International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 07, July-2022

Verification of Universal Asynchronous Receiver and Transmitter (UART) using System Verilog

M Srinath
Department of Electronics and Communication Engineering
RV College of Engineering
Bengaluru-59, India

Professor Sujatha Hiremath
Department of Electronics and Communication Engineering
RV College of Engineering
Bengaluru-59, India

Abstract—**This paper aims at designing a UART using SystemVerilog and verifying the same using a UVM based testbench. UART is one of the most frequently utilized serial communication protocol without the need for a clock excitation. The design includes baud rate generator, control, bus interface, interrupt control and the receiver-transmitter FIFO blocks. Parallel data from system bus are serialized and transmitted and the vice versa process is performed at the receiver end. The verification environment is a reusable and flexible one thereby reducing the complexity and time. Constrained randomization is adopted for random stimulus generation and a functional coverage of 100% and approximately 76.4% code coverage was achieved.**

Keywords— UART, UVM, constrained randomization, functional and code coverage, baud rate generator, FIFO, SystemVerilog

I. INTRODUCTION

UART (Universal Asynchronous Receiver Transmitter), is a standardized serial communication protocol which is primarily used to transfer data between systems in a serialized manner. Unlike I2C and SPI, UART is not only a protocol but also a built in block inside the controller that basically forms an integral part of the computer's peripheral interface. As the name suggests, UART does not require a clock excitation for data transfer and the speeds of data transfer can be configured based on the requirements of the customer. UART converts parallel data from system bus into a serialized data for transmission while the reverse process is performed at the receiver end. In order to have reliable, bug free designs with reduced complexity, enhanced efficiency and reduced turnover period the design of the UART is written using SystemVerilog HDL and verified using Universal Verification Methodology which also yields coverage metrics in addition to functional verification

II. DESIGN

The design of the UART majorly involves definition of the transmitter module, receiver module and the control logic along with the baud rate generator. The full duplex serial communication between two UART devices is depicted in Fig 1.

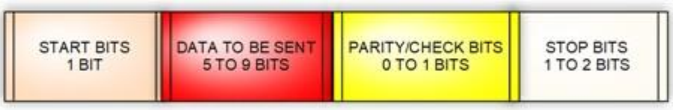

Fig. 1. UART Frame Format

The UART follows data transmission as per defined frame protocol in which the frame consists of a start bit, followed by the data that has to be transmitted, followed by the parity check bits which is particularly intended to detect any case of data corruption or packet loss (and is an optional field) and concluded by the stop bit. The transmitter first pulls down the start bit indicating that it is ready to send data which when sensed by the receiver shows that it is ready for receiving the data. The data and parity parts are then sent through the line post which the line is pulled high for two clock cycles indicating that it is done with the transfer (stop bits) and would initiate the transmission of succeeding frames and the same is interpreted at the receiver end. At The receiver, the start, stop and parity are implicitly removed after checking for errors in order to access the data and transfer it back into system bus in a parallel manner. The block diagram of the UART is shown as per Fig 2

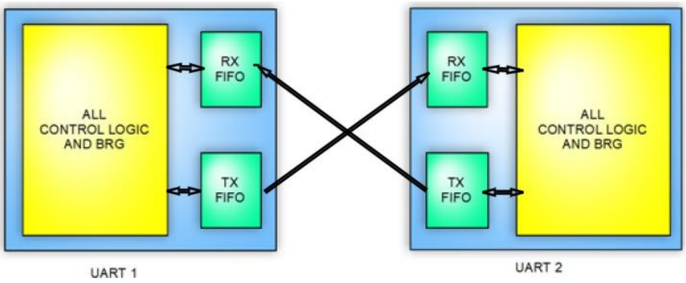

Fig. 2. Generic double UART Block Diagram

The synchronization of both the receiver and transmitter FIFOs are achieved by configuring them through the frequency dividers known as baud rate generators which defines the rate of sampling of data. A typical baud rate generator has a processor clock as its primary input, contains a clock divisor generating the baud rate for both the receiver and the transmitter. The generator also has option to use the same rate as that of the system clock which supports data transfer upto 115.2Kbps.

The transmitter and receiver follows the same architecture having a shift register, hold register and a FIFO block where in the transmitter the hold register receives data from the system bus while the shift register and FIFO serializes the data with framing and is transmitted once both the transmitter and receiver are ready for communication, while at the receiver the data received through the shift register and the data is converted back into a parallel format and put into the system bus using the buffer register and the FIFO module.

Different registers are utilized to oversee the operation of the UART module of which interrupt enable register (IER) is responsible for interrupt handling based on the status of the transmitter and receiver FIFOs which is indicated by the Interrupt Identification Register (IIR). The FIFOs are implicitly cleared after every trans-mission and reception by the FIFO control register while the serial communication line is handled by the line status and control registers.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 07, July-2022

III. Verification Methodology

Increasing demand for reliable systems with lower time to market has substantially increased complexity of verification process. Verilog was primarily used to verify designs but due to its low flexibility industries have moved to SystemVerilog based verification due to its reduced complexity, enhanced efficiency and increased reusability across designs. Much improved efficiency was achieved using Universal Verification Methodology (UVM) which is opted in this work.

A. Verification Environment

Building the UVM based testbench basically involves construction of the verification environment along with its built in components - driver, sequence_item, sequence and sequencer, driver, virtual interface, scoreboard, monitor and connecting it to the Design Under Test (DUT) which is the UART in this case. A typical verification environment using UVM is shown in Fig 3

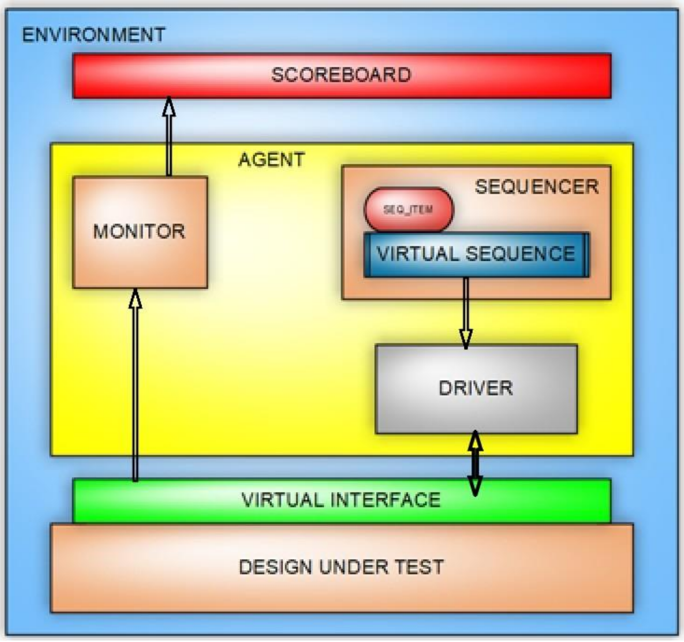

Fig. 3. Verification Environment

The environment is a part of the top module defined for functional verification and drives a sequence to instantiate the DUT through the virtual interface and can encapsulate one or more than one agents. Agent is the main block that encapsulates the transaction (sequence_item) which basically has all the I/O details and the sequencer. Agent also encapsulates the monitor which receives the values obtained from simulations runs of the DUT.

B. Test Cases

Constrained Randomization is utilized in order to generate the stimulus as it is proved as a better method in comparison to directed testing and also reduces the total verification turnover time. Randomization also aids in verification of corner cases which are typically difficult to verify. The sequence is the data that has to be transmitted through the sequencer to the driver which triggers the DUT through the virtual interface as defined by the protocol. The monitor receives the values obtained from the DUT and forwards it to the scoreboard which compares the actual simulated values with the golden signatures as per protocol definition. If the values obtained from DUT matches the golden signatures, the design is functionally correct else it indicates anomaly in protocol functioning

C. Simulation Metrics

In order to perform the functional verification various test cases are developed such as reset tests - to check if the software or hardware based resets clear the internal registers, interrupt tests - where the status of the transmitter and receiver FIFOs triggers the interrupt on non-readiness of communication, standard full duplex UART communication with different baud rates and FIFO depths and combination of both.

The functionality is successfully verified through the aid of SystemVerilog assertions (SVA) where the user can define checkers, coverage points and constraints. Through SVA in UVM, efficiency of verification is also determined as bugs are easily identified thereby improving the quality of building reliable systems

In order to measure the progress of the verification, coverage metrics is utilized namely functional coverage and code coverage. The progress is tracked by verifying if all the boundary conditions, value sets are covered during the simulation procedure. Ideally 100% code as well as functional coverage is expected from the simulation data which indicates every functionality is tested to its maximum possible efficiency.

IV. Results

The UART module was designed and successfully verified through functional verification using UVM and SystemVerilog. The major metrics and data drawn from simulations suggest the following observations

- Reduced complexity, increased reusability and efficiency
- Randomization helps to verify difficult-to-reach corner cases
- Reduced turnover time for verification
- 100% assertions, functional coverage achieved and ∼77% code coverage was achieved

Simulations were performed on Cadence Xcelium engine while the waveform debugging and analysis was performed on Cadence SimVision debug environment. The metrics derived from the simulations were thoroughly analyzed and extracted using Cadence Vmanager and IMC software. The simulation results obtained are illustrated in Fig 4 and 5

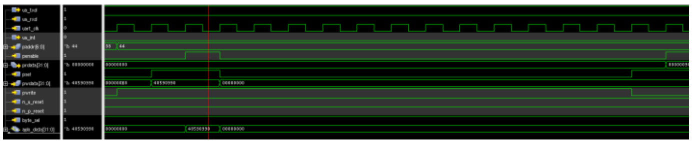

Fig. 4. Simulation Result - 1

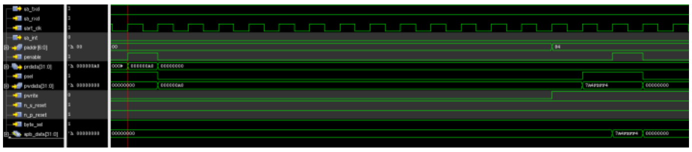

Fig. 5. Simulation Result - 2

The waveforms obtained from the simulations show that the data at the receiver is same as that of the transmitter. All the individual test cases were verified accordingly and were found to be compliant with the expected functionality including that of half duplex and full duplex mode communication. Table 1 illustrates the metrics derived from the simulations. The test pass percentage of 86% is majorly attributed to the operational constraints of the SoC of which UART module forms an important part.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 11 Issue 07, July-2022

SVA Assertions	100%
Functional Coverage	100%
Code Coverage	76.4%

V. Conclusion

Therefore the complete UART module was designed using SystemVerilog and functionally verified using UVM. The design involved a detailed description of baud rate generator module, control logic, register logic and the FIFO control at transmitter and receiver. Multiple test case scenarios were implemented to verify the block in a robust and efficient manner. Due to the utilization of UVM the testbench has reduced complexity and enhanced reusability. The boundary conditions were verified successfully owing to constrained randomization technique in the stimuli generation. The simulations yielded a functional coverage of 100%, overall code coverage of 76.4% and a test pass percentage of 86%

References

[1] D. D. Sharma, "Keynote 1: Compute Express Link (CXL) changing the game for Cloud Computing," in 2021 IEEE Symposium on High- Performance Interconnects (HOTI), IEEE, 2021, pp. xii–xii.

[2] B. Priyanka, M. Gokul, A. Nigitha, and J. Poomica, "Design of UART Using Verilog And Verifying Using UVM," in 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, vol. 1, 2021, pp. 1270–1273.

[3] R. K. Agrawal and V. R. Mishra, "The design of high speed UART," in 2013 IEEE Conference on Information & Communication Technologies, IEEE, 2013, pp. 388–390

[4] B. Panchal, Y. Parmar, and H. Suthar, "Implementation of AMBA based AHB2APB bridge," 2020

[5] D. Wang, J. Yan, and Y. Qiao, "Research on Chip Verification Technology Based on UVM," in 2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT), IEEE, 2021, pp. 117–120

[6] P. Gurha and R. Khandelwal, "Systemverilog assertion based verification of amba-ahb," in 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), IEEE, 2016, pp. 641–645

[7] . Badam Suresh, V. Srihari, and S. Sivanantham, "ASIC Implementation of Low Power Universal Asynchronous Receiver Transmitter," World Applied Sciences Journal, vol. 32, no. 3, pp. 472–477, 2014

[8] A. Fiergolski, "Simulation environment based on the Universal Verification Methodology," Journal of Instrumentation, vol. 12, no. 01, p. C01001, 2017

[9] M. F. Oliveira, C. Kuznik, H. M. Le, et al., "The system verification methodology for advanced TLM verification," in Proceedings of the eighth IEEE/ACM/I- FIP international conference on Hardware/software codesign and system synthesis, 2012, pp. 313–322.

[10] Y.-y. Fang and X.-j. Chen, "Design and simulation of UART serial communication module based on VHDL," in 2011 3rd International Workshop on Intelligent Systems and Applications, IEEE, 2011, pp. 1–4.

[11] A. Kaur and A. Kaur, "An approach for designing a universal asynchronous receiver transmitter (UART)," IJERA, vol. 2, no. 3, 2012

[12] S. Kamshette, S. A. Gama, and B. Patil, "VERIFICATION OF UART IP CORE USING UVM," ICCCA2015

[13] T. Yamuna and M. Dasu, "Systemverilog Implementation of UART with Single Error Correction and Double Error

[14]

[15]

[16]

[17] A. K. Gupta, A. Raman, N. Kumar, and R. Ranjan, "Design and implementation of high-speed universal asynchronous receiver and transmitter (UART)," in 2020 7th International Conference on Signal Processing and Inte- grated Networks (SPIN), IEEE, 2020, pp. 295–300

[18] N. F. Jusoh, A. Ibrahim, M. A. Haron, and F. Sulaiman, "An FPGA implementation of shift converter block technique on FIFO for UART," in 2011 IEEE International RF & Microwave Conference, IEEE, 2011, pp. 320–324

[19] P. Jain and S. Rao, "Design and Verification of Advanced Micro- controller Bus Architecture-Advanced Peripheral Bus (AMBA-APB) Protocol," in 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), IEEE, 2021, pp. 462–467.

[20] S. Van Doren, "HOTI 2019: compute express link," in 2019 IEEE Symposium on High-Performance Interconnects (HOTI), IEEE, 2019, pp. 18–18

[21] R. Neugebauer, G. Antichi, J. F. Zazo, Y. Audzevich, S. L oopez Buedo, and A. W. Moore, "Understanding PCIe performance for end host networking," in Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, 2018, pp. 327–341

[22] R. Gallo, M. Delvai, W. Elmenreich, and A. Steininger, "Revision and verifi- cation of an enhanced UART," in IEEE International Workshop on Factory Communication Systems, 2004. Proceedings., IEEE, 2004, pp. 315–318.

[23] T. Pavithran and R. Bhakthavatchalu, "UVM based testbench architecture for logic sub-system verification," in 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), IEEE, 2017, pp. 1– 5