

Design & Simulation of I2C Master Slave Communication Protocol Using Verilog HDL

Keval¹, Dr. Sharanbasappa Shetkar²

^{1,2}Lingaraj Appa Engineering College, Bidar, Karnataka

Abstract—The Inter-Integrated Circuit protocol widely known as I²C protocol is being used in serial communication protocol for connecting digital devices using two lines: SDA and SCL. This research designs and simulates an I²C Master-Slave communication system using Verilog HDL. The master system used to indicate communication and as it also controls data transfer, while the slave responds to commands and exchanges data. This design implements key I²C features such as START/STOP conditions, address transmission, read/write operations, and ACK/NACK signaling. The system is designed at the Register Transfer Level (RTL) for accurate control and timing representation. A Verilog testbench has been developed to verify communication functionality and protocol compliance. Simulations are performed making use of ModelSim and Xilinx Vivado to analyze timing and waveform behavior. Hence the results demonstrate reliable data transfer and correct implementation of the I²C protocol.

Index Terms—I²C, Master-Slave communication system, SDA and SCL, Register Transfer Level (RTL), ModelSim

I. INTRODUCTION

Serial communication is a method of transmitting data between electronic devices and it transmits data one bit at a time. Hence it is widely used in embedded systems and digital electronics for reliable data transfer. Unlike parallel communication, it requires fewer wires, reducing complex hardware required and cost. Communication protocols provide a set of rules for exchanging data between devices. These protocols ensure proper synchronization, addressing, and error-free communication. Serial communication can be classified as synchronous or asynchronous. Synchronous communication uses a shared clock signal, while asynchronous communication uses start and stop bits. The most commonly used serial protocols include UART, SPI, and I²C. Among these,

I²C is a popular synchronous communication protocol developed by Philips Semiconductor. It uses only two communication lines: SDA (Serial Data Line) and SCL (Serial Clock Line). Here the master device controls communication, while slave devices respond based on their addresses. I²C supports both read and write operations with acknowledgment signals. It allows multiple devices to communicate on the same bus using only two wires. There by reduces wiring requirements and simplifies system design. Hence it is simplified and highly efficient and due to this, I²C is widely used in sensors, EEPROMs, RTCs, microcontrollers, and FPGA-based applications. I²C protocol is supported by nearly all major semiconductor manufacturers and is integrated into microcontrollers, processors, FPGA devices, sensors, and memory chips. Its widespread adoption demonstrates its importance in modern electronic system design and embedded communication architectures.

Features of I2C Communication

1. Two-wire communication (SDA and SCL).
2. Supports multi-master and multi-slave communication.
3. Uses 7-bit and 10-bit addressing schemes.
4. Synchronous data transfer using a clock signal.
5. ACK/NACK mechanism for reliable communication.
6. START and STOP conditions for communication control.
7. Arbitration support to avoid data collisions.
8. Clock synchronization between devices.
9. Open-drain bus architecture.
10. Supports multiple speed modes (100 kbps, 400 kbps, 1 Mbps, and 3.4 Mbps).
11. Low hardware complexity and reduced wiring.
12. Easy integration with sensors, memory devices, and microcontroller

Objectives:

1. Design an I²C Master-Slave communication system using Verilog HDL.
2. Implement master and slave modules for data communication.
3. Generate START and STOP conditions according to the I²C protocol.
4. Support read and write data transfer operations.
5. Implement ACK/NACK signaling for reliable communication.
6. Verify functionality through simulation and waveform analysis.
7. Analyze timing behavior and protocol compliance.
8. Gain practical knowledge of RTL design and hardware verification.
9. Provide a foundation for FPGA and SoC implementation.

II. METHODOLOGY

The methodology of this project begins with studying the I²C protocol including SDA, SCL, addressing,

START/STOP conditions and its communication principles. The I²C Master-Slave architecture is designed using Verilog HDL in order to generate clock signals and initiate communication and control data transfer. This is done using master and slave system. The master module is developed to generate the clock signal and initiate communication. A slave module is designed to respond based on address matching. START and STOP conditions are implemented to control data transfer. An FSM is also used to manage different states of communication such as idle, address, data, and acknowledgment. The master and slave modules are integrated together over the SDA and SCL lines to form the I²C bus. A Verilog testbench is created to apply input test cases. The overall design is simulated using ModelSim or Xilinx Vivado. Finally, waveform analysis is performed to verify the functionality of the designed system and protocol compliance.

III. SYSTEM DESIGN

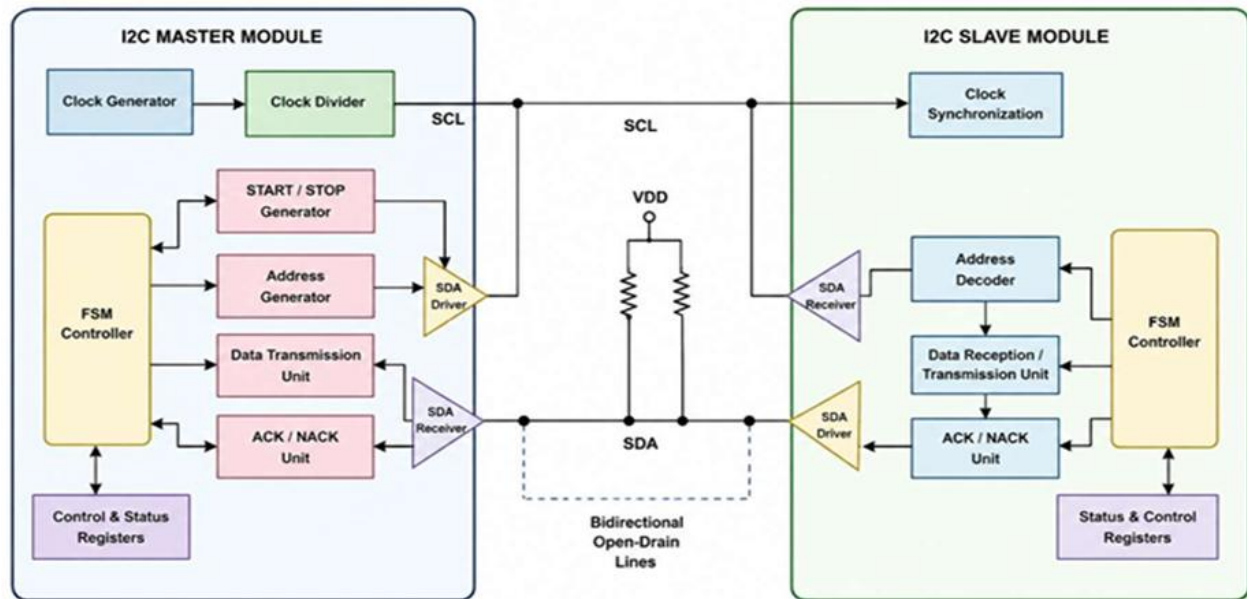


Fig 1: System design

The proposed I²C Master-Slave communication system is designed using Verilog HDL for reliable data transfer serially. The architecture thus designed is based on the standard I²C protocol using two lines: SDA and SCL. This system consists of a master module, slave module, clock generator, FSM controller, address decoder, and control logic. The

master system is used to initiate communication and it also controls the entire data transfer process there by generating the SCL clock signal for synchronization. Then comes the slave system, the slave responds to the master based on address matching. Hence the Communication begins with a START condition and ends with a STOP condition. First the master system

sends the slave address along with a read/write bit. The slave continuously monitors the bus for valid address detection. If the address sent by the master matches, the slave sends an ACK signal. The system supports both read and write operations. During write mode, the master sends data to the slave. During read mode, the slave sends data to the master. Data is transferred serially through the SDA line. Here Shift registers are also used. The shift registers are used for serial-to-parallel and parallel-to-serial conversion. The FSM controls all communication states such as IDLE, START, ADDRESS, DATA, ACK, and STOP. The clock generator ensures proper timing for synchronization by generating the clock signals. The address decoder identifies the correct slave device. ACK/NACK logic confirms whether the data transmission took place successfully. Open-drain configuration is used for safe bus sharing. In order to maintain the logic HIGH on the bus Pull-up resistors used. Multiple slave devices can be connected to the same bus. This system supports multi-device communication using unique addresses. Timing synchronization ensures data integrity and avoids errors. This modular design improves scalability and reuse. The architecture is implemented using RTL modeling in Verilog HDL. Simulation is performed using ModelSim and Vivado. At last waveform analysis is done the waveform are used to verify correct functionality of the designed system. The waveforms confirm proper timing between SDA and SCL signals and validate protocol compliance. Performance analysis shows that the system achieves low communication latency, stable clock synchronization, and optimized FPGA resource utilization. The modular RTL design improves scalability and reduces propagation delay. Reliability analysis confirms correct behavior under normal and abnormal conditions such as invalid address detection and missing acknowledgment handling. Overall, the architecture provides an efficient, scalable, and reliable I²C communication system suitable for FPGA implementation in embedded systems, IoT devices, and digital communication applications.

IV. FUTURE SCOPE

This I²C Master-Slave communication system can be improved for higher-speed operation using optimized clock and pipelined FSM designs. Multi-master

support can be added with proper arbitration for advanced communication systems. The design can also be expanded in order to support multiple slave devices with dynamic addressing. We can also enhance the system performance by integrating it with other protocols like SPI, UART, or CAN which can improve system flexibility and functionality. AI-based techniques can be used for fault detection and communication optimization. Low-power methods such as clock gating can reduce energy consumption in IoT applications. It may act as major advantage in the security system by adding security features like encryption and authentication for safe communication. FPGA and SoC implementation can also improve performance and scalability. Wireless communication modules like Wi-Fi or Bluetooth can enable remote applications. Overall, the system can be upgraded for industrial and real-time embedded applications.

V. CONCLUSION

This project successfully implements an I²C Master-Slave communication system using Verilog HDL. It demonstrates reliable serial data transfer between master and slave devices. The design used is based on RTL methodology with modular components. Key modules include master, slave, FSM, clock generator, and address decoder. The master system controls communication by generating clock and START/STOP signals. The slave responds based on address matching and acknowledgment signals. FSM ensures proper sequencing of all communication operations. This system supports synchronized data transfer using SDA and SCL lines. Simulation confirms correct functionality of all I²C operations. Waveform analysis verifies proper timing and signal synchronization. Performance analysis shows low latency and stable operation. This design is optimized for FPGA implementation with efficient resource usage. The proposed system also handles error conditions like invalid addresses and missing ACK. Overall, it provides a reliable and scalable I²C communication solution for embedded systems.

REFERENCES

- [1] S. Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, 3rd ed. Noida, India: Pearson Education, 2023.

- [2] M. D. Ciletti, *Advanced Digital Design with the Verilog HDL*. Noida, India: Pearson Education, 2022.
- [3] D. Harris and S. Harris, *Digital Design and Computer Architecture*, 3rd ed. Burlington, MA, USA: Morgan Kaufmann, 2021.
- [4] P. Chu, *FPGA Prototyping by Verilog Examples*. Hoboken, NJ, USA: Wiley, 2021.
- [5] J. Bhasker, *A Verilog HDL Primer*. Hyderabad, India: BS Publications, 2022.