JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 5343-5350
https://publishoa.com
ISSN: 1309-3452

UVM based Verification of Watchdog Timer with APB

Sharath S G, Dr. Venkateshappa
REVA UNIVERSITY

Received: 2022 March 15; **Revised:** 2022 April 20; **Accepted:** 2022 May 10

Abstract

Background: The Number of transistors in recent computer chip has crossed billion mark. The VLSI design is becoming complex day by day. The Verification of such designs is a tedious process as majority of time of development is consumed in verification. Most the blocks in these chips are duplicated or replicated with some change in configuration. Verilog and System Verilog Verification Methodology doesn't have seamless support for reusability of the verification components.

Objectives: The verification of Watchdog Timer with APB interface using UVM.

Methods: UVM Verification Methodology uses System Verilog framework to build the testbench.UVM methodology defines various verification components. The tests of sequences are kept apart from the original testbench hierarchy which helps in reusability.

Results: Watchdog timer with APB interface is designed and verification is done using UVM verification methodology. APB Write / Read and Watchdog Timer functionalities verified. Random stimulus is generated.

Conclusions: Successfully designed the Watchdog Timer with APB interface by using Verilog and simulated with the UVM based Testbench.The main advantages of this method of verification are using the virtual random coverage driven, for which the verification engineer takes minimal time for verification in the complex design system. The tests and Verification components can be reused in a different design which uses APB Interface.

Keywords: Verification IP, System Verilog, UVM, SOC, APB protocol, Watchdog Timer.

1. Introduction

The fast development in Computer-Aided Design (CAD) and CMOS technology has helped in designing complex VLSI CHIPs. With this developed many usages in Intellectual Property Cores (IP). System on

CHIP – SOC design become more popular with the help of IP core combination. SOC Designs use BUS Protocols for synchronization and data transfer. Hence, the Design Functional Verification process is so important due to the reusability of IP cores in complex design (as the design requires 30 % of the Complete Development time and 70% for verification). To reduce the time in verification, Verification engineers have been developing verification methodologies for checking the functionality of the module by using an integrated verification environment and it is referred to as IP Verification. Generally, Advanced Peripheral Bus, Advanced high-performance Bus, and Advanced Extensible Interface or also called APB, AHB, and AXI respectively bus protocols are used in the modern SoC development by the current industry. Compare with protocols APB protocol consumes limited power and low chip area. In this environment, the proper test cases are implemented in this project for the operation of DUV (Design under verification).

Watchdog timer is used in the SOC to detect the unresponsive system. In this case watchdog timer resets the system which brings the system to normal operating condition.

This paper presents the development of Watchdog timer with APB interface using Verilog and the Verification of the same using the UVM. Further in this chapter APB Overview, Watchdog Timer Overview and UVM overview is provided to understand more about the Design and verification methodology.

APB Overview**:** APB is one of the Bus protocols of Advanced Microcontroller Bus Architecture (AMBA) Protocols. APB Bus protocol has low complexity, consumes less power and it's a low-cost interface. APB Protocol is a non-pipelined. APB protocol is used to Connect low – bandwidth peripherals whose functionality is not depending on the speed of operation. APB is mostly used to connect SOC with peripherals for configuring the functionality of the peripheral. APB Peripherals can be interfaced with AHB and AXI protocols using the AHB-APB or AXI-APB Bridges respectively.

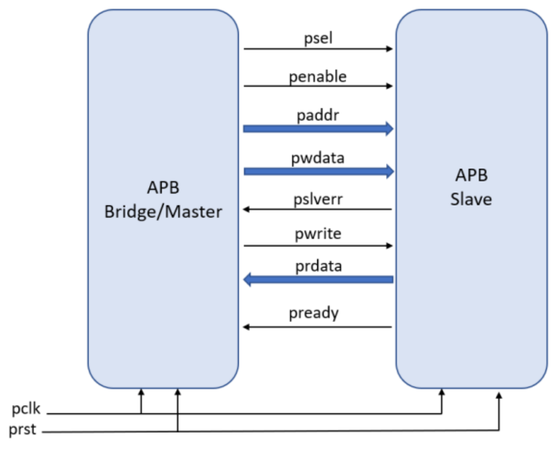

Fig 1. APB Interface Signals

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 5343-5350
https://publishoa.com
ISSN: 1309-3452

Here is a diagram of a System or SOC System. In this diagram AHB is the High-Speed Bus and APB is relatively low speed Bus. At High level, the high-performance components are connected with the core using AHB bus. The Low Bandwidth peripherals are connected to the Core using APB through Bridge. The ARM processor is the Core of the system provided in the block diagram. The low bandwidth peripherals like UART and Timer are connected to the system bus through the AHB – APB Bridge using the APB (Peripheral bus). Here, the Bridge is a AHB Slave to the Core AHB Master. The other components like High-bandwidth on-chip RAM, and High-bandwidth Memory Interface are connected to the Core by AHB (System Bus). And it also acts as the APB Master corresponding to remaining low-bandwidth external peripherals.

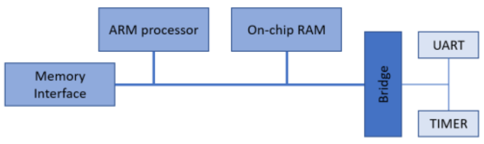

Fig 2. Block Diagram of a Typical System or SOC System

In the above diagram there are no components which generates the APB transfers. AHB – APB Bridge converts the AHB access to APB access and acts as the APB Master in the above system.

Watchdog Timer: A Watchdog Timer is a Hardware component implemented in System to automatically without human intervention detect the software failures and reset the processor if there are any failures. A Watchdog Timer in based on a counter with a pre-defined optimum initial value and is count down to zero value. The processor needs to reset the counter periodically indicating Processor is working fine. If the processor is not able to reset, it and the counter reaches to value zero then the software is presumed to be malfunctioning and processor is reset.

Typical Watchdog Timer Setup is Provided Below

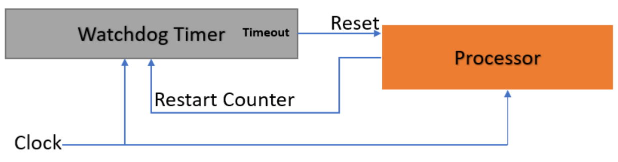

Fig 3. Typical Watchdog Timer Setup

Applications of Watchdog Timers are:Watchdog Timer are commonly implemented in Embedded System or SOCs controlled equipment where humans cannot easily control the equipment or unable to react to the faults in a timely manner. In Such Systems Watchdog Timers are implemented to invoke reset of the system as the computer cannot depend on the human to take reactive step.

JOURNAL OF ALGEBRAIC STATISTICS
Volume 13, No. 3, 2022, p. 5343-5350
https://publishoa.com
ISSN: 1309-3452

The Safety Compliant System needs to implement Watchdog timer to detect the System Faults.

UVM:A Universal Verification Methodology is a standardized methodology released by Accellera with the support from multiple vendors like Aldec (Riviera-PRO), Cadence (NCSim), Mentor Graphics (which is now Siemens EDA- Questa), Synopsys (VCS), Xilinx Simulator (XSIM) to create an automated verification environment. It consists of an open-source SV-based class library from which a functional testbench is built for any design file written in C, VHDL, Verilog, or SystemVerilog.

This methodology specifies a set of verification guidelines to be followed when a testbench is created. By following its approach, any verification engineer can develop verification components that are uniform and can be portable from project to project. This will reduce the effort of developing an unfamiliar environment and can easily modify any components as per requirement.

Advantages of UVM:

• It separates the test environment from the testbench. This way the environment is much more reusable.

• It follows a standard approach to developing a testbench which makes the verification flow stable.

• Supports transaction-level communications using TLM connections.

• It uses a set of uvm_root library classes which can customize any objects and components according to a particular requirement. They are Report messaging, Factory methods, End of test methods, RAL, etc.

• All advantages mentioned above will result in reduced coding work.

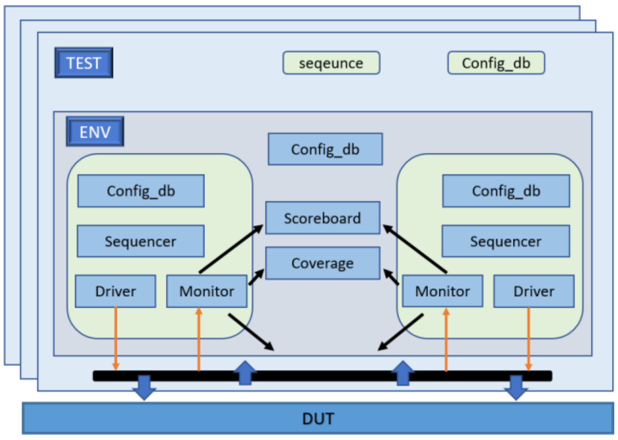

Fig 4. UVM Environment

UVM testbench consists of the following components and objects:

• Test: The class derived from uvm_test will represent a test case that would check the various features of DUT (design-under-test).

• Environment: The uvm_env would act as a model and should control the test environment. It consists of all verification components and objects created in the test. It is connected to DUT through a virtual interface.

• Agent: An uvm_agent will encapsulate everything which is needed to generate the stimulus and monitor any logical connection with DUT. It is the wrapper intended for the driver, monitor, and sequencer. They have their functionality meaning they can be configured to be active or passive.

• Driver: The uvm_driver will pull the transactions from the sequencer and drive them to the DUT via a physical interface.

• Sequence Item: It represents the basic user-defined transactions within a sequence. When sequences are executed, they generate one or more sequence items that the sequencer passes to the driver’s boundary. Typically, uvm_sequence is derived from uvm_sequence_item which generates transactions that are required to drive to the DUT.

• Sequencer: The uvm_sequencer will randomize sequence items and send them to the driver via TLM exports. They can be used for both sequencers and virtual sequencers.

• Monitor: Uvm_monitor is used to detect transactions on a physical interface and to make those transactions available to other parts of the testbench through an analysis port.

• Interface: The interface would help the testbench to communicate with the DUT.

• Scoreboard: The uvm_scoreboard will compare the expected values and the actual values of various inputs of the DUT signals. It would check if any transactions appeared or not from the DUT outputs.

• Coverage: The coverage information is collected from the monitor via the analysis port. This component ensures that all tested and untested scenarios of the design are covered based on the design specification.

2. Objectives

Design the Watchdog Timer with APB Interface using the Verilog Hardware Description Language and Verify the design using the Hardware Verification Language - System Verilog based UVM. The Verification components developed can be reused similar IP having APB Interfaces. Generate Stimulus to cover all the functionality of the design.

3. Methods

The DUT – Watchdog Timer with APB Interface is designed with one top Module which encapsulates Watchdog Timer Module and APB Interface Module. The Timer Value for the Watchdog timer is configurable via the APB Interface.

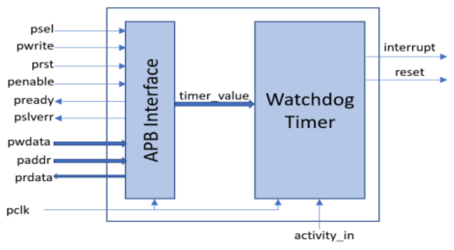

Fig 5. DUT – Watchdog timer with APB Interface

Watchdog timer takes the input of timer_value and activity_in and gives interrupt and reset outputs. The watchdog timer runs on the pclk, and the timer in the watchdog timer block is reset with every activity_in. If there is no activity_in before the timer times out (timer_value) the interrupt and reset are set which signals the system is not working as expected as there is no activity_in.

Testbench

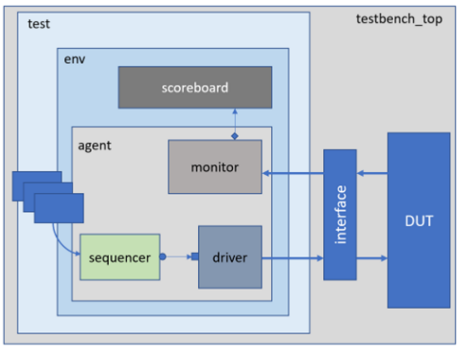

Fig 6. Block Diagram of UVM Testbench with DUT

The testbench is developed using UVM methodology.

The UVM Components – UVM Environment, UVM Agent, UVM Driver, UVM Sequencer, UVM Test, UVM Sequence, UVM Monitor and UVM Scoreboard which can be reused for similar APB Interface Module are developed to verify the Design.

Sequences are developed to generate 1. Random Stimulus, 2. APB Read and 3. Write Functionalities and 4. Watchdog Timer Functionalities. The Design under Test and Testbench are simulated using Synopsys VCS Simulator provided by EDA Playground.

4. Results

In this verification process, UVM is used to verify the environmental test components of the system. Watchdog timer with APB interface is designed and verification is done using UVM verification methodology. The components contain UVM ENVironment, UVM Agent, UVM Driver, UVM Sequencer, UVM test, and the main test cases. By internet sources, the EDA tool is used to design and verify the protocol. During verification, we must verify all APB READ and WRITE architectures and the Watchdog Timer Functionality.

a. Random Verification

All the signal inputs are randomized using the System Verilog Random constructs. The write happens only when the PSEL, PENABLE, and PWRITE are High and the Read happens when the PSEL, and PENABLE are High and PWRITE is Low

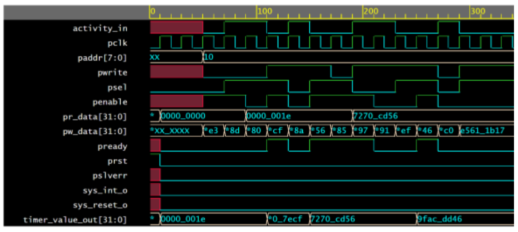

Fig 7. Random Simulation

b. APB Write and Read Verification

The test is written to generate only the write accesses and only the Address and Data are randomized. The register is written only when its address is provided in the address lines.

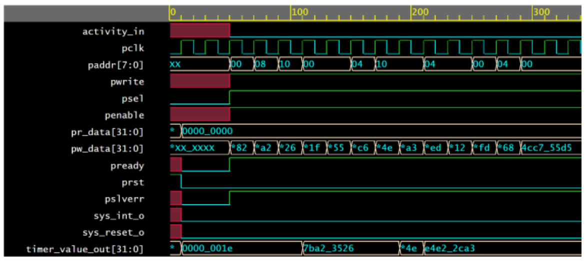

Fig 8. APB Write Simulation

The test is written to generate only the read accesses and only the Address and Data are randomized. The register is Read only when its address is provided in the address lines.

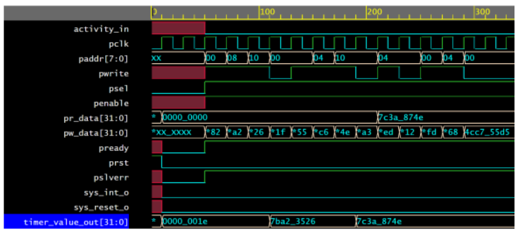

Fig 9. APB Read Simulation

c. Watchdog Timer Functionality Verification

The test is written where activity_in is delayed for the verification of the Watchdog timer functionality. Due to delayed activity_in, the timer expires, and Interrupt and Reset is asserted.

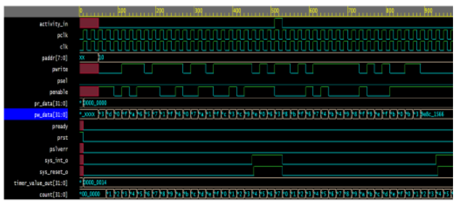

Fig 10. Watchdog Timer Functional Simulation.

5. Discussion

In this paper, successfully designed the Watchdog Timer with APB interface by using Verilog and generated the simulation. Design is verified by creating a test bench verification environment using UVM Verification Methodology and its components like interface, environment, driver, agent, sequencer, sequence, sequence item, etc., and DUV were implemented with help of Classes. Finally, the Synopsys VCS simulator is used to simulate the Design and Testbench. The main advantages of this method of verification are using the virtual random coverage driven, for which the verification engineer takes minimal time for verification in the complex design system. The tests and Verification components can be reused in a different design which uses APB Interface.

Refrences

[1]. ARM, “AMBA Specification Overview”, http://www.arm.com/. .

[2]. APB Example-AMBA system, Technical reference manual, ARM Inc.,1999.

[3]. Akhilesh Kumar, Richa Sinha, “Design and Verification analysis of APB3 Protocol with Coverage,” IJAET, Nov 2011.

[4]. SanthiPriyaSarekokku, K. Rajasekhar, “Design and Implementation of APB Bridge based on AMBA AXI 4.0,” IJERT, Vol.1, Issue 9, Nov 2012.

[5]. VERILOG Reference Manual, http://www.accellera.com

[6]. Samir Palnitkar, “Verilog HDL: A Guide to Digital Design and Synthesis (2nd Edition), Pearson, 2008.

[7]. C. Spear, SystemVerilog for Verification, Second Edition: A Guide to Learning the Testbench Language Features, 2nd ed. Springer Publishing Company, Incorporated, 2008.

[8]. Accellera, UVM 1.1 Class Reference, 2011.

[9]. J. Bromley " If SystemVerilog Is So Good, Why Do We Need the UVM? Sharing Responsibilities between Libraries and the Core Language" FDL, 2013.

[10]. Murphy, Niall. “Watchdog Timers,” Embedded Systems Programming, November 2000, p. 112.

[11]. Santic, John S. “Watchdog Timer Techniques,” Embedded Systems Programming, April 1995, p. 58.

[12]. UVM Verification Testbench Example https://www.chipverify.com/uvm/uvm-verification-testbench-example