International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

Design and Verification of UART using System Verilog

Yamini R, Ramya M V

Abstract- The main objective of this paper is to design and verify a full duplex UART module using System Verilog (SV). It is a serial communication protocol which provides communication between the systems without using clock signal. It converts parallel data into serial format and transmits the same. Once the data in serial format is received it is converted into parallel format. Designing of UART includes designing of baud rate generator, receiver, transmitter, interrupt and FIFO modules. Verification involves verifying the design by creating verification environment which allows to reuse the testbench and reduces the code complexity. Randomization is used to check the corner conditions which are hard to reach. 100% assertion and 100% functional coverage is achieved. UART operation is simulated using Questasim software.

Keywords-UART, SV, BRG, DUT, ASV

I. INTRODUCTION

Universal Asynchronous Receiver Transmitter – UART is a universal serial communication protocol that transmits data serially between systems. It is a computer hardware or a built in IC in microcontroller to control computer interface. UART can be used for both transmission and reception. Clock is not required for data transmission as it is asynchronous communication. The data format and transmission speed can be configured, hence the name Universal Asynchronous Receiver Transmitter. Most of the peripherals uses parallel data format for communication. UART receives data in parallel format converts it into serial data in transmitter section and sends it to the receiver. Receiver should convert the serial data to parallel format before sending it to the peripheral devices. In the fast development IC verification methodology improves the efficiency. SV is extended version of Verilog which includes more features than Verilog and reduces code complexity.

II. DESIGN

UART designing involves designing reception side, Baud Rate generator, transmitter side. To achieve full duplex communication between systems two UART modules are designed as shown in Fig 1.

A. Data Format

Transmitter frames the received parallel data into serial format by adding start bit, stop bit, parity bit. Start bit (logical 0) is added as prefix to data. Data is followed by parity bit (optional) which is used for error checking. Data frame ends with stop bit (logic 1). On the reception side, receiver removes start, parity and stop bits and then converts the serial data into parallel data.

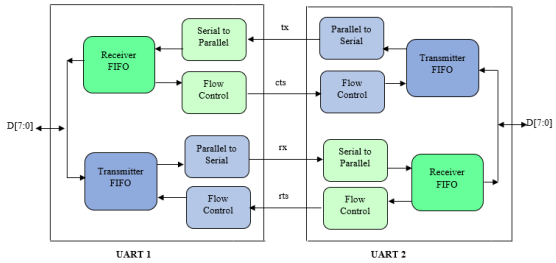

Fig 1: Overview block diagram

B. Wishbone Interface

Is a standard computer bus interface which allows parts of integrated circuits to communicate with each other. Used to overcome SOC integration problems by making the design faster.

C. BRG

BRG is a frequency divider. UART has a programmable baud generator as shown in Fig 2. BGR takes an input clock from the processor clock generator, which is divided by a divisor stored in divisor latch to produce a baud clock (BCLK). Baud clock is sixteen times the baud rate i.e. data lasts for 16 baud clock cycles.

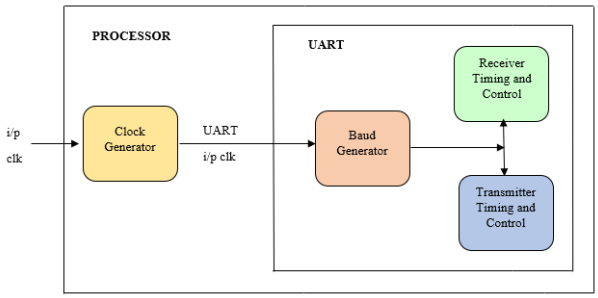

Fig 2: Baud Rate Generator

D. UART Transmitter

It consists of shift register, FIFO module, hold register. Function of transmitter is to frame the parallel data in serial format. Transmitter hold register (THR) receives data from internal data bus. Transmitter FIFO and Transmitter shift register (TSR) is used to serialize the parallel data and frames it by adding parity bit, start bit and stop bit. Through UARTn_TXD pin serial data frame is transmitted.

E. UART Receiver

Manuscript Received on June, 2020
* Correspondence Author
Yamini R*, MTech student, Electronics and communication department, JSSSTU, Mysuru, Karnataka, India. Email: yaminirece.95@gmail.com
Ramya M V, Assistant professor, Electronics and communication department, JSSSTU, Mysuru, Karnataka, India. Email: ramyamv@sjce.ac.in

Similar to transmitter section UART Receiver also has shift register, FIFO module. Along with that a buffer register is used. The data frame is received at UARTn_RXD pin which is stored in Receiver shift register (RSR).

Before de-framing the data receiver checks the parity bit for error and removes the start and stop bits. FIFO module and Receiver buffer register serial data is converted to parallel format.

F. Registers

To control UART operation different registers are used. These registers are listed in Table 1.

Registers DLL, RBR, THR share same address. When DLAB bit in LCR is 1, any modification in the shared address alters DLL content. When DLAB bit in LCR is 0, reading from the shared address reads RBR, and writing modifies THR.

Table 1: UART Registers

Register	Description
RBR	Receiver Buffer Register
THR	Transmitter Hold Register
IER	Interrupt Enable Register
IIR	Interrupt Identification Register
FCR	FIFO Control Register
LCR	Line Control Register
LSR	Line Status Register
DLL	Divisor LSB Latch
DLH	Divisor MSB Latch

Every interrupt is enabled/disabled by IER and the interrupts enabled by IER is forwarded to CPU. All the interrupts enabled by IER is indicated by IIR. IIR is a read only register, FCR is a write only register and both share same address. Reading from this shared address gives IIR content and writing allows to modify FCR. FCR enables FIFO and also used to clear FIFO content. LCR controls the data format asynchronous communication. LSR provides status of data transfer to the CPU.

III. VERIFICATION METHODOLOGY

Fast development of IC design is increasing the complexity of verification. Verification is viewed as a different process from design, which has led to a huge development in the verification discipline. Verification industry is moving towards SV which is extended version of Verilog. SV is not just a hardware description language but is also hardware verification language. In digital systems, SV is a dominant language used for designing and verification. Using SV Verification environment is developed which increases the verification efficiency and reduces the complexity.

A. Verification Environment

Major focus of this work is to create verification environment to verify the design using SV. Verification environment is created as shown in Fig 3, to verify the functional correctness of the Design Under Test (DUT).

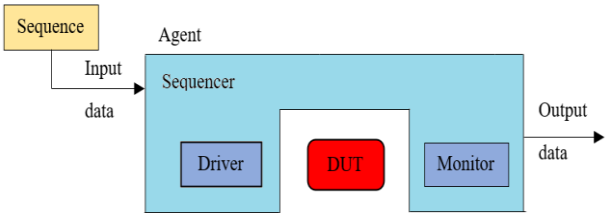

Fig 3: Verification environment

Verification environment drives a predefined sequence to the design and compares it with the obtained output. DUT represents the designed UART module. Environment can have one or more agents, which makes the testbench reusable. Two different DUTs represent two UART modules and each DUT has its own agent for verificatio. Agent acts as encapsulation of monitor, driver and sequencer. Two agents are creted to verify full duplex operation. Randomization is used which randomizes the sequence to test all the random conditions.

Randomize feature is used to generate random testcases that will check the corner cases of all the functions which are hard to reach. Random testing is more effective than other approaches as it automatically generates tests for verifying the design. Sequence is the data under transmission in and Sequencer is just a gateway between driver and sequence. Sequencer passes the randomized data from sequence to driver. Driver drives this data to DUT as per protocol. The commucication data is monitored by a monitor which gives the data to the scoreboard. Scoreboard compares the squence with the output of the DUT.

B. Test Cases

To verify functional correctness of the design testcases are developed. Different test cases are created to verify the half duplex and full duplex operation of the designed UART module and are verifed. The different test cases developed are shown in Fig 4. UART1 and UART2 are two UART modules designed. Tx1 and Rx1 represents transmitter and receiver of UART1 module respectively. Similarly Tx2 and Rx2 for UART2.

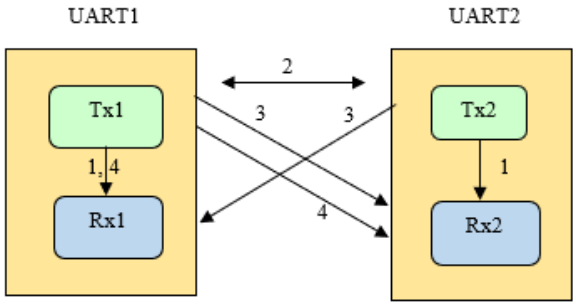

Fig 4: Different test cases developed

Following different test cases are developed and verified:

1. Data from transmitter of one UART is received by its own receiver to check each module separately.
2. Data is transmitted from the first UART and is received by the second UART module and vice

versa to verify half duplex operation.

3. Data is transmitted simultaneously from both the transmitters of UART1 and UART2, and are received by the receivers of UART2 and UART1 respectively.
4. Data from one UART is transmitted to both the receivers simultaneously.

C. Assertion

The behavior of the design is validated by using System Verilog Assertion (SVA). It is used to check whether the design is working correctly or not as per the requirements. It allows the user to define rules, checkers, constraints and cover points for the design. SVA also gives information about effieciency of the testcases. It is mainly used to check the working of different function in verification. SVA helps to reduce the verification time. During simulation simulator monitors the asstions and if the design violates an assertion error will be generated. Bugs can be easily identified using SVA which helps to improve the quality of the design.

D. Functional coveraagae

To measure the progress of the verification the term coverage is used. The test plan execution is observed by functioal coverage. It generates the progress report by collecting the simulation information. It tracks whether the boundary conditions, important set of values and other features are covered completely. It is an important factor in testing to know what set of values or features have been covered by the tests performed. Using the progress report the tets cases is modified or new test cases is added to increase the efficiency of the design. In a DUT the coverage points for the functions is defined by the user. 100% coverage indicates all the functions in the test plan are tested proprly.

IV. RESULTS

UART module is designed and verified by compiling and simulating the same by creating verification environment using SV. SV has many advantages over other languages and has helped to improve the design as listed below:

a. Using SV complexity of coding is reduced.
b. In verification environment testbench is reused by using agent to verify DUTs.
c. Randomize feature is used to check all the hard to reach corner cases.
d. Verification process time is reduced by using assertions.
e. 100% assertion is achieved.
f. 100% functional coverage is achieved.
g. Using SV high quality design is developed and verified effectively.

Simulation is done using Questasim tool and the results are shown below. Fig 5 shows full duplex transmission under randomized condition and simulation result for the same is shown in Fig 6. Data stored in sequence is randmozied in sequencer and is driven to DUTs. In Fig 5, first line urt1 represents the transmiter of UART1 which transmits the randomized data to receiver of UART2 given in second line as urt2. Similarly urt2 represents transmitter of UART2 and urt1 represents receiver of UART1.

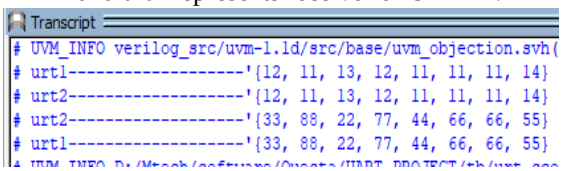

Figure 5: Result of Fulduplex mode

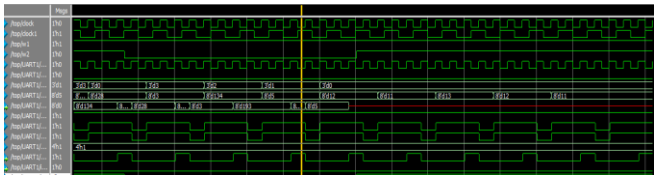

Figure 6(a): Simulation results of UART1

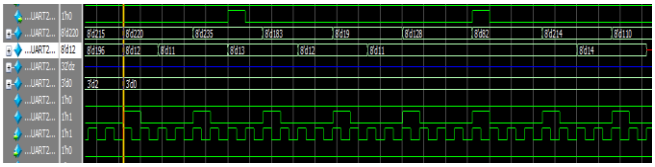

Figure 6(b): Simulation results of UART2

The received data is same as transmitted data. Thus, full duplex mode is verified. Simulation of UART1 and UART2 in full duplex mode are as shown in Fig 6(a) and 6(b) respectively. All the test cases are verifed including individual modules testing and half duplex mode opeartion of the designed UARTs as defined in test cases of section III.

Fig 7 shows the SVA report, where 7(a) represents assertion report of test package. Assertion missed is zero and target hit is 17. Similarly in 7(b), assertion missed is zero and target hit is 8. Thus, by acheiveing a 100% assertion graph.

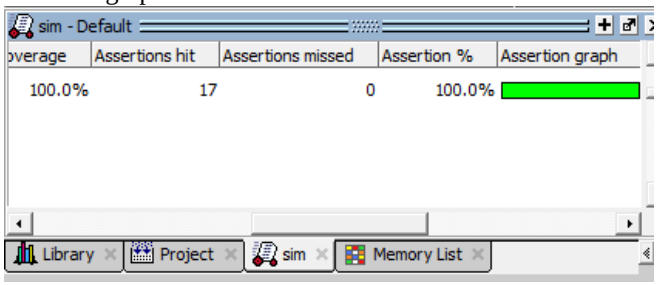

Fig 7(a): Assertion repot of testpackage

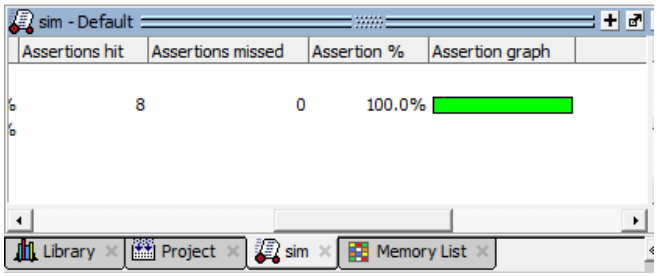

Fig 7(b): Assertion report of testcases

Functional coverage report is shown in Fig 8. 100% Coverage is obtained using random testing and SVA. Overall functional coverage of the test package is shown in 8(a) and 8(b) gives the detailed report of functional coverage of different modules. Thus, all the functions are verified completely and the design is made more efficient.

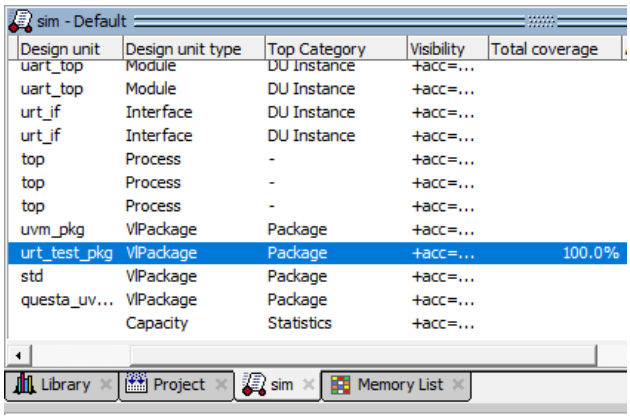

Fig 8(a): Functional coverage report

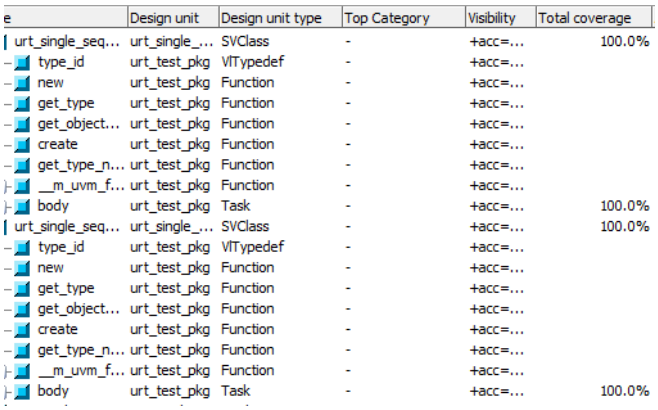

Fig 8(b): Coverage report of different modules

V. CONCLUSION

Full duplex UART is designed and verified effectively using SV. Design part involves designing of all the submodules of transmitter and receiver sections, BRG, registers. In verification, different test cases are used to verify the working operation of the design in half duplex mode and full duplex mode. By using SV design reuse method is applied which also reduces the complexity of writing the code. The corner cases are verifed by using randomization in the verification and verification time is reduced by using assertion. The design is simulated using Questasim software. 100% asserion graph and 100% functional coverage is achieved.

REFERENCES

1. Wei Ni, Xiaotian Wang, “Functional Coverage-Driven UVM-based UART IP Verification”, IEEE 11th International Conference on ASIC (ASICON), 21 July 2016.
2. Kumari Amrita, Avantika Kumari, “Design And Verification Of Uart Using Verilog Hdl”, International conference on Recent innovations in Management, Engineering, Science and Technology, RIMEST, 2018.
3. Spear, Chris, Tumbush, Greg, “SystemVerilog for Verification” Book.
4. Doron Bustan, Dmitry Korchemny, Erik Seligman, Jin Yang, “SystemVerilog Assertions: Past, Present, and Future SVA Standardization Experience”, IEEE Design & Test of Computers, Volume: 29 , Issue: 2 , April 2012.
5. Renduchinthala H H S S Prasad, Ch. Santhi Rani, ”UART IP Core Verification by using UVM”, International Journal of Industrial Electronics and Electrical Engineering, ISSN: 2347-6982, Volume-4, Issue-7, Jul.-2016
6. Mahat, Nennie Farina. "Design of a 9-bit UART module based on Verilog HDL." Semiconductor Electronics (ICSE), 2012 10th IEEE International Conference on 2012.
7. Ambika, Prof. Anuradha S, “High Speed UART Design Using Verilog”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 2, February 2016.

AUTHORS PROFILE

Yamini R, Final year MTech student, Industrial Electronics branch, Department of Electronics and Communication, JSS Science and Technology University, Mysuru. Working as Project Trainee Intern in L&T, Mysuru.

Ramya M V, working as Assistant Professor, Department of Electronics and Communication, JSS Science and Technology University, Mysuru. Has work experience of 8 years in teaching field. Area of specialization Embedded Systems, IoT and Automation.