DESIGN AND VERIFICATION OF I2C PROTOCOL USING UVM

Mr. Santosh Bujari[1], Ms. Bhagya F Radder[2], Ms. Pooja Akki[3], Ms. Pratika Yaligar[4] , Ms. Priya B Kitli[5]

[1] *Assoc. Prof., Dept. of Electronics and Communication Engineering SKSVMACET, Laxmeshwar, Karnataka, India*
[2] *Final Year Student Dept. of Electronics and Communication Engineering SKSVMACET, Laxmeshwar, Karnataka, India*
[3] *Final Year Student Dept. of Electronics and Communication Engineering SKSVMACET, Laxmeshwar, Karnataka, India*
[4] *Final Year Student Dept. of Electronics and Communication Engineering SKSVMACET, Laxmeshwar, Karnataka, India*
[5] *Final Year Student Dept. of Electronics and Communication Engineering SKSVMACET, Laxmeshwar, Karnataka, India*

ABSTRACT

The I2C protocol was given by the Philips Semiconductors in order to allow faster devices to communicate with slower devices and also allow devices to communicate with each other over a serial data bus without data loss. I2C plays an important role as an interface in communication between devices. Electrically Erasable Programmable Read Only Memory (EEPROM), Analog to Digital Converter (ADC), Digital to Analog Converter (DAC) and Real Time Clock (RTC) requires an interface for communication and I2C is used as an interface between them. The serial communication reduces the cost of connection and the number of IC pins. I2C is one of the serial wired communication protocols. Transmitting information over I2C will improve the performance of the system. As the design becomes complex, verifying the functionality of the design by using traditional test benches will be difficult. Thus the hardware verification languages are used for designing the modules. The verification environment is composed of different verification components which are used for different purpose such as simulating, checking and collecting functional coverage. More time and more resources are required for verification compared to design. The System Verilog is a hardware description language as well as hardware verification language. It is more helpful for the verification process. Object Oriented Programming (OOP) technique is applied in verification environment of System Verilog. A verification environment may be prepared using System Verilog without using any particular methodology but that will be different for every variation of the design. There are various verification methodologies out of which Universal Verification Methodology (UVM) is widely preferred by the verification industry worldwide, as the verification environment created using UVM is reusable, efficient and well structured.

Keyword: *I2C, UVM, Verification methodology, and Verification environment.*

1. INTRODUCTION

The Inter-Integrated Circuit (I2C) bus from Philips Company is a bi-directional bus which supports multi-masters and multi-slaves. SDA (Serial data line) and SCL (Serial clock line) are two lines used in I2C bus. I2C provides support for communication with various devices. It also provides various other benefits like on-chip bus interface, synchronization and software based addressing and data transfer, reusability of modules and easy fault diagnosis to ensure no data loss. There is a decrease in size of I2C, so there is no need for large number of pin connection needed for transfer of data.

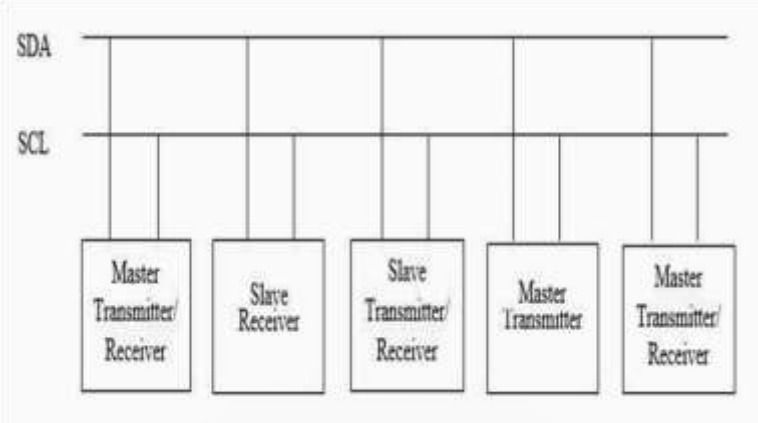

Fig-1: I^2C Bus Configuration

USB/SPI/Microwire and mostly UARTS are all just "one point to one point" data transfer bus systems. They use multiplexing of the data path and forwarding of messages to service multiple devices. To overcome this problem, the I2C protocol was introduced. It requires only two lines for communication with two or more chips and can control a network of device chips with just a two general purpose I/O pins whereas, other bus protocols require more pins and signals to connect devices. The I²C (Inter-Integrated IC) bus developed by Philips company, is a simple bidirectional serial bus that supports multiple masters and slaves.

2. I2C SPECIFICATION

I2C provides chip-to-chip serial communication with the help of two lines in an interface. A master is a device which is used to initiate a data transfer and also responsible for generating clock signal. At that time, the device which is addressed will be considered to be a slave. In I2C bus master is always used to initiate the transaction. Transmission of data will take place through SDA. SCL is responsible for the communication. Clock and reset are used to initiate the bus controller process. The read/write signal is used as an input signal for transmission. If the read/write signal is high, the master is writing the data and if the read/write signal is low, the data will be read by the master from the slave.

2.1 I2C Terminology

Transmitter: This is the device which is used to transmit data to the bus.
Receiver: This is used to receive data from the bus.
Master: This is used to generate clock signal and to start the communication.
Slave: This is the device that is addressed by the master.

2.2 Start and Stop Conditions

All communication begins with a START signal and can be finished by a STOP signal. In Idle condition, both the SDA and SCL are high. A high to low transition on the SDA line, while SCL is high, is the start condition. A low to high transition on the SDA line, while SCL line is high, is the stop condition. The start and the stop signal condition is shown in Figure 2.

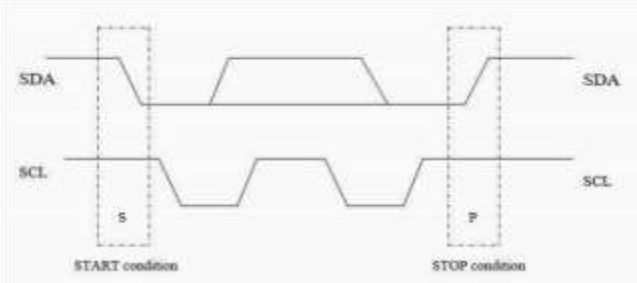

Fig-2: Start and Stop condition

2.3 Acknowledgement

In I²C an ACK needs to be sent by Receiver shown in Fig.3. After transmitting the address byte, master pulls the SDA line in high impedance state, thus allowing the slave to respond with an acknowledgement which it does by pulling the SDA line low. After releasing the SDA line master generates the clock to read the ACK bit. If ACK is received master can proceed with read/write bit operation. If the transmitted address does not match any of these slaves none of the slaves would pull down the SDA and the line remains high. Master interprets this as NACK. If NACK is received master can issue STOP command or reissue the START command.

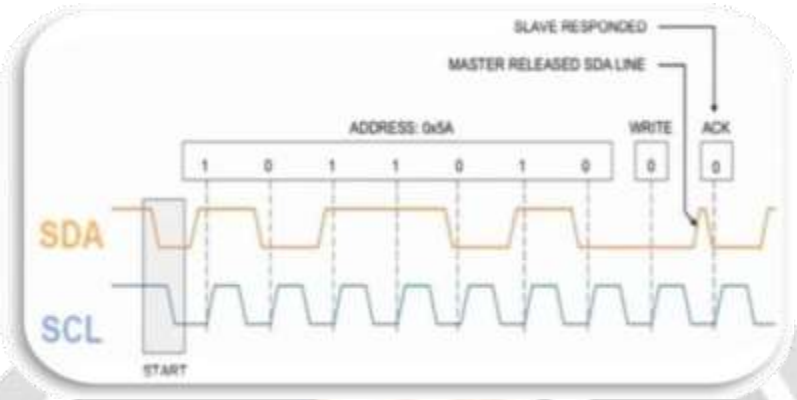

Fig-3: Acknowledge Report of I^2C protocol

2.4 Frame format

The frame format of the I^2C protocol is shown in Fig. 4. In normal state both lines (SDA and SCL) are high.

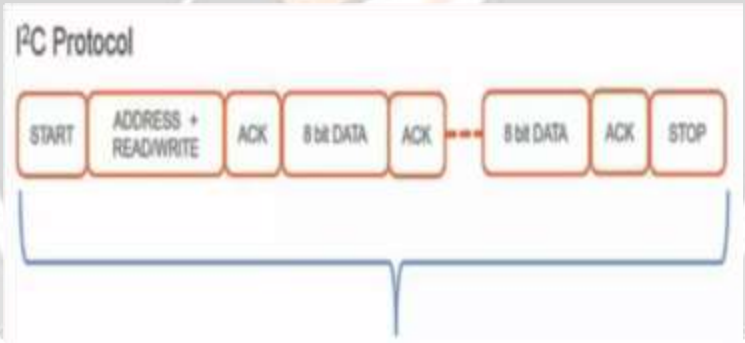

Fig-4: Data Communication Flow

The master will initiate the communication with the start condition followed by 7-bit address of the slave device. The read/write bit is followed by the acknowledgment bit and 8-bit data. The master will terminate the data by sending stop condition.

3. METHODOLOGY

Transaction: It is a class Defines the pin level activity generated by agent (to drive to DUT through the driver) or the activity has to be observed by agent (Placeholder for the activity monitored by the monitor on DUT signals).

Generator: Is a class Generates the stimulus (create and randomize the transaction class) and send it to Driver.

Driver: Is a class Receives the stimulus (transaction) from a generator and drives the packet level data inside the transaction into pin level (to DUT).

Monitor: Is a class Observes pin level activity on interface signals and converts into packet level which is sent to the components such as scoreboard

Agent: An agent is a container class, which groups the class's (generator, driver, and monitor) specific to an interface or protocol.

Scoreboard: Is a class Receives data items from monitors and compares them with expected values. Expected values can be either golden reference values or generated from the reference model.

Environment: The environment is a container class for grouping higher level components like agent's and scoreboard.

Test: is a program it is responsible for,

- Configuring the testbench.
- Initiate the testbench components construction process.
- Initiate the stimulus driving.

Test bench top: This is the topmost file, which connects the DUT and Test Bench. It consists of DUT, Test and interface instances, the interface connects the DUT and Test Bench.

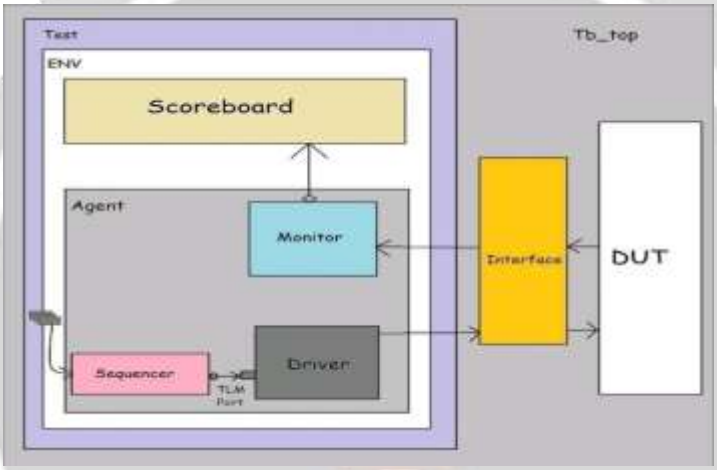

Fig-5: Methodology

4. WORKING

Master initiates the data transfer by sending START signal. The Start signal is followed by the slave address. Then read/write bit which acts as a control bit to decide the direction of data transfer. Master releases the bus and waits for the acknowledgement (ACK) bit from the slave. The desired slave pulls the SDA line LOW to send ACK signal. Depending on R/W bit value either master or slave sends the 8-bit data and acts as the transmitter. The other device acts as a receiver and sends ACK after receiving 8-bit data by pulling SDA LOW. After (N) ACK the master ends the transmission by sending STOP signal. Any master can now initiate a new data transfer by taking the control of bus.

5. ADVANTAGES AND DISADVANTAGES OF I2C PROTOCOL

Advantages

- Requires only 2 lines for communication.
- Supports multiple masters & slaves.
- Chip addressing.

- Simple than UART.
- Designing I2C using hdl reduces time of manufacturing.

Disadvantages

- Slow data rate.
- Requires more space.
- Conflicts may arise because of chip addressing.

6. RESULTS

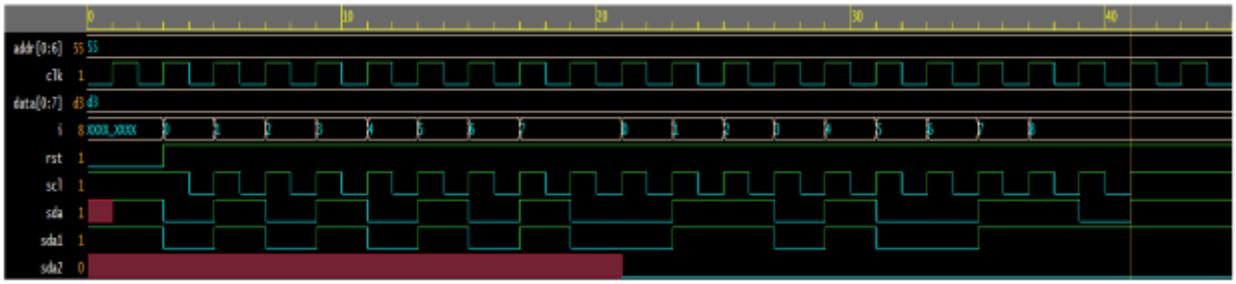

Fig-6: Slave simulation result

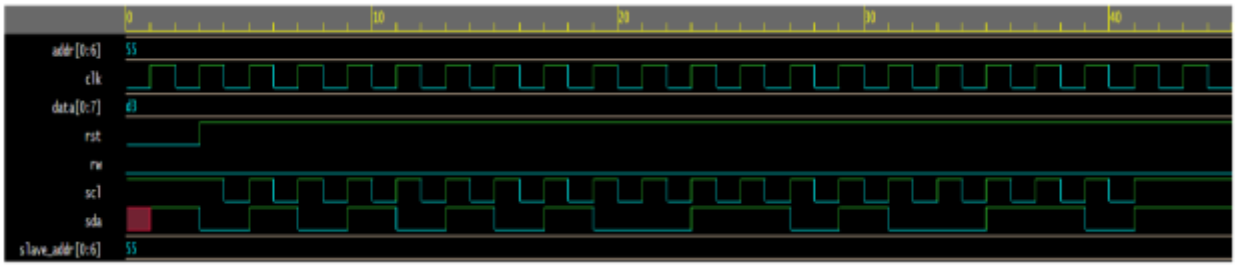

Fig-7: Master simulation result

CONCLUSION:

The I2C Master Controller bus was successfully designed using Verilog and Simulated. The Verification Environment was created using uvm. The proposed verification environment comprised of interface, generator, monitor, driver and scoreboard were implemented by using OOP concepts. The functionality of the design is verified by the uvm. The constrained randomization technique is applied for the design and verification environment.

7. REFERENCES

[1]. M.Sukhanya and K. Gavaskar, "Functional verification environment for I2C master controller using system verilog" 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN) pp, 1-6 .
[2]. I2C Bus specification, Philips semiconductor, version 2.1, January 2000.
[3]. Sireesha Bhuvanagiri , Shaik Khasim Beebi, "Design and development of I2C Protocol using Verilog".
[4]. https://verificationguide.com
[5]. Shruthi S Mugali and Dr.Meghana Kulkarni ,"An Efficient Design & Verification of I2C Master Core," 2018 International Journal of Computer and Mobile Computing pp,1-6, June 2018.
[6]. https://www.chipverify.com/