASH

3.6. Test_all_chipselects_rom

In this testcase, write and read operations are performed to the ROM chips connected so as to check for data corruption only after configuring the registers to operate for ROM.

3.7. Test_all_chipselects_different_memories

In this testcase, write and read operations are performed to all the four different memory chips, that is, SRAM, SDRAM, FLASH, ROM connected so as to check for data corruption only after configuring the registers to operate for different memories.

3.8. Test_suspend_resume

In this testcase, after configuring the registers to operate for different memories, the suspend signal is pulled high after one write operation and after that read operation is performed. This testcase cannot read the data, as the operation which was suspended is not resumed as shown in Figure 7.

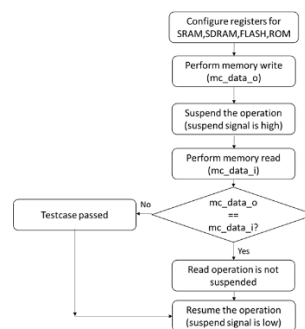


Figure 7. Suspend and resume testcase

4. CONCLUSION

A reusable, scalable and configurable environment and a coverage-driven constraint random-based coverage model are thus presented to verify the functions of a memory controller in a microprocessor in this paper. It's proved by the results that this method used to verify memory controller is more time-efficient than the directed test method. Future work in this direction involves developing an algorithm for automatic generation of memory access groups, given a set of memory timings and a burst size.

REFERENCES

- [1] Rudolf Usselmann, "Memory Controller IP Core", Open Cores Revision.1.7 January 21, 2002.
- [2] 2nd Generation Intel® Core™ Processor Family Desktop, Intel® Pentium® Processor Family Desktop, and Intel® Celeron® Processor Family Desktop Datasheet, Volume 1, 2015.
- [3] Barbara Johnson, "MSC711x Memory Controller Usage Guidelines Supporting Double Data Rate (DDR) SDRAM Devices", NXP, Revision. 1, 3/2007.
- [4] K. Khalifa, H. Fawzy, S. El-Ashry, K. Salah, "Memory controller architectures: A comparative study", *8th IEEE Design and Test Symposium*, Dec 2013.
- [5] Yingpan Wu, Lixin Yu, Lidong Lan, and Haiyang Zhou: "A Coverage-Driven Constraint Random-Based Functional Verification Method of Memory Controller" in *Proc. 19th IEEE/IFIP International Conf. Rapid System Prototyping*, pp. 99-104, Jun 2008.
- [6] K. Agarwal, V. K. Magraiya and A. K. Saxena, "Verification and Simulation of New Designed NAND Flash Memory Controller", *International Conference on Communication Systems and Network Technologies*, Gwalior, pp. 762-766, 2013.
- [7] Park Soo Il, Song Jae Yeol, Park Seok Hwi and Jung Ji Hoon, "Design of memory controller Design of general-purpose memory controller," *International SoC Design Conference*, Busan, pp. III-39-III-41, 2008.
- [8] I. Silas, I. Frumkin, E. Hazan, E. Mor, G. Zobin, "System-level validation of the intel pentium M processor," *Intel Technol. J.*, vol. 7, no. 2, pp. 38-43, May 2003.
- [9] Shawn Kung, "Native PCIs SSD controllers next generation enterprise architecture for scalable I/O performance", White Paper Marvell, Jan-2012.
- [10] A. Aharon, D. Goodman, M. Levinger et al., "Test program generation for functional verification of power PC processors in IBM", *Proc.32th DAC IEEE*, pp. 279-285.

BIOGRAPHIES OF AUTHORS



Sowmya K B received the B.E degree in Electronics and Communication Engineering from the Vivekananda College of Engineering and Technology, Puttur, India, in 2006, M.Tech degree in Electronics from the Sir. M. Visveswaraya Institute of Technology, Bengaluru, India, in 2012, bagging rank from VTU, Belagavi. She was working as Assistant Professor in PA College of Engineering, Mangaluru, India for Nine years. Currently she is working as Assistant Professor in R V College of Engineering, Bengaluru, India. Her research interests include VLSI and Signal Processing, Image processing, Low power Architectures and VLSI Design. She has contributed many National and International journals to various reputed journals and conference.



Ms. Gagana P is doing her Bachelor of Engineering in Electronics and Communication Department, RV College of Engineering, Bengaluru, India.