VERIFICATION ENVIRONMENT FOR I2C MASTER PROTOCOL

S.Anitha[1], Ch.Hima Bindu[2], B.Hemanth Kumar[3], U.Ajay Krishna[4] , I.V.Nagaraju[5], R.Gopi[6] ,M.Pavan Kalyan[7],
Assistant Professor[1], Professor & HOD ECE[2], B.Tech Students[3,4,5,6,7]
Department Of Electronics & Communication Engineering,
QIS College Of Engineering And Technology, Ongole.

Abstract: This paper mainly concentrates on the use of Inter Integrated Circuit (I2c) by using the SYSTEM VERILOG and explains the construction of an environment which is used for verification and implementation of various test cases for this I2C Master Protocol. The trending techniques in fabrication unit showed a path in logical functioning of silicon die which making the verification process as most challenging one. Today many methodologies are required for the verification itself because it consumes seventy percent of the entire design process. One of the testing and verifying methodologies like UVM was introduced to create an environment which is used for verification and without interfacing with DUT.

IndexTerms **- Inter Integrated Circuit (I2C), Verification environment, System Verilog, DUT.**

I. INTRODUCTION

1.1 INTER INTEGRATED CIRCUIT

Inter integrated circuit is bidirectional serial bus which is having two wires. In the applications of short distance communication among number of devices, this technique will be most helpful. I2C can be considered as a multi-master bus with true standards including detection of collision and an arbitration which prevents the corruption of the data corruption if more than one master tries to control the bus simultaneously.
The I2C interface defined in 3 transmission speeds.
a) Normal transmission speed 100 KBPS
b) Fast transmission speed 400 KBPS
c) High speed transmission speed 3.5 MBPS

While coming to operation 100 KBPS mode and 400 KBPS mode are supported directly but for the operation of 3.5 MBPS high speed some special IOs are needed.

1.2 FEATURES OF I2C

1. Compatible with Multi-Master operation.
2. Here is a possibility of usage of software programmable clock frequency.
3. Existence of software acknowledgement bit.
4. It supports two kinds of address modes.
    a) 7- bit address mode
    b) 10-bit address mode
5. Generation of start or stop or separated start or acknowledgement signals is possible.

1.3 WISHBONE INTERFACE

Forcing of input to an interface of master, that interface can be called as wishbone interface. We need to give input signal to master as it is not automated to take inputs on its own.

Verification consumes 50% to 70% of the effort of design cycle and is on the critical path in the design flow of multimillion gate ASICs, so verification became the main bottleneck in the design process. The functional verification bottleneck is an effect of raising the design abstraction level. For this verification test benches plays a vital role.

1.4 TEST BENCH

A Test Bench can mimic the environment in which the design will fit. It checks whether the RTL Implementation suits the design specifications or not. This Environment provides invalid as well as valid and expected and unexpected conditions to test the entire design. Different types of test benches are

- LINEAR TESTBENCH
- LINEAR RANDOM TESTBENCH
- A TESTBENCH WHICH IS BASED ON FILE IO
- A TESTBENCH WHICH IS BASED ON STATE MACHINE
- TASK BASED TESTBENCH
- SELF CHECKING TESTBENCHS

The entire paper is managed in four parts. The part2 explained about the method that to be proposed. Part 3 focused on experimental results in practical way. Finally, the paper is concluded with the future scope in part 4.

1.5 IMPORATNCE OF VERIFICATION

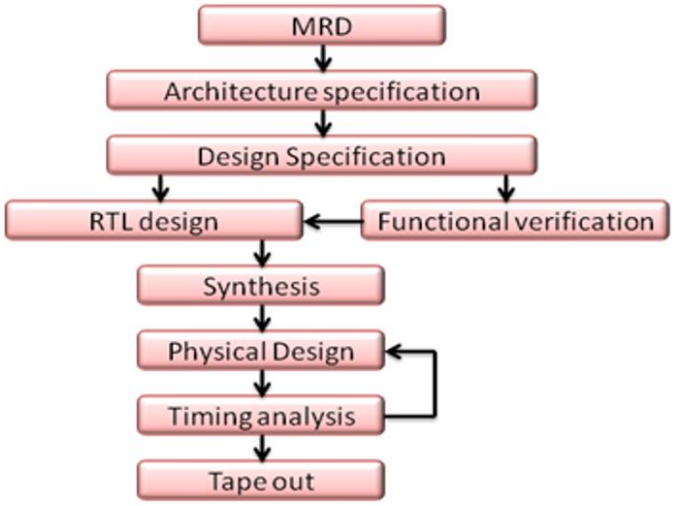


Fig1. IC Design Process

II. METHOD TO BE PROPOSED

2.1 VERIFICATION COMPONENTS

Verification Environment for I2C Master Protocol is proposed to verify the data transmission processing from master to slave is in correct manner or not.I2C is a bidirectional serial bus which is having two kinds of lines.

- SERIAL CLOCK LINE (SCL)
- SERIAL DATA LINE (SDA)

Where Serial Clock Line discuses about the clock pulse and SDA describes about the data, which we want to read or write to the slave from master. By using this we are able to find out the data transmission from to slave is occurring properly or not. To know about this in detail we need to know some verification components given below by going through this paper.

- **DESIGN UNDER TEST (DUT)**

The design what we have with us to test and verify will be called as DUT. Generally, his can be in the form of code or circuit. It is the design in which the specifications of the final product need to be meet. This is actually the RTL description in the designing aspect. It tells the functions and their features of a particular design flow.

- **SEQUENCER**

Sequencer is the place on which the sequences are able to run. To examine the DUT behavior, sequence of transactions has to be fulfilled. Sequencer able to run the stimulus generation code and sends sequence items down to driver whenever driver demands by driver.

- **MONITOR**

A monitor can be treated as passive element of the verification process. It scans the DUT signals coming on the interface, adds information as a packet and then sends it to coverage collector and for coverage information to the scoreboard.

- **DRIVER**

Driver can drive the signals of DUT. It is able to receive the items in sequence from the sequencer and able to put on the interface. It can be considered as the active part of the verification environment.

- **AGENT**

Agent is like a container which having a monitor, a driver and a sequencer. It has two modes of operation: passive and active. It scans the DUT signals in passive mode while it drives the signal to the DUT in active mode operation.

- **ENVIRONMENT**

It assembles the entire structure. It contains number of agents, scoreboard and some other components for measuring and checking by depending on design specifications.

- **SCOREBOARD**

It is a component to examine the output from the DUT against the expected response by comparing them to the model which we have with us as Reference Model. It tells how many times the response matched and how many times it mismatched.

- **TEST**

Test is the top-level of the component hierarchy. In SYSTEM VERILOG tests can be treated as classes that are derived from test class. It enables configuration of the test bench and verification components to test the dynamic behavior of the procedures by using the sequences.

❖ **SEQUENCES**

Sequences are procured from sequence items to construct a real set of inputs. This will create randomized transactions or a pre-determined set of transactions.

❖ **SEQUNCE ITEMS**

These are the main data objects that are passed at a level of abstraction between the components in verification.

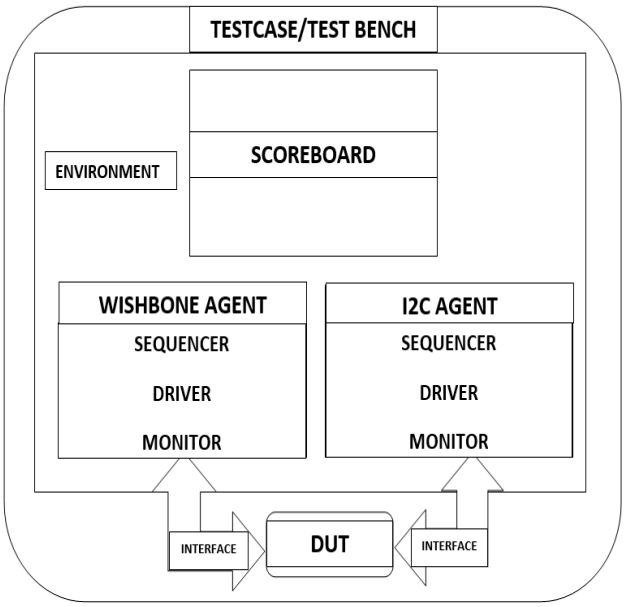


Fig2. Connection Diagram for I2C master protocol

2.2 BLOCK DIAGRAM

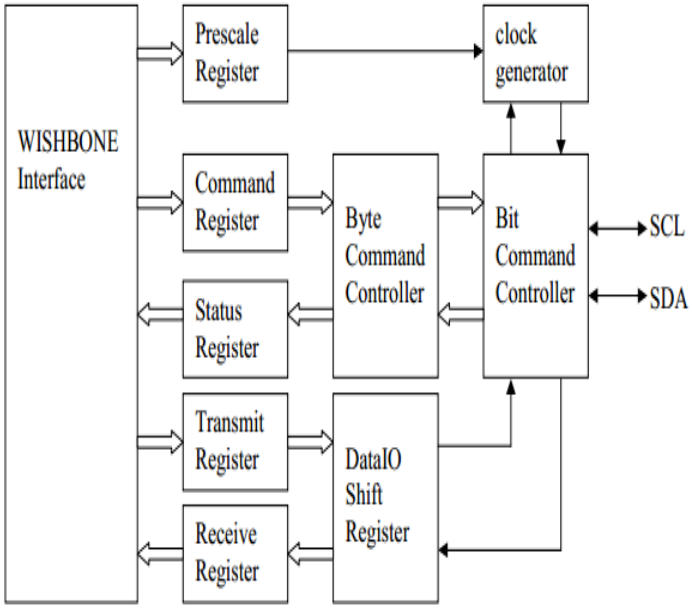
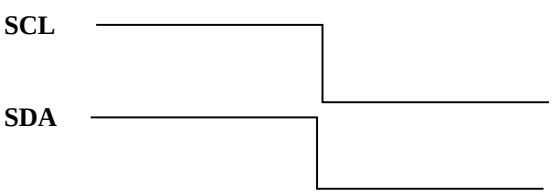


Fig3. BLOCK DIAGRAM

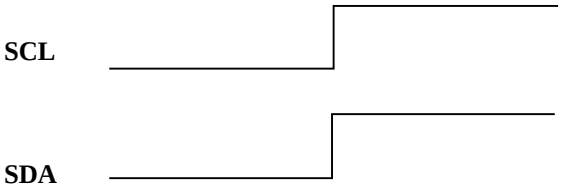
2.3 I2C COMMUNICATION PROCEDURE

Various control signals used for I2C master protocol are defined as follows.

a) **START SIGNAL:** When clock is moving from high to low the data also transmitted as high to low. This kind of transmission can be called as START signal generation.



b) **STOP SIGNAL:** When clock is moving from low to high the data also transmitted as low to high. This kind of transmission can be called as STOP signal generation.



c) **ACKNOWLEDGEMENT SIGNAL:** Which is used to give the acknowledgement on the transfer in progress status. If the transfer in progress, we need to wait till it should be complete. Whenever the transfer is completed then only the next operation going to takes place.

d) **DATA:** The that to be transmitted or received can be flow in this line only. The total operations on data can be done here only by the reference of clock line.

2.4 ALGORITHM:

The process of data transfer can be started from an IC called as master and the other IC which is addressed can be called as slave. Step by step procedure of I2C communication is as-

Step1. Check for the condition where the SCL & SDA are high. If not make sure that transfer is in progress. Wait till both lines become high.

Step2. Now the bus will be in Free State and the master is ready to generate the start condition.

Step3. Here the reference signal can be taken as SCL signal which provides clock signal to all ICs which are connected through bus.

Step4. The data on the SDA line can be treated as valid when the clock wire must be switches from low to high at that time**.**

Step5. The address on the slave can be put in the serial form on the data line. Here we are considering all hexa- decimal values.

Step6. Now remove MSB (Most Significant Bit) and add the LSB (Least Significant Bit) on the SDA line to know whether the operation is transmission or reception of data.

Step7. Now one-bit signal can be used to represent the acknowledgement bit to inform the master that whether the slave is ready or not for further operation.

Step8. Now put data serially based on the acknowledgement of master on SDA line. This data represents the memory location where the data to be stored on the slave address.

Step9. Now again one-bit data is generated as acknowledgement for the transmission of original data to that particular memory location.

Step10. Put data serially based on the acknowledgement and then stop condition is generated when all data received and then the bus will be free again.

By using these ten steps the data on I2C can be transmitted or received based on the acknowledgement bits and read/write bits.

2.5 FLOW CHART

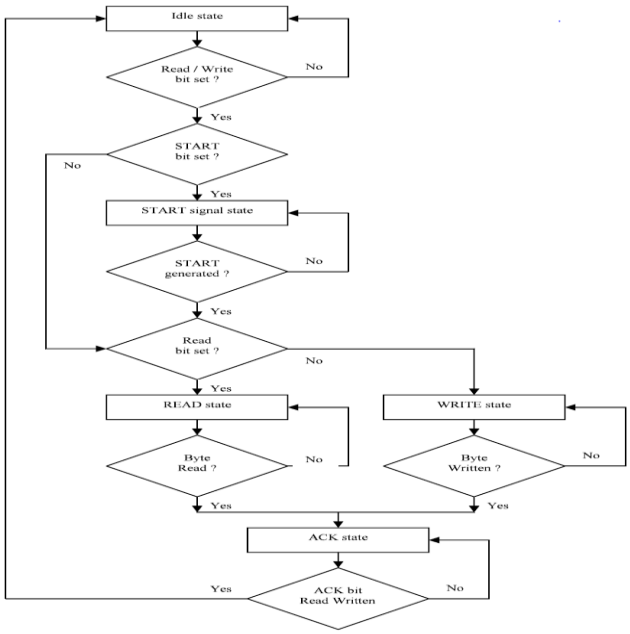


Fig: 3 Communication Procedure flow chart

III. EXPERIMENTAL RESULTS

The proposed method tested under different test cases like I2C read operation, I2C read & write operation.
For every test case the experimental procedure is identical. That is explained as follows.

I2C Sequence Format:

a) Generation of START command.
b) Writing of SLAVE ADDRESS + write bit.
c) Reception of ACKNOWLEDGE from slave.
d) Writing of DATA.
e) Receiving of acknowledgement from slave device.
f) Generation of STOP command.

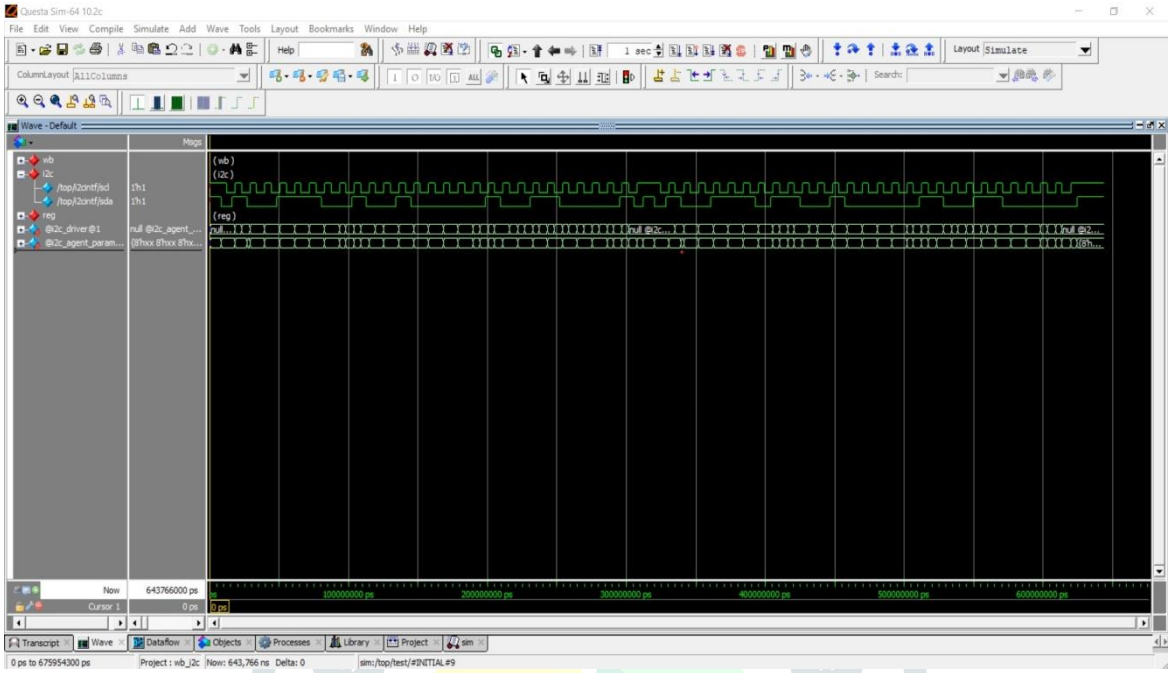


Fig:4 I2c Read & write operation

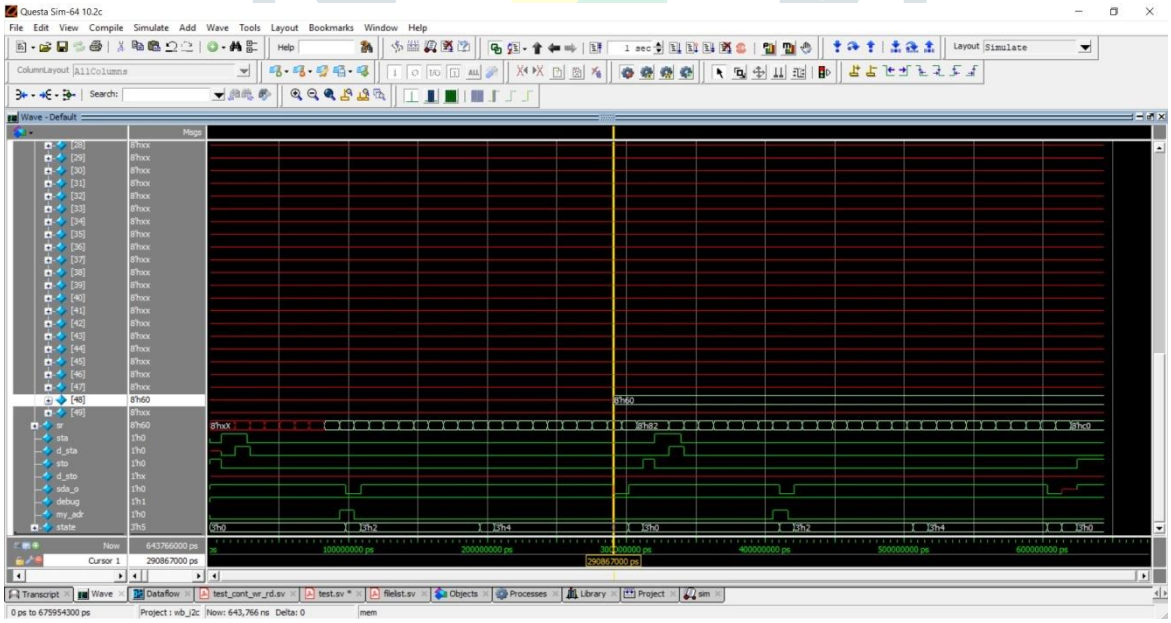


Fig:5 I2cADDRESS AND DATA ALLOCATION FOR READ & WRITE FOR TESTCASE 1

Here we are distinguishing the experimental values by varying the data and addresses for fig4 can be showed in fig5 and fig6 respectively.

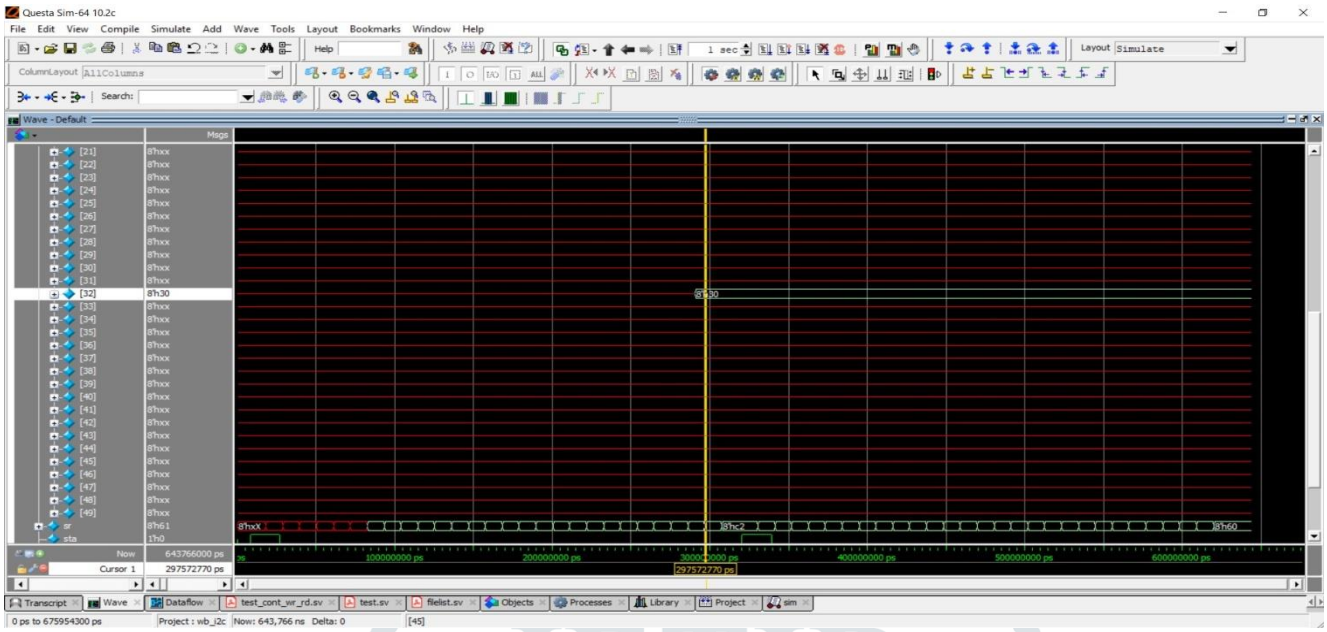


Fig:6 I2c ADDRESS AND DATA ALLOCATION FOR READ & WRITE FOR TESTCASE 2

In fig5 & fig6 the address values can be converted from hexadecimal to be displayed as decimal values and the address values are represented for different testcases.

ADVANTAGES

- More flexible.
- Usage of tasks and functions.
- More efficient.
- Performs many calculations.
- Focuses on single functionality.
- Faster approach compared to all.
- Introducing some self-checking techniques to reduce the burden of complex model checking.

IV. CONCLUSION

The total more than half percentage of the process of verification consumed by the design cycle and the design flow. So, the process of verification became main target in the process of design. By using this proposed method of verification environment for i2c master protocol we can get better results as compared to other existing verification methods. By this proposal the future scope in Self-checking test benches which may require more difficulty during the creation phases of initial testbenches. This method can highly reduce the effort needed to check again and again of a design after a change has occurred in the Design Under Test.

ACKNOWLEDGMENT

The authors would like to thank to the QIS, SNES for their ultimate support to execute and to make the project successful.

REFERENCES

[1] System Verilog for Verification – Chris Spear
[2] Writing testbenches using System Verilog – Janick Bergeron
[3] http://www.testbench.in/
[4] IEEE Computer Society and the IEEE Standards Association Corporate Advisory Group, "IEEE Standard for SystemVerilog - Unified Hardware Design, Specification, and Verification Language", IEEE Std 1800TM-2012, 21 February 2013
[5] Chris Spear, "SystemVerilog for Verification", Second Edition, 2008.
[6] Accellera Organization, "Universal Verification Methodology (UVM) 1.1 Class Reference", June 2011
[7] Datasheet for Microchip 24LC256 – 2K I2C Serial EEPROM.
[8] R. Herveille. I2C-Master Core Specification, Rev. 0.9, 2003
[9] Clifford E. Cummings, Tom Fitzpatrick, "OVM & UVM Techniques for Terminating Tests" DVCon 2011
[10] Mentor Graphics® Verification Academy "Cookbook Online Methodology Documentation"
[11] IEEE Computer Society "IEEE Standard Verilog® Hardware Description Language", IEEEStd 1364TM- 2005, 07 April 2006.
[12] Janick Bergeron, "Writing Testbenches: Functional Verification of HDL Models", Springer US, 28-Feb- 2003
[13] Philips Semiconductor, "I2C-bus specification and user manual", Rev. 6 - 4 April 2014.