Implementation of Multi Master Multi Slave I2C Protocol

[1]Gandam Sravan Kumar, [2]Dr. S. Poongodi

[1]M.Tech Scholar, Department of Electronics and Communication Engineering, CMR Engineering College, Hyderabad, Telangana, India.

[2]Professor, Department of Electronics and Communication Engineering, CMR Engineering College, Hyderabad, Telangana, India.

Abstract: The Inter-Integrated Circuit (I^2C) protocol is a widely adopted synchronous, multi-master, multi-slave serial communication standard designed to enable efficient and flexible data transfer between integrated circuits or peripherals in embedded systems. It operates through two bidirectional lines Serial Data Line (SDA) and Serial Clock Line (SCL) which allow multiple devices to share a common bus with minimal wiring, making it well-suited for applications such as sensor interfacing, industrial automation and resource limited environments. This paper focuses on creating a Universal Verification Methodology (UVM) based verification environment for the I2C protocol, specifically configured with a single master and four slaves two functioning as input devices and two as output devices. The verification environment employs UVM components like drivers, monitors and sequencers to model the master and slave agents, enabling the simulation and validation of fundamental I2C read and write operations. Rather than using assertions and coverage metrics, the approach highlights systematic test case generation and the implementation of scoreboards for data integrity checks and bus interaction validation. Supporting transaction-level modeling and reusable test benches, the environment ensures efficient debugging and functional verification of designs handling both data acquisition and distribution over the shared I2C bus. This methodology demonstrates the practical application of UVM principles in a targeted functional setup, establishing a solid base for future scalability and advanced verification techniques.

Keywords: Inter-Integrated Circuit (I^2C), Serial Data Line (SDA), Serial Clock Line (SCL), Universal Verification Methodology (UVM), Future Scalability, Advanced Verification Techniques.

I. INTRODUCTION

The Inter-Integrated Circuit (I^2C) protocol is a widely used serial communication standard designed for short-distance data transfer between integrated circuits on a circuit board. Developed by Philips Semiconductor (now NXP Semiconductors) in the early 1980s, I^2C was introduced as a simple and cost-effective solution to reduce wiring complexity in consumer electronics such as televisions. By enabling microcontrollers to interface seamlessly with peripherals like sensors, memory devices, displays, and real-time clocks, I^2C quickly

became an essential communication protocol in embedded systems.

I²C communication is based on just two bidirectional lines: the Serial Data Line (SDA) for transferring data and the Serial Clock Line (SCL) for clock synchronisation. These lines, supported by pull-up resistors, allow multiple devices to share a common bus, supporting both single-master and multi-master configurations. Each device is assigned a unique address, enabling selective and organised communication over the same two wires, thereby significantly reducing hardware complexity compared to parallel buses.

Functionally, I²C operates in a synchronous, half-duplex mode where the master device controls the clock signal and initiates all communication. Transactions begin with a start condition, followed by the address of the target slave, data transfer, and a stop condition. Every byte exchanged is acknowledged, which enhances reliability. The open-drain (or open-collector) implementation of SDA and SCL ensures safe bus sharing, as devices can only pull the line low, with pull-up resistors restoring it high when idle. This design prevents signal conflicts and allows multiple devices to exist together without interference.

Over time, I²C has evolved to meet the growing demands of embedded applications. Initially supporting only 100 kHz speed with 7-bit addressing, the protocol expanded in 1992 to include fast modes up to 400 kHz and 10-bit addressing, increasing bus capacity to over 1000 devices. Today, higher-performance derivatives such as SMBus and I3C have been developed, providing extended functionality while maintaining backward compatibility. Despite faster alternatives, I²C remains widely preferred in scenarios where low data rates, minimal wiring, and cost efficiency are more critical than speed.

Due to its simplicity, scalability and flexibility, the I²C bus has become one of the most popular communication standards in modern electronics. It is extensively used in sensor interfacing, memory integration, and microcontroller-based embedded systems, particularly in industrial control and IoT applications. Its efficient architecture, low pin count and robust design have made I²C a long-standing and reliable choice for board-level communication, ensuring its continued relevance in digital systems today.

II. LITERATURE SURVEY

M. Trehan, P. Kumar and N. Gaur et al. In this research paper, The Multi-Protocol Conversion Unit (MPCU) is designed and simulated using Hardware Descriptive Language (HDL). This unit acts like a bridge and can perform data communication between three different protocol sets which are among the most prevalent methods used for serial communication of data. The following protocols are Serial Peripheral Interface (SPI), Inter Integrated Circuit (I2C), and Universal Asynchronous Receiver Transmitter (UART). The designed conversion module will take input from any of the three different protocols along with a value of Conversion Select (COSE) input. In accordance with the input value of COSE, the data will be transferred from the data bus that is within the MPCU module to any one

of the three protocols slave. Which will in turn convert the received data and produce the 8-bit output data at the end. Thus, in terms of prototyping devices if there is a need for receiving data from any of the protocols and to send that data onto a slave based on different protocols there is no need for the data to be processed by the microprocessor or microcontroller. It is possible when data is received by the MPCU module and sent to the receiving slave just by changing the value of COSE. The simulation of this MPCU shows that conversion is feasible between masters of the three protocols and their slaves, and it has a wide range of applications in research and development as well as device prototyping devices. Moreover, it helps the Microcontroller to do other tasks without making it process data for the conversions. This unit could be modified further to convert data received by slaves so that it can be sent to the other slaves based on different protocols connected to the MPCU [1].

Y. Chen, F. Xiao, L. Yu and P. Cui et al. In order to solve the problems of fixed speed, single-way conveying, conveying speed and loom speed mismatch of knitting machine elastic belt conveying device, a design scheme of multi-line elastic band conveying control system based on I2C (Inter-Integrated Circuit) communication is proposed. The scheme realizes the real-time communication between the control host and the PWM (Pulse Width Modulation) signal output slave through the I2C protocol communication of the master-slave MCU (Microcontroller Unit). The slave machine controls the stepper motor through the subdivision circuit and the drive circuit, and uses the stepping motor to speed up and down the differential. Algorithm and speed matching algorithm to realize multi-speed variable speed elastic belt conveyor control system with good speed matching. At the same time, because I2C protocol has easy scalability, it can realize multi-channel through bus analog communication and expansion between MCU I/O ports. delivery. Finally, by setting up simulation experiments, it is verified that the speed matching between the stepping motor and the knitting machine needle is better. When the stepping motor driver (TB6600) is 32 subdivided and the duty ratio of the PWM signal is 60%, the motor speed is stable, low vibration and noise. The system can play a positive role in improving fabric tightness, improving fabric production efficiency and adjusting fabric morphology [2].

Hardik Kaneriya, Santosh Jagtap et al. This paper implements serial data communication using I2C (Inter-Integrated Circuit) master/slave bus controller. The I2C master/slave bus controller was designed, which act as either master or slave as per the requirement. This module was designed in Verilog HDL and simulated and synthesized in Questasim 10.0c. I2C master initiates data transmission and in order of operation slave responds to it. It can be used to interface low speed peripherals like motherboard, embedded system, mobile phones, set top boxes, DVD, PDA's or other electronic devices [3].

Z. Meng, J. Deng, M. Zhang, S. Song, Z. Liu and J. Feng et al. I2C bus is a serial communication bus that is widely used in embedded system. It is based on SCL (Serial

Clock Line) and SDA (Serial Data Line), to support bi-directional communication between multiple interconnected devices in a master-slave structure. Based on the traditional characteristics and operating principle of I2C protocol, this paper has proposed an innovative I2C-Compatible design for slave based on asynchronous circuit, which is compatible with standard I2C protocol. It is characterized by switchable communication speed, supporting a maximum speed of 3.4 Mbps, ultra-low power and small hardware cost. To ensure reliable communication system, a Schmitt-RC filter and the structure of fine-tuning phase are adopted in our design. After RTL simulation and logic synthesis, the design is verified on FPGA platform communicated with MCU I2C master interface, and finish Placement and Route. The proposed design is fabricated on Dongbu 0.18μm CMOS process, and the test on chip level is carried out after tape-out. The results show that the I2C bus for slave has excellent performance, power and area (PPA) compared with traditional I2C bus, which is very suitable for end-side devices in the IOT and Smart City [4].

III. METHODOLOGY

The Universal Verification Methodology (UVM) is a standardized, SystemVerilog-based framework developed by Accellera Systems Initiative to address the challenges of verifying modern System-on-Chip (SoC) designs, which often involve high integration, multiple protocols and complex functional scenarios. Built on transaction-level modeling (TLM), constrained random stimulus generation and functional coverage, UVM enables efficient exploration of diverse test scenarios while promoting reusability through standardized components such as sequences, drivers, monitors, scoreboards, and agents. Its coverage-driven verification approach ensures thorough functional testing guided by coverage goals, reducing the risk of undetected bugs and shortening verification cycles, while also supporting assertion-based verification (ABV) for protocol compliance. Due to these strengths, UVM has become the industry-standard methodology for verifying complex digital systems, including widely used communication protocols such as I²C, SPI, AXI, and PCIe.

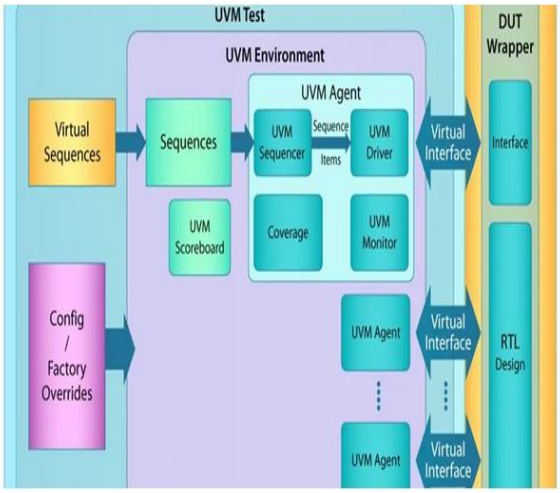

Fig. 1: UVM Architecture

UVM is a standardized approach used to verify digital designs, particularly complex ASICs and SoCs. Based on SystemVerilog, it offers a structured framework for developing reusable, scalable, and flexible testbenches. The methodology relies on a library of base classes that form the foundation of UVM testbenches, organized in a layered architecture.

The UVM test bench forms the backbone of the verification environment, containing the Design Under Test (DUT), connecting interfaces and reusable verification components. Communication is handled through Transaction-Level Modelling (TLM), enabling efficient transaction exchange. A UVM Test, instantiated dynamically at run time, serves as the top level component that configures parameters, invokes sequences and applies stimulus to the DUT. Base tests set up the environment, while extended tests modify it for specific scenarios with different sequences, coverage goals or configurations.

The UVM Environment groups together agents, scoreboards and sub-environments, making it modular and reusable. Agents manage a specific DUT interface and typically include a sequencer, driver and monitor. The sequencer controls stimulus flow, the driver converts transactions into pin-level signals and the monitor observes DUT activity and translates it back into transactions. Agents can operate in active mode to generate stimulus or passive mode to simply monitor DUT activity and may also include coverage collectors and protocol checkers.

Sequence Items and Sequences handle stimulus generation at the transaction level. A sequence item represents a data transaction with constraints, while sequences are ordered collections of these items. The sequencer delivers sequence items to the driver, which drives DUT signals, while the monitor captures outputs for further processing. Scoreboards then compare DUT results with expected outputs from reference models (written in System Verilog, C, C++, or SystemC) to ensure correctness. This separation of stimulus, monitoring, and checking improves structured and reusability.

Additional UVM elements strengthen flexibility and scalability. Virtual interfaces allow safe and modular DUT access, while virtual sequences coordinate activity across multiple agents for complex, system-level scenarios. TLM provides a high-level communication mechanism using ports, exports, and imps, supporting FIFO buffering and one-to-many broadcasting. Analysis ports extend this by forwarding transactions to multiple connected components, enabling efficient scoreboarding, logging, and coverage collection. Together, these features make UVM a powerful framework for verifying complex SoC and ASIC designs.

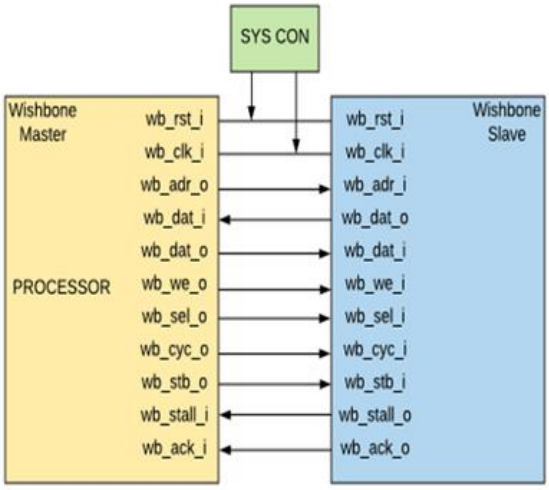

Fig. 2: Wishbone Interface

The WISHBONE System-on-Chip Interconnection Architecture provides a portable and flexible method for integrating semiconductor IP cores, simplifying SoC

design and promoting faster reuse. It defines MASTER interfaces, which initiate bus transactions, and SLAVE interfaces, which respond to them, with communication handled through an interconnection block (INTERCON). The INTERCON can be implemented as point-to-point, data flow, shared bus, or crossbar switch interconnections, while arbitration for bus access is managed through centralized or distributed arbiters.

Table 1: Wishbone Interface signals

Port	Width	Direction	Description
wb_clk_i	1	Input	Master clock
wb_rst_i	1	Input	Synchronous reset, active high
arst_i	1	Input	Asynchronous reset
wb_adr_i	3	Input	Lower address bits
wb_dat_i	8	Input	Data towards the core
wb_dat_o	8	Output	Data from the core
wb_we_i	1	Input	Write enable input
wb_stb_i	1	Input	Strobe signal/Core select input
wb_cyc_i	1	Input	Valid bus cycle input
wb_ack_o	1	Output	Bus cycle acknowledge output
wb_inta_o	1	Output	Interrupt signal output

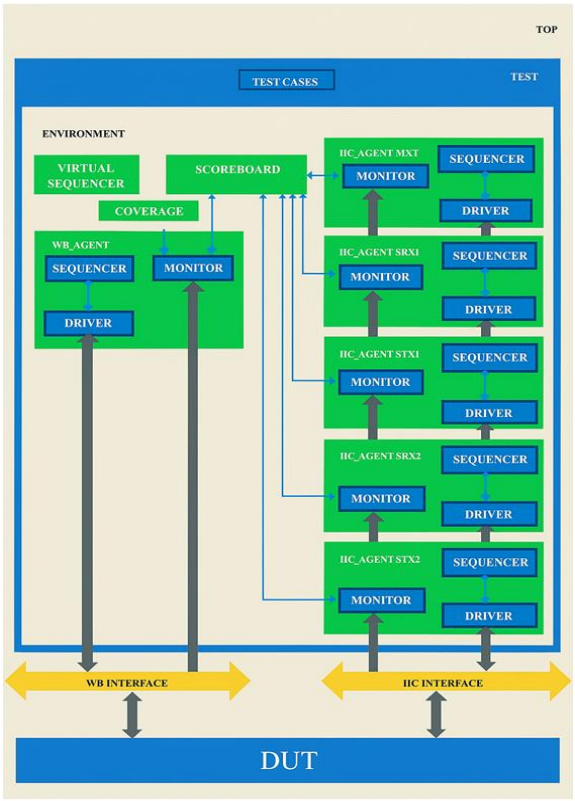

Fig. 3: Top Level I2C Verification Environment

The IIC master core UVC uses two active agents: the Wishbone agent and the IIC agent, both capable of sending transactions. The Wishbone agent drives transactions by configuring DUT registers, while its monitor samples register data and forwards Wishbone sequence items to the scoreboard. The IIC agent drives data on the SCL and SDA lines, with its monitor sampling this activity and sending IIC sequence items to the scoreboard. Finally, the scoreboard compares sequence items from both monitors to check for matches or mismatches.

IV. RESULTS

Wishbone master write and read operations shown in Figure 4, successful data transfer between the master and slave.

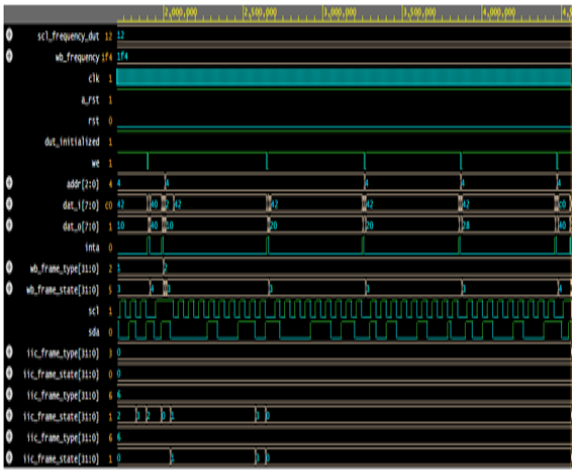

Fig. 4: Wishbone Master Write and Read Operation

The Figure 5 demonstrate correct IIC master write and read operations, with data successfully written to and read from the slave device. The presence of proper start, stop and acknowledgment signals confirms reliable communication.

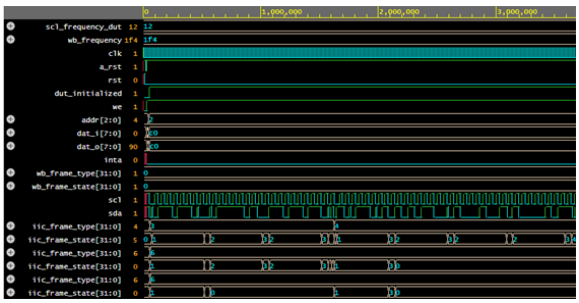

Fig. 5: IIC Master Write and Read Operation

V. CONCLUSION

In conclusion, the UVM-based verification environment for the I2C protocol successfully models a single master with multiple slaves, enabling thorough validation of fundamental read and write operations. By leveraging UVM components such as drivers, monitors, sequencers and scoreboards, the setup ensures reliable data integrity checks and efficient bus interaction analysis. The approach highlights systematic test case generation and reusable test benches, providing a scalable and practical framework for future advanced verification needs.

VI. REFERENCES

[1] M. Trehan, P. Kumar and N. Gaur, "Design and Analysis of Multi-Protocol Conversion Unit for SPI, I2C and UART," 2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT), Dehradun, India, 2024, pp. 324-329, doi: 10.1109/DICCT61038.2024.10533106.

[2] Y. Chen, F. Xiao, L. Yu and P. Cui, "Design of Multi-Line Elastic Belt Conveying Control System for Knitting Machine Based on I2C Protocol," in IEEE Access, vol. 8, pp. 51803-51809, 2020, doi: 10.1109/ACCESS.2020.2979627.

[3] Hardik Kaneriya, Santosh Jagtap, "Design Of I2C Master With Multiple Slave," in: International Journal on Recent and Innovation Trends in Computing and Communication, vol. 3, no. 5, pp. 2890-2893, May 2015.

[4] Z. Meng, J. Deng, M. Zhang, S. Song, Z. Liu and J. Feng, "Design and Implementation of Ultra-Low Power I2C-Compatible Bus based on Asynchronous Circuit," 2023 6th International Conference on Electronics Technology (ICET), Chengdu, China, 2023, pp. 79-84, doi: 10.1109/ICET58434.2023.10211915.

[5] S. Verma and A. Sharma, “Design and Verification of I2C Protocol using SystemVerilog,” International Journal of Advanced Research in Electronics and Communication Engineering, vol. 5, no. 3, pp. 320–324, 2016.

[6] P. Ramesh and M. Kumar, “UVM Based Verification of I2C Protocol with Assertion-Based Verification,” International Journal of Engineering Research & Technology, vol. 9, no. 8, pp. 570–574, 2020.

[7] B. Jose and J. S. Immanuel, “Design of BIST(Built-In-Self-Test)Embedded Master-Slave communication using SPI Protocol,” 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 2021, pp. 581–585.

[8] Y. Guo, “A SPI Interface Module Verification Method Based on UVM,” 2020 IEEE International Conference on Information Technology, Big Data and

Artificial Intelligence (ICIBA), Chongqing, China, 2020, pp. 1219–1223.

[9] L. M. Kappaganthu, M. D. Prakash and A. Yadlapati, “I2C protocol and its clock stretching verification using system verilog and UVM,” 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2017, pp. 478–480.

[10] L. Li, Y. Wang, X. Chen and X. Ren, “Research on Improvement of Configurable I2C controller IP Core,” 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China, 2023, pp. 484–489.

[11] K. Kumar, A. Kaur, S. N. Panda and B. Pandey, “Effect of Different Nano Meter Technology Based FPGA on Energy Efficient UART Design,” 2018 8th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2018, pp. 1–4.

[12] S. Harutyunyan, T. Kaplanyan, A. Kirakosyan and A. Momjyan, “Design And Verification Of Autoconfigurable UART Controller,” 2020 IEEE 40th International Conference on Electronics and Nanotechnology (ELNANO), Kyiv, Ukraine, 2020, pp. 347–350.

[13] A. Gupta, S. Soni, S. K. Muthukumar, D. J. Kurian and V. Parvatha, “Low Power, Light Weight and Transparent Multi-Chip Communication Link for IoT Segment,” 2021 IEEE 18th India Council International Conference (INDICON), Guwahati, India, 2021, pp. 1–6, doi: 10.1109/INDICON52576.2021.9691750.

[14] Akopyan F, Sawada J, Cassidy A, et al. “TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 34 (10): 1537–1557.

[15] C. Liu, Q. Meng, T. Liao, X. Bao and C. Xu, “A Flexible Hardware Architecture for Slave Device of I2C Bus,” 2019 International Conference on Electronic Engineering and Informatics (EEI), Nanjing, China, 2019, pp. 309–313, doi: 10.1109/EEI48997.2019.00074.