

Verification of I3C Slave Controller

Project report submitted in partial fulfilment of the requirement for the Award of the Degree of

MASTER OF TECHNOLOGY

in VLSI DESIGN

Submitted By

Inderpreet Singh

602362012

Under Supervision of

Dr. Sudhanshu Tyagi

Assistant Professor

&

Dr. Mayank Agarwal

Assistant Professor

&

Archana Mishra

Group Manager

ST Microelectronics



THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

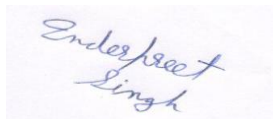
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

JULY, 2025

CERTIFICATE

Certified that project entitled “Verification of I3C Slave Controller” which is being submitted by Inderpreet Singh (University Registration No. 602362012) to the Department of Electronics and Communication Engineering, TIET, Patiala, Punjab, is a record of project work carried out by him under guidance and supervision of Mrs. Archana Mishra (Group Manager at ST Microelectronics), Mrs. Ana Monga (Technical Leader at ST Microelectronics), Dr. Sudhanshu Tyagi (Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology(Deemed to be university) and Dr. Mayank Agarwal (Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology(Deemed to be university). The matter presented in this project report does not incorporate without acknowledgement any material previously published or written by any other person except where due reference is made in the text.

A handwritten signature in blue ink that reads "Inderpreet Singh". The signature is written in a cursive style and is positioned above the printed name.

Inderpreet Singh
602362012

CERTIFICATE

Date : 25th June 2025

This is to Certify that project entitled “Verification of I3C Slave Controller” which is being submitted by Inderpreet Singh (University Registration No. 602362012) to the Department of Electronics and Communication Engineering, TIET, Patiala, Punjab, is a record of project work carried out by him under guidance and supervision of Mrs. Archana Mishra (group manager, APMS IMAGING, STMicroelectronics India Pvt. LTD), during a period from 3 July 2024 to 27 June 2025. This is to certify that the candidate’s above statement is correct to the best of my knowledge.

A rectangular box containing a handwritten signature in blue ink. The signature reads "Archana" in a cursive style, with a horizontal line underneath the name.

Mrs. Archana Mishra
Group Manager
STMicroelectronics India Pvt. LTD

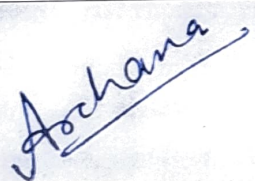


DECLARATION

I, **Inderpreet Singh** hereby declare that the work presented in this thesis entitled “**Verification of I3C Slave Controller**” in partial fulfillment of the requirement for the award of degree of **Master of Technology (VLSI Design)** submitted at **Electronics and Communication Engineering Department**, Thapar Institute of Engineering & Technology (Deemed to be University), Patiala is an authentic record of work carried out under supervision of Dr. Sudhanshu Tyagi (Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology(Deemed to be university), Dr. Mayank Agarwal (Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology(Deemed to be university), Archana Mishra (Group Manager at ST Microelectronics) and Ana Monga (Technical Leader at ST Microelectronics) from July 2024 TO July 2025. The matter presented in this has not been submitted either in part or full to any other university or institute for the award of any other degree.

Inderpreet Singh

Date:20/06/2025

(Inderpreet Singh)
(602362012)

 Mrs Archana Mishra Designation: - Group Manager, ST Microelectronics, Greater Noida Date:07/06/2025	 Dr. Sudhanshu Tyagi (Name of TIET Supervisor-I) Designation: - Assistant Professor Department of Electronics and Communication Engineering, Thapar Institute of Engineering & Technology, Patiala
 Mrs Ana Monga Designation: - Technical Leader, ST Microelectronics, Greater Noida Date:07/06/2025	 Dr. Mayank Agarwal Designation: - Assistant Professor Department of Electronics and Communication Engineering, Thapar Institute of Engineering & Technology, Patiala

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to the individuals who have played an instrumental role in my internship journey at STMicroelectronics. Their guidance, support, and encouragement have been invaluable, and I am truly grateful for their contributions.

First and foremost, I would like to extend my sincere thanks to my mentor, **Mrs. Ana Monga**. Her wealth of knowledge, expertise, and guidance have been instrumental in shaping my professional growth during this internship. Her patience, encouragement, and willingness to share her insights and experiences have been truly inspiring. I am grateful for the time and effort she dedicated to mentoring me and for her unwavering support throughout the internship.

I would also like to express my deep appreciation to my manager **Mrs. Archana Mishra**. Her leadership, guidance, and trust in my abilities have been invaluable. She provided me with challenging projects, allowed me to take ownership of my work, and provided constructive feedback that greatly contributed to my personal and professional development. I am grateful for her mentorship and for creating a positive and inclusive work environment that fostered learning and growth.

I would like to express my gratitude to my college mentors, **Dr. Sudhanshu Tyagi and Dr. Mayank Agarwal** for their continuous support and guidance throughout my internship. The advice, wisdom, and encouragement have been invaluable in helping me navigate through the challenges and make the most out of this experience. I am thankful for their unwavering support and for being a constant source of motivation, their guidance, support, and encouragement have been invaluable, and I am truly grateful for their contributions.

I am deeply grateful to **Dr. Kulbir Singh**, Head of the Department (ECE) and **Dr. Bharat Garg**, M.Tech(VLSI) Coordinator, Thapar Institute of Engineering & Technology, Patiala for providing me with a learning Environment and Infrastructure in ECED.

I will be ignorant if I do not express my gratitude to the author of the references and other literature cited in this seminar. Finally, I want to thank all of my colleagues for their encouragement through potential discussions and suggestions.

ABSTRACT

The major project undertaken was regarding the Verification of I3C Slave Controller. This paper presents a comprehensive approach to the design and functional verification of I3C (Improved Inter-Integrated Circuit), a communication interface that addresses the limitations of traditional I2C and SPI protocols. The design phase includes the development of a UVM testbench for verification of I3c Slave Controller. A detailed verification framework is employed, incorporating both simulation-based testing and hardware validation to ensure compliance with the I3C specification. Functional verification focuses on main aspects such as data integrity, timing accuracy, and error handling under various operating conditions. Through extensive testing, the design shows improved data rates, reduced power consumption, and enhanced device addressing capabilities. The results affirm the successful implementation of I3C Protocol, paving the way for its adoption in next-generation applications across consumer electronics, automotive systems, and IoT devices. This work provides a foundational reference for engineers in the design and verification of I3C interfaces, highlighting best practices and potential challenges in real world applications.

Table of Contents

CERTIFICATE	ii
DECLARATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER 1	
INTRODUCTION.....	1
1.1 Communication Protocol.....	2
1.1.1 I3c.....	2
1.2 Functional Verification.....	3
1.2.1 System Verilog.....	4
1.2.2 UVM (Universal Verification Methodology).....	4
1.3 Formal Verification.....	5
1.3.1 Connectivity.....	6
1.3.2 Superlint.....	7
1.3.3 Software and Tools Used.....	7
1.3.3.1 Cadence JasperGold.....	7
1.3.3.2 Tcl (Tool Command Language).....	8
CHAPTER 2	
LITERATURE REVIEW.....	9
2.1 Literature Review on verification of I3c.....	9
2.2 Literature Review on verification of SOCs.....	13
2.3 Literature Review on Reusability of Testbench.....	16
2.4 Literature Review of Communication Protocols.....	18
2.5 Literature Review of Formal Verification Approach for Circuits.....	20
CHAPTER 3	
OBJECTIVES.....	24
CHAPTER 4	
WORK DONE AND RESULTS.....	27
4.1 Objective: Send dynamic address to I3c Slave.....	27
4.2 Objective: Perform I2c write and read operations.....	29
4.3 Objective: Perform SETAASA CCC command.....	32
4.4 Objective: Perform I3c Write and Read operations.....	35
4.5 Objective: Perform IBI functionality.....	38
4.6 Objective: Perform Superlint Design Checks.....	41

4.7 Objective: Perform Connectivity Checks.....	42
---	----

CHAPTER 5

CONCLUSION AND FUTURE SCOPE.....	43
----------------------------------	----

REFERENCES	44
------------------	----

List of Tables

Table No.	Description	Page No.
2.1	Literature Survey on verification of I3c	11
2.2	Literature Survey on verification of SOCs	15
2.3	Literature Survey on Reusability of Testbench	17
2.4	Literature Survey of Communication Protocols	19
2.5	Literature Survey Comparison on Formal Verification Approach for Digital Circuits	22

List of Figures

Figure No.	Description	Page No.
1.1	I3C Bus Architecture	1
1.2	Verification Languages and Methodologies	3
1.3	Flow of UVM	5
1.4	Formal Verification	6
1.5	Cadence JasperGold Logo	7
2.1	Implementation Block Diagram	9
2.2	Automated test scenario flow	10
2.3	Customized UVM Testbench for SOC	13
2.4	I3c hardware architecture	14
2.5	Power Internet of Things communication architecture	18
2.6	Converging Formal Verification in a High-Level Synthesis Environment	20
2.7	A Formal Verification Approach for Robotic Workflows	21
4.1	Flow of Test 1	27
4.2	Output Log	28
4.3	Tracker	28
4.4	Waveform	28
4.5	Flow of Test 2	29
4.6	Output Log	30
4.7	Tracker	30
4.8	Waveform	31
4.9	Flow of Test 3	32
4.10	Output Log	33
4.11	Tracker	33
4.12	Waveform	34
4.13	Flow of Test 4	35
4.14	Output Log	36
4.15	Tracker	36
4.16	Waveform	37
4.17	Flow of Test 5	38
4.18	Output Log	39
4.19	Tracker	39
4.20	Waveform	40
4.21	Superlint Reports	41
4.22	Connectivity Reports	42

Chapter 1

Introduction

The Improved Inter-Integrated Circuit (I3C) protocol represents a significant advancement in the field of communication interfaces offering enhanced performance and capabilities compared to its predecessors I2C and SPI. As devices in the Internet of Things (IoT), automotive systems and consumer electronics continue to proliferate, the need for robust, high-speed and power-efficient communication protocols has never been more pressing. The design verification of I3C is a critical process that ensures the integrity, functionality and compliance of this emerging interface standard. I3C Bus Architecture[1] is shown in fig 1.1 below.

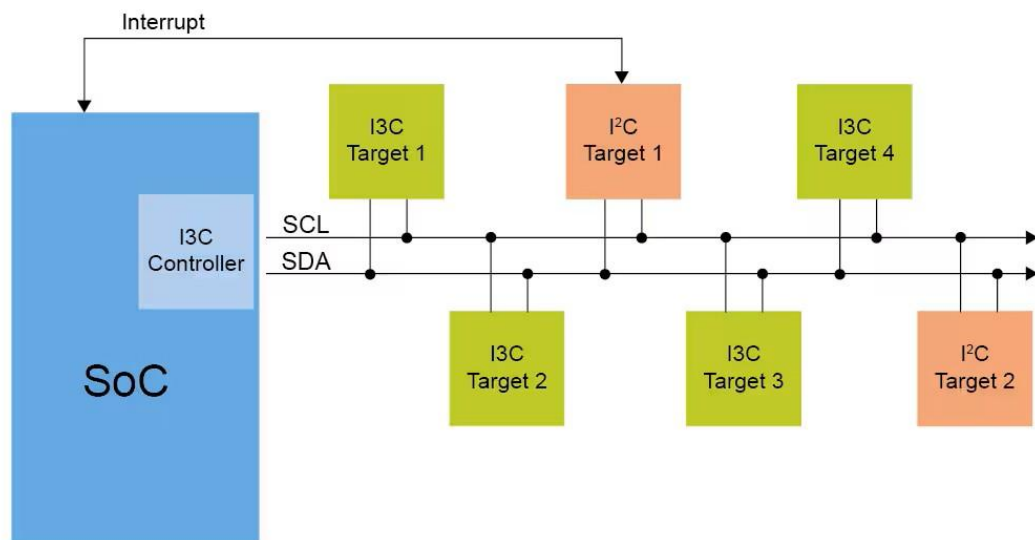


Fig1.1: I3C Bus Architecture[1]

Design verification of I3C encompasses a systematic approach to validating the architecture, protocols and functionality of I3C controllers and devices. Given the complexity of the protocol which includes features such as multi-master support, dynamic addressing and in-band interrupts, thorough verification is essential to identify potential design flaws and ensure reliable performance under various operating conditions.

The verification process begins with a comprehensive understanding of the I3C specification which defines the protocol's features, timing requirements and electrical characteristics. This understanding informs the creation of a detailed verification plan outlining the necessary test cases and metrics for success. The verification strategy typically includes both simulation-based testing and formal verification techniques, ensuring that the design meets its intended functionality and performance criteria.

Simulation allows for the modeling of I3C interactions in a controlled environment enabling the verification team to assess data integrity, timing accuracy and error-handling capabilities.

Additionally formal verification methods can be employed to mathematically validate the design providing a higher level of assurance for critical applications where failure is not an option.

Furthermore hardware validation is essential to the process of design verification. Once the I3C controller and peripheral devices are implemented physical prototypes undergo rigorous testing to ensure they perform as expected in real-world scenarios. This phase helps identify any discrepancies between simulated behavior and actual performance leading to further refinement of the design.

In summary the design verification of I3C is a vital process that combines simulation, formal methods and hardware testing to ensure the protocol meets the stringent requirements of modern electronic systems. As the demand for high performance low power communication interfaces continues to grow effective design verification will be essential in facilitating the successful adoption and deployment of I3C in diverse applications.

1.1 Communication Protocol

A communication A collection of guidelines and customs known as protocol regulates the exchange of information in a network or system. These protocols define how data is formatted and processed to guarantee dependable and effective communication. Communication protocol are essential in enabling devices to understand and interact with each other forming the foundation of modern digital systems.

Purpose of Communication Protocols:

- **Standardization:** Ensures interoperability between different devices, manufacturers and platforms.
- **Error Handling:** Provides mechanisms to detect and correct errors during data transmission.
- **Efficiency:** Optimizes data transfer processes for speed and reliability.
- **Scalability:** Enables seamless addition of new devices or expansion of networks.
- **Security:** Protects data integrity and privacy during communication.

1.1.1 I3c

I3C is an serial communication protocol developed by the MIPI Alliance to address the limitations of the legacy I2C protocol while meeting the growing demands of modern embedded systems. I3C combines the simplicity and multi-master capability of I2C with significant enhancements in speed, power efficiency and functionality.

The I3C protocol was designed to:

- Simplify system integration by enabling communication between a variety of devices including sensors, microcontrollers and processors.
- Enhance data transfer speeds to meet the requirements of high performance systems.
- Provide robust power management features for energy sensitive applications.

- Offer backward compatibility with I2C to ensure easy adoption.

1.2 Functional Verification

The process of confirming that a system, design, or product satisfies its requirements and operates as intended is known as verification. In engineering particularly in hardware and software development verification plays a critical role in validating that a design adheres to its specifications before it is deployed or manufactured. Verification Languages and Methodologies[2] are shown in fig 1.2 below.



Fig 1.2: Verification Languages and Methodologies [2]

Purpose of Verification:

- **Error Detection:** Identify and eliminate errors or inconsistencies in the design or implementation.
- **Compliance:** Ensure that the product complies with industry standards, protocols and customer requirements.
- **Quality Assurance:** Improve the overall reliability, safety and quality of the system.
- **Cost Reduction:** Detecting issues early in the design phase prevents costly fixes in later stages of development or production.

Key Methods and Tools:

- **Simulation:** Mimics the behaviour of a system in a virtual environment.
- **Formal Methods:** Uses mathematical models to verify logic and functionality.
- **Testbenches:** Frameworks used to apply stimuli and check responses in hardware or software verification.
- **Tools:** Popular verification tools include UVM (Universal Verification Methodology) for hardware and tools like Selenium for software testing.

1.2.1 System verilog

The capabilities of Verilog, a conventional hardware description language (HDL), are

expanded by System Verilog (SV), a potent language for hardware description and verification. Developed as an IEEE standard (IEEE 1800), System Verilog combines features for hardware modelling, design and verification making it a cornerstone of modern digital design and verification workflows.

System Verilog is designed to:

- **Model Hardware:** Describe the structure and behavior of digital systems.
- **Simulate Designs:** Enable pre-silicon validation through simulation.
- **Verify Functionality:** Provide advanced constructs for functional verification.
- **Facilitate Reusability:** Support reusable verification environments and IPs.

1.2.2 UVM

UVM is a standardized framework used for functional verification of digital hardware designs primarily written in System Verilog. Developed by Accellera and widely adopted across the semiconductor industry UVM provides a structured, reusable and scalable approach to building verification environments for complex systems-on-chip (SoCs), ASICs and FPGAs. Flow of UVM RAL Model is shown in fig 1.3 below.

UVM was created to:

- **Standardize Verification:** Provide a unified methodology to enable seamless collaboration and interoperability across teams and companies.
- **Increase Reusability:** Encourage the creation of reusable verification components to reduce development time for future projects.
- **Handle Complexity:** Offer advanced features to manage and verify large and complex designs efficiently.

UVM Components

A UVM environment consists of modular components working together to perform verification tasks. Key components include:

- **Driver:**
Sends stimulus (transactions) to the Design Under Test (DUT) by converting them into pin-level signals.
- **Monitor:**
Observes DUT signals and converts them back into transactions for analysis.
- **Sequencer:**
Generates sequences of transactions that the driver sends to the DUT.
- **Scoreboard:**
Compares the expected output with the DUT's actual output to detect mismatches.

- **Environment:**
A container for all verification components connecting them to form a complete testbench.
- **Test:**
Orchestrates the verification process by configuring the environment starting sequences and collecting results.

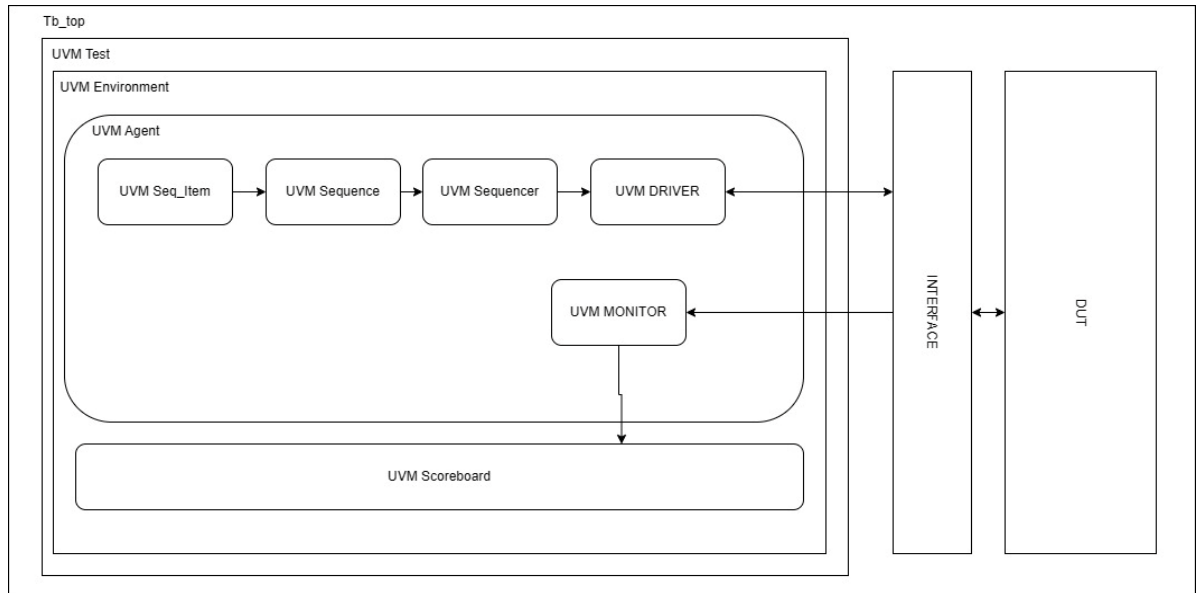


Fig 1.3: Flow of UVM

1.3 Formal Verification

A mathematical technique called formal verification is used to support or refute a system's design's accuracy in relation to a formal specification or property. Unlike simulation which tests the design with a finite set of input vectors formal verification exhaustively explores all possible states and transitions of the system ensuring comprehensive coverage. This exhaustive nature makes formal verification particularly effective in identifying corner-case bugs and subtle design errors that might be missed by simulation. Different types of Formal Verification[3] are given in fig 1.4.

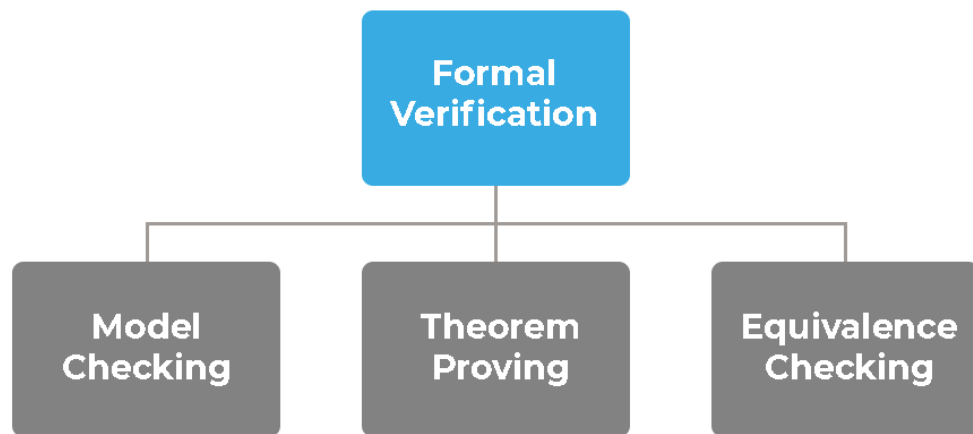


Fig. 1.4: Formal Verification[3]

The primary methodologies in formal verification include a model checking, a theorem proving and a equivalence checking:

- **Model Checking:** This technique involves creating a finite-state model of the system and verifying properties expressed in temporal logic. The model checker systematically explores all possible states of the model to ensure that the specified properties hold true. If a property is violated the model checker provides a counterexample illustrating the sequence of events leading to the violation.
- **Theorem Proving:** Theorem proving involves constructing mathematical proofs to establish the correctness of design properties. This method uses formal logic and proof techniques to verify that the design adheres to its specifications. Theorem proving can handle more abstract and complex properties but often requires significant expertise and manual effort.
- **Equivalence Checking:** This technique ensures that two representations of a design such as the Register Transfer Level (RTL) description and the gate-level netlist are functionally identical. Equivalence checking is crucial for verifying that optimizations and transformations during the synthesis process do not alter the intended functionality of the design.

1.3.1. Connectivity

Formal verification of connectivity involves using mathematical methods to prove that all connections in a design adhere to the specified connectivity rules. This process ensures that signals are correctly routed between modules that there are no unintended shorts or opens and that the design meets the intended architectural specifications. Formal connectivity verification can identify issues such as missing connections incorrect connections and unintended signal overlaps.

The primary goal of formal connectivity verification is to ensure that the design's connectivity is both correct and complete. This involves verifying that all required connections are present and correctly implemented and that there are no unintended connections that could lead to functional errors or performance issues.

1.3.2. Superlint

Superlint is a formal verification tool provided by Cadence JasperGold that focuses on linting and static analysis of RTL (Register Transfer Level) designs. Unlike traditional simulation-based linting tools Superlint leverages formal verification techniques to detect a wide range of design issues early. It combines benefits of static analysis and formal methods to ensure high-quality RTL code adherence to coding standards and early detection of potential design bugs.

Superlint is particularly valuable for designers and verification engineers as it helps identify issues that might otherwise remain undetected until later stages of the verification or synthesis process

1.3.3. Software and Tools Used

Software and tools used during formal Verification are discussed below.

1.3.3.1 Cadence JasperGold



Fig. 1.5: Cadence JasperGold Logo[4]

Cadence JasperGold[4] is a comprehensive formal verification platform developed by Cadence Design Systems. It leverages formal methods to provide exhaustive verification of digital designs, ensuring that they meet their specifications under all possible conditions. JasperGold is designed to complement traditional simulation-based verification, offering a range of formal verification applications tailored to specific verification tasks. These applications enable designers and verification and improving overall design quality. Cadence JasperGold Logo is given in fig 1.2.

Cadence JasperGold is a powerful and comprehensive formal verification platform that addresses the challenges of verifying complex digital designs. By providing exhaustive coverage, early bug detection, and targeted verification capabilities, JasperGold ensures

that designs meet their specifications under all possible conditions. Its integration with simulation-based verification environments and advanced debugging and visualization tools make it an essential component of the modern verification landscape. As digital designs continue to grow in complexity,

1.3.3.2 Tcl (Tool Command Language)

The interpreted scripting language TCL was created with user-friendliness and speed in mind. It is renowned for being easy to use and adaptable, which makes it a great option for both inexperienced and seasoned programmers. Tcl is often used in conjunction with Tk, a toolkit for creating graphical user interfaces (GUIs), to build cross-platform applications with ease.

Tcl's design philosophy emphasizes simplicity and minimalism. The language syntax is straightforward, with commands and arguments separated by whitespace, and it supports dynamic typing, where variables do not have fixed types and can hold any type of value. Tcl's extensibility allows developers to create custom commands and integrate Tcl with other programming languages, making it a versatile tool for a wide range of applications.

Chapter 2

Literature Review

2.1 Literature Review on verification of I3c

The paper titled "Development of UVM Testbench for I3C protocol"[5] by Shreyash Naresh Chauhan and Ganesh K Andurkar focuses on the creation of a reusable and efficient Universal Verification Methodology (UVM), testbench for MIPI I3C-based design verification. The MIPI Alliance created the I3C protocol, a sophisticated serial communication system that expands on the features of the popular I2C standard. Creating a thorough UVM testbench with MIPI I3C protocol-specific sequences, transactions, and coverage models is the primary goal of this research.

The paper titled "Architecture Analysis & Verification of I3C Protocol"[6] by Anusha Mahale and Dr. Kariyappa B S focuses on the verification of the I3C protocol using the UVM. The I3C protocol, developed by the MIPI Alliance is an advanced communication protocol designed to interface sensor peripherals in mobile SoCs. This paper provides a detailed study of the I3C bus protocol and describes the development of a verification environment using System Verilog and UVM. The verification environment aims to achieve high functional and code coverage ensuring the correctness of the I3C protocol implementation. Test Bench Implementation UVM Flow Diagram is shown in fig 2.1 below.

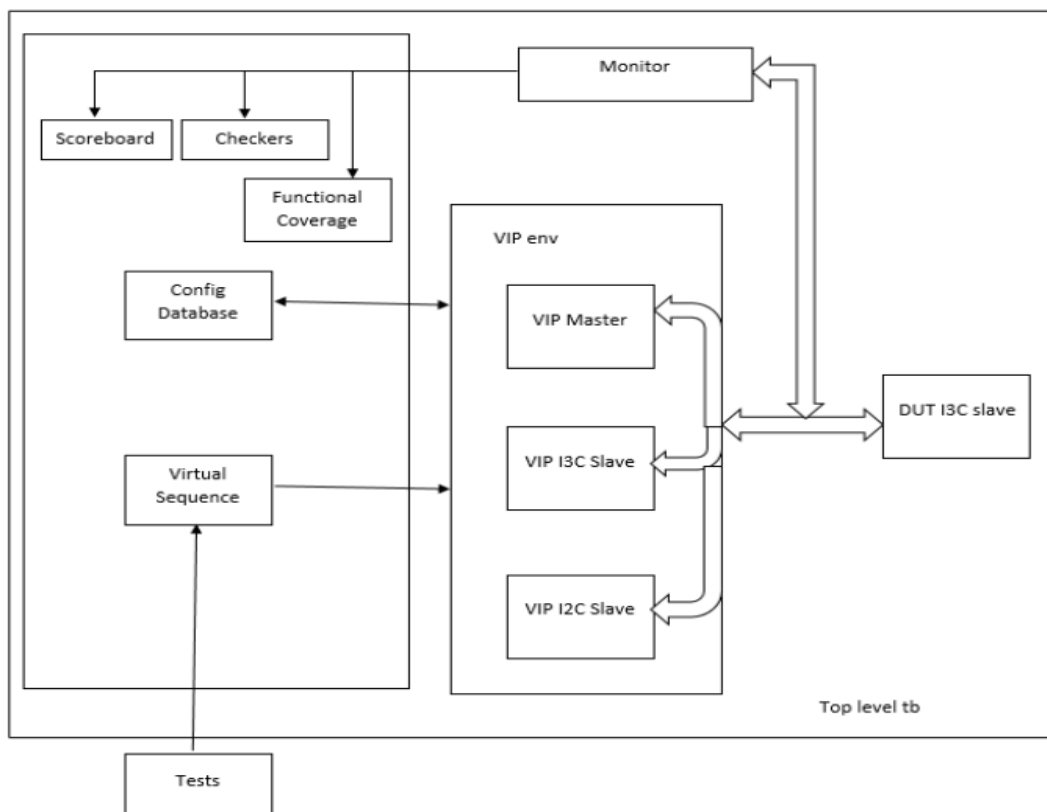


Fig 2.1: Implementation Block Diagram[6]

The paper titled "Design & Implementation of MIPI I3C Master Controller SubSystems"[7] by Yadhu Krishnan S and Ramesh Bhakthavatchalu focusses on utilising Xilinx Vivado Spartan 7 FPGA to simulate and perform the design and construction of a MIPI I3C master controller using Verilog HDL. The MIPI Alliance's I3C protocol offers a fast, low-pin-count interface for sensor communication, addressing the increasing design challenges in mobile, automotive, and Internet of Things devices. Enhancing sensor communication capabilities with better performance and power economy is the goal of this work.

The paper titled "Verification of the Legacy Compatibility of the MIPI I3C Master"[8] by Mario Golubic, Jurica Kundrata and Adrijan Baric focuses on confirming the MIPI I3C master's legacy interoperability with I2C devices. A backward-compatible replacement for the I2C protocol, the MIPI Alliance's I3C protocol offers enhancements in speed, functionality, and energy economy. A test procedure designed to validate I3C master parameters by assessing both I3C and older I2C signal parameters is presented in this work.

Automated test scenario flow is shown in fig 2.2 below

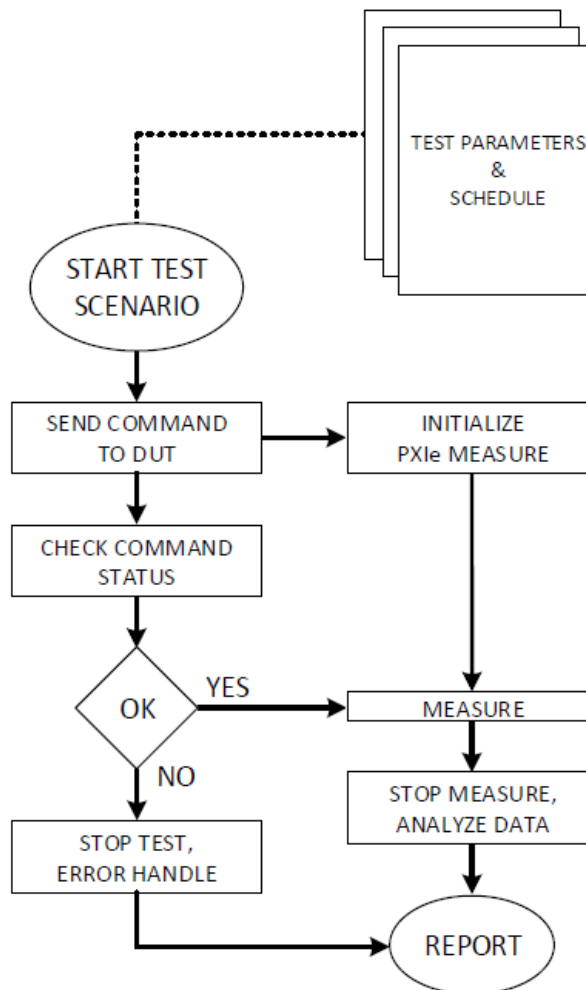


Fig 2.2: Automated test scenario flow[8]

The paper titled "An Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC's): A Review"[9] by N B Harshitha, Praveen Kumar Y G and Dr. M Z Kurian focuses on UVM and its use in the validation of integrated circuit (IC) digital designs. A reliable, reusable, and interoperable verification IP and testbench components are offered by UVM, a standard verification technique. This study examines UVM's architecture, use in functional verification, and involvement in complex digital design verification.

Literature Survey Comparison on verification of I3c is shown in table 2.1 below.

Table 2.1: Literature Survey on verification of I3c

Aspect	Development of UVM Testbench for I3C Protocol[5]	Architecture Analysis & Verification of I3C Protocol[6]	Design & Implementation of MIPI I3C Master Controller SubSystems[7]	Verification of Legacy Compatibility of the MIPI I3C Master[8]	Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC's): A Review[9]
Publication Year	2023	2019	2023	2021	2021
Main Focus	Development of UVM testbench for I3C protocol	Architecture analysis & verification of I3C protocol	Design & implementation of I3C master controller subsystems	Verification of legacy compatibility of I3C master	Review of UVM for digital design verification
Research Gaps	- Need for more verification scenarios and enhanced coverage plan.	- Need for more verification scenarios and enhanced coverage plan.	-Further optimization and exploration of additional use cases for I3C protocol.	-Further improvement of test methods and test setup for comprehensive verification.	-Continued exploration of UVM in various verification scenarios and its application to different protocols and designs.
Applications	- Automotive SoCs	- Automotive devices - Mobile SoCs	-Sensor communication - Mobile devices	- Stationary devices - ASICs	- Digital design verification across various industries

	- Mobile SoCs				
Key Contributions	- Development of a comprehensive UVM testbench for I3C protocol verification.	- Detailed study and verification of I3C protocol architecture.	- Design & implementation of I3C master controller using VHDL.	- Development of a test method for verifying I3C master parameters and legacy compatibility.	- Review of UVM and its application in digital design verification.

2.2 Literature Review on verification of SOCs

Design verification is a critical phase in the development of Intellectual Property (IP) subsystems & SoC. Traditional verification methods are time-consuming, resource-intensive and costly. They also pose a high risk of bug escape especially for complex designs. The paper "Deploying Artificial Intelligence in Design Verification to Accelerate IP/SoC Sign-off with Zero Escape"[10] by Bhattacharjee et al. explores the use of Machine Learning (ML) to enhance the verification process focusing on a Temperature Sensor (TS) in Dual-In-line Memory Module (DIMM) of Dynamic Random Access Memory (DRAM).

Verification of I3C (Improved Inter-Integrated Circuit) is crucial for ensuring the reliability and functionality of integrated circuits. The paper "Formal & Simulation Verification: Comparing & Contrasting two Verification Approaches"[11] by Girish M, Gopakumar G and Divya D S provides a comprehensive comparison between formal verification and simulation-based verification methods. The study uses an encoder block implementing Manchester encoding for the MIL-1553 protocol as the design under test (DUT).

The integration & verification of Intellectual Property cores on SoC are critical for ensuring the functionality and performance of modern electronic devices. The paper "Integration & Verification of IP Cores on SoC"[12] by Arpitha O Naik, Elizabeth Kuruvilla and Arunkumar P Chavan discusses the challenges and methodologies involved in integrating various IP cores and verifying their interactions within an SoC. This review focuses on the verification of I3C (Improved Inter-Integrated Circuit) in the context of the discussed methodologies. Customized UVM Testbench for SOC is shown in fig 2.3 Below.

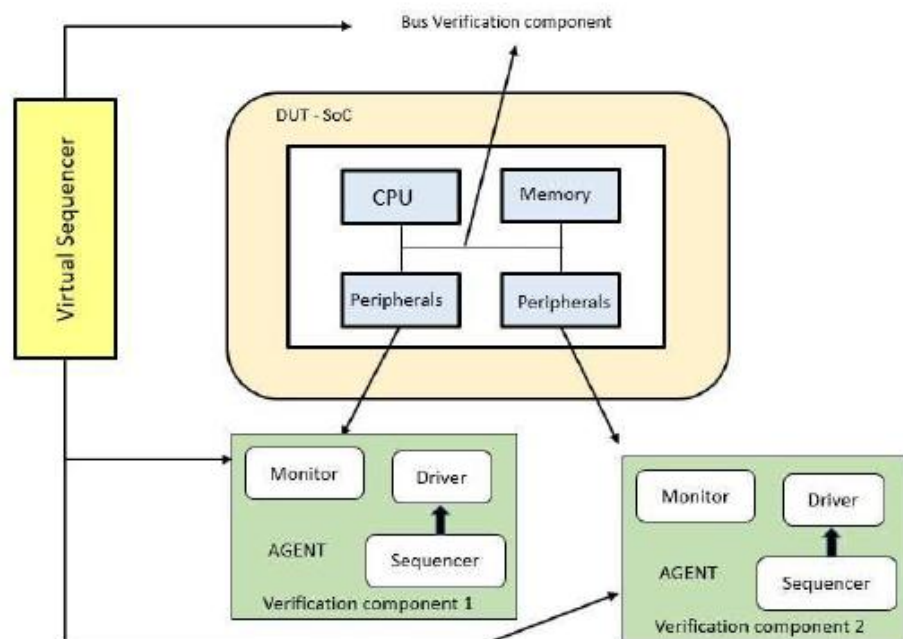


Fig 2.3: Customized UVM Testbench for SOC[12]

The I3C protocol developed by MIPI Alliance is an evolution of the I2C protocol offering enhanced performance and features. The paper "Design Implementation & Verification of a Flexible I3C Hardware Architecture"[13] by Peng Gao, Neng Xu, Xin Zheng, Jian Gong and Xu Zhong discusses the design, implementation & verification of a flexible I3C hardware architecture. This review focuses on the verification aspects of the I3C protocol as presented in the paper. I3c hardware architecture Shown in fig 2.4 below.

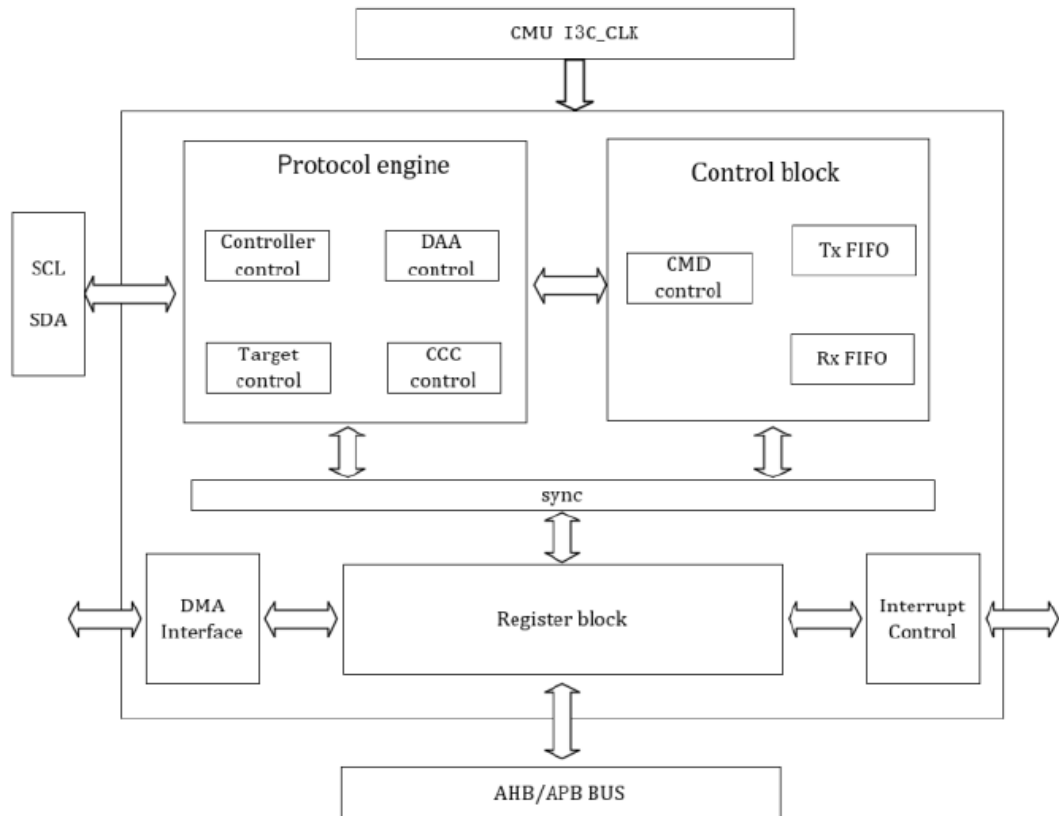


Fig 2.4: I3c hardware architecture[13]

The paper "SoC Level Verification Using System Verilog"[14] by Purvi D. Mulani presents a comprehensive approach to the verification of System-on-Chip (SoC) components specifically focusing on the AMBA AHB Interconnect Matrix & I2C Design Under Test (DUT). This review will focus on the verification methodologies discussed in the paper particularly in the context of verifying the Improved Inter-Integrated Circuit (I3C) protocol. Literature Survey comparison on verification of SOCs is shown in table below.

Table 2.2: Literature Survey on verification of SOCs

Aspect	Deploying Artificial Intelligence in Design Verification[10]	Formal and Simulation Verification [11]	Integration & Verification of IP Cores on SoC[12]	Design Implementation & Verification of a Flexible I3C Hardware Architecture[13]	SoC Level Verification Using SystemVerilog[14]
Publication Year	2024	2021	2021	2024	2009
Main Focus	AI-driven verification of IP/SoC	Comparison of formal and simulation verification	Integration & verification of IP cores on SoC	Flexible I3C hardware architecture and its verification	SoC level verification using SystemVerilog
Research Gaps	Limited to a specific design module (Temperature Sensor); potential for exploring ML in other verification scenarios	Limited to a specific protocol (Manchester encoding); need for broader application	Focused on specific IP cores; may not cover all SoC aspects	Focus on I3C; may not generalize to other protocols; lacks exploration of post-silicon verification	Limited to AMBA AHB and I2C; may not cover other protocols; lacks detailed coverage metrics
Applications	VLSI design, IP/SoC verification, semiconductor industry	Digital design verification, protocol verification	SoC design, IP core integration, semiconductor industry	I3C-based systems, sensor integration, SoC design	SoC design, AMBA AHB-based systems, I2C devices
Key Contributions	75% faster IP development and execution cycle	Formal verification is advantageous for small, protocol-based designs	High functional coverage with UVM, zero errors after integration	Flexible and reusable I3C hardware architecture, thorough verification	Reusable verification environment, efficient protocol checking

2.3 Literature Review on Reusability of Testbench

The paper "Design & UVM based Verification of UART, SPI and I2C Protocol"[15] by Nishant Sahay and Sachin Gajjar provides a comprehensive study on the design and verification of three prevalent communication protocols: UART, SPI and I2C. This review focuses on the verification aspects of the I2C protocol which combines the benefits of both UART and SPI protocols.

The paper "Exploration on Reusability of Universal Verification Methodology"[16] by Mrs. N. Bhuvaneshwary, Dr. J. Deny and Dr. A. Lakshmi explores the reusability aspects of the Universal Verification Methodology (UVM) in the context of verifying communication protocols with specific focus on the I2C protocol. This review will focus on the verification aspects of the I3C protocol which is an advanced version of the I2C protocol using UVM.

The paper "Functional Verification Environment for I2C Master Controller using System Verilog"[17] by M. Sukhanya and K. Gavaskar presents a comprehensive study on the verification of I2C Master Controller using SV. This review will focus on the verification aspects of the I3C protocol which is an advanced version of the I2C protocol using the methodologies and techniques discussed in the paper.

The paper "Exploration of Using Direct Programming Interface to Improve the Reusability of Verification IP"[18] by Xiaocheng Wang, Hao Ruan and Long Zou investigates the use of Direct Programming Interface (DPI) to enhance the reusability of verification IP particularly in the context of verifying the APB_I2C IP core. This review will focus on the verification aspects of the I3C protocol an advanced version of the I2C protocol using the methodologies and techniques discussed in the paper. Literature Survey comparison on Reusability of Testbench is given in table 2.3 below.

Table 2.3: Literature Survey on Reusability of Testbench

Aspect	Design & UVM based Verification of UART, SPI, and I2C Protocols[15]	Exploration on Reusability of Universal Verification Methodology [16]	Functional Verification Environment for I2C Master Controller using System Verilog [17]	Exploration of Using Direct Programming Interface to Improve the Reusability of Verification IP[18]
Publication Year	2024	2022	2017	2022
Main Focus	<ul style="list-style-type: none"> - FSM-based design - UVM sequences - QuestaSim tool 	<ul style="list-style-type: none"> - Reusability of UVM components - Reusable test cases 	<ul style="list-style-type: none"> -Constrained randomization - Functional and code coverage - Mentor Graphics tool 	<ul style="list-style-type: none"> -Multi-language collaborative verification -Interactive debugging library - Vertical reuse of C test cases
Research Gaps	<ul style="list-style-type: none"> - Lack of detailed coverage metrics - Limited focus on advanced I3C features 	<ul style="list-style-type: none"> - Limited practical examples - Lack of advanced debugging techniques 	<ul style="list-style-type: none"> - No explicit focus on reusability - Limited to I2C, not addressing I3C enhancements 	<ul style="list-style-type: none"> - Limited focus on specific I3C features - Needs more detailed application examples
Applications	<ul style="list-style-type: none"> - Verification of communication protocols in FPGA designs -Resource utilization analysis 	<ul style="list-style-type: none"> -Reusable verification components for various protocols -Efficient verification methodologies 	<ul style="list-style-type: none"> - Verification of I2C Master Controller in complex designs -Constrained randomization techniques 	<ul style="list-style-type: none"> - Limited focus on specific I3C features - Needs more detailed application examples
Key Contributions	Comparison of UART, SPI, and I2C verification using UVM	Reusability of UVM verification components and test cases	Verification environment for I2C Master Controller using System Verilog	Improved reusability & efficiency of verification IP using DPI and interactive debugging

2.4 Literature Review of Communication Protocols

The paper "Analysis and Research on UART Communication Protocol"[19] by Weilun Huang and Guolun Sheng provides a comprehensive study of the Universal Asynchronous Receiver Transmitter (UART) protocol which is fundamental in digital communication. This review will focus on the verification of I3C (Improved Inter-Integrated Circuit) in the context of the UART communication protocol highlighting the principles, hardware structure, implementation, advantages and challenges of UART and drawing parallels to I3C.

The paper "Research on Communication Protocol Adaptability of Power Internet of Things Application Layer"[20] by Jian Gao et al. provides an in-depth analysis of the adaptability of various communication protocols in the context of the Power Internet of Things (IoT). This review focuses on the verification of the I3C (Improved Inter-Integrated Circuit) protocol, emphasizing its adaptability, reliability and performance in the Power IoT application layer as discussed in the referenced paper. Power Internet of Things communication architecture is shown in fig 2.5 below.

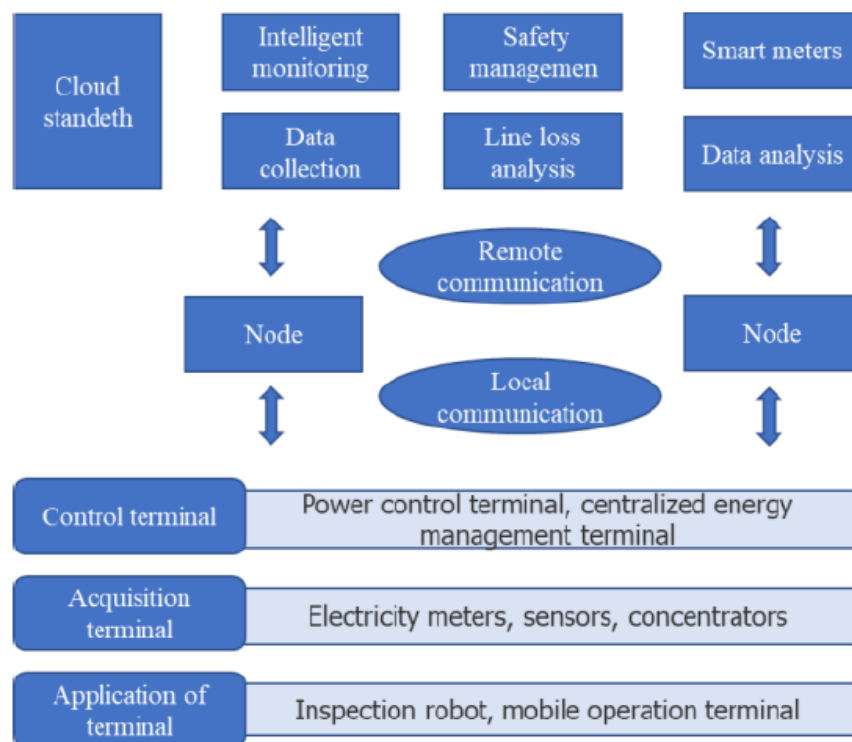


Fig 2.5: Power Internet of Things communication architecture [20]

The paper "I2C Protocol and its Clock Stretching Verification using System Verilog & UVM"[21] by Lakshmi Manasa Kappaganthu, Durga Prakash M and Avinash Yadlapati provides an in-depth analysis of the I2C protocol focusing on its verification using System Verilog and UVM. This review will focus on the verification of the I3C (Improved Inter-Integrated Circuit) protocol emphasizing its advanced features, particularly in comparison to I2C and the methodologies used for its verification. Literature Survey comparison of Communication Protocols is shown in table 2.4 below.

Table 2.4: Literature Survey of Communication Protocols

Aspect	Analysis and Research on UART Communication Protocol[19]	Research on Communication Protocol Adaptability of Power Internet of Things Application Layer[20]	I2C Protocol and its Clock Stretching Verification using System Verilog and UVM[21]
Publication Year	2024	2022	2017
Main Focus	Comprehensive study on UART protocol its principles hardware structure, and applications	Evaluation of communication protocol adaptability in Power IoT using AHP and Bayesian methods	Verification of I2C protocol and clock stretching using System Verilog and UVM
Research Gaps	<ul style="list-style-type: none"> - Lack of specific verification methodologies for UART. - Limited focus on advanced features and future enhancements. 	<ul style="list-style-type: none"> - Need for more research on application layer protocols specific to the power industry. - Lack of detailed case studies on real-world implementations. 	<ul style="list-style-type: none"> - Focused on I2C but lacks comparison with other advanced protocols like I3C. - Limited discussion on the integration of I2C with modern SoC designs.
Applications	<ul style="list-style-type: none"> - Internet of Things - Industrial control - Healthcare - Telecommunications - Aviation 	<ul style="list-style-type: none"> - Power Internet of Things - Smart grids - Digital grids 	<ul style="list-style-type: none"> - System on Chip (SoC) - Microcontrollers - Peripheral devices
Key Contributions	UART principles, hardware structure, data frames, advantages, challenges applications in various fields	Power IoT communication architecture business requirements application layer protocols, adaptability evaluation	I2C protocol basics, clock stretching, verification environment, System Verilog, UVM, SimVision tool

2.5 Formal Verification Approach for Digital Circuits

The paper "Formal Verification of Floating-Point Division"[22] by Ashish Kapoor, Warren Ferguson, Himanshu Jain and Sudipta Kundu presents a formal verification methodology for verifying floating-point (FP) dividers. Floating-point division units are complex and critical components in hardware designs, and ensuring their correctness is essential to avoid significant financial and functional repercussions. Traditional verification techniques such as random testing and scenario-based testing often fall short in guaranteeing completeness making formal verification a preferred approach.

The paper "Converging Formal Verification in a High-Level Synthesis Environment"[23] by Michael Dossis presents a formal verification methodology integrated within a high-level synthesis (HLS) environment. The approach focuses on verifying the correctness of digital designs by combining high-level synthesis with formal verification techniques. This integration aims to address the challenges of traditional simulation-based verification which is often time-consuming and prone to missing corner-case bugs. Converging Formal Verification in a High-Level Synthesis Environment is shown in Fig.2.2 below.

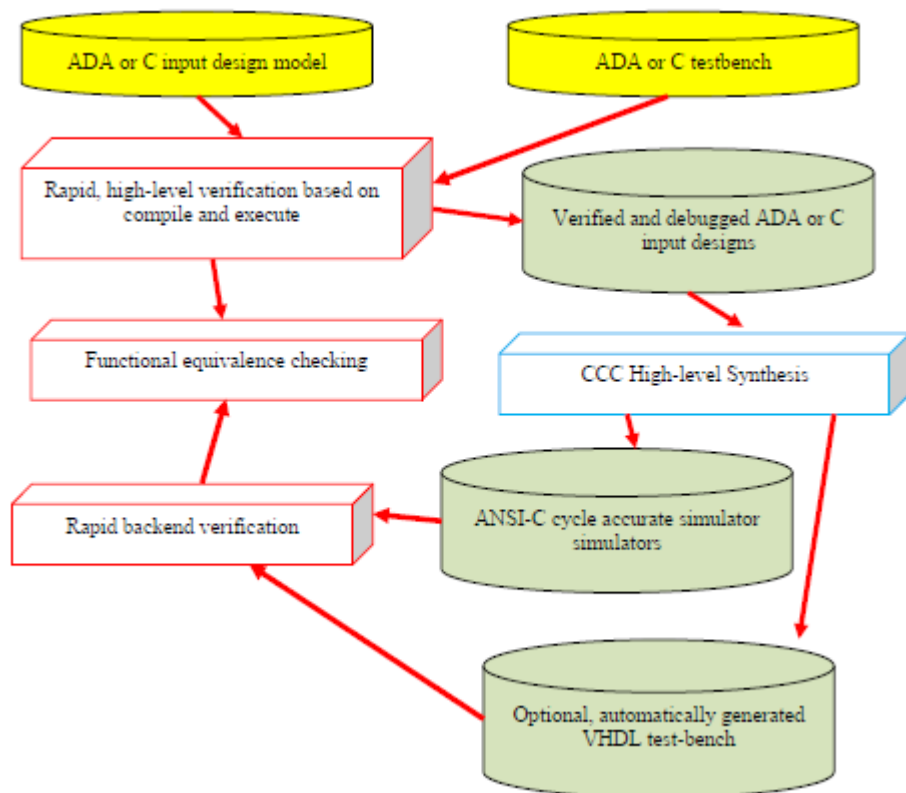


Fig. 2.6: Converging Formal Verification in a High-Level Synthesis Environment [23]

The paper "A Formal Verification Approach for Robotic Workflows"[24] by Michael Rathmair, Thomas Haspl, Titanilla Komenda, Bernhard Reiterer and Michael Hofbaur presents a formal verification methodology tailored for robotic workflows in manufacturing processes. The approach integrates Business Process Model and Notation (BPMN)-based input models into structured verification & design architecture leveraging formal verification to ensure the correctness & reliability of complex robotic systems. A Formal Verification Approach for Robotic Workflows is shown in Fig 2.1.



Fig. 2.7: A Formal Verification Approach for Robotic Workflows[24]

The paper "A Fault Localization Approach Derived From Testing-based Formal Verification"[25] by Rong Wang, Shaoying Liu and Yuji Sato presents a novel fault localization algorithm that leverages testing-based formal verification with symbolic execution (TBFV-SE). This approach aims to enhance efficiency & accuracy of fault localization in safety-critical systems by integrating formal verification and testing methods.

The paper "C Programs Formal Verification Using Maude"[26] by Fateh Boutekkouk presents a novel approach for the formal verification of C programs using the Maude formal language. The approach involves transforming C code into a Control/Data Flow Graph (CDFG) which is then used to generate a Maude system module. This module can be utilized for simulation, reachability analysis and verification against Linear Temporal Logic (LTL) specifications. The primary objective is to leverage Maude's capabilities to verify both control-oriented and data-oriented properties of C programs.

Literature Survey Comparison on Formal Verification Approach for Digital Circuits is done in table 2.5 Below.

Table 2.5: Literature Survey Comparison on Formal Verification Approach for Digital Circuits

Aspect	Converging Formal Verification in a High-Level Synthesis Environment[22]	C Programs Formal Verification Using Maude[23]	A Formal Verification Approach for Robotic Workflows[24]	A Fault Localization Approach Derived From Testing-based Formal Verification [25]	Formal Verification of Floating-Point Division[26]
Publication Year	2021	2024	2021	2020	2023
Main Focus	Integrating formal verification within high-level synthesis (HLS)	Formal verification of C programs using Maude	Formal verification of robotic workflows using BPMN-based models	Fault localization using testing-based formal verification with symbolic execution	Formal verification of floating-point division
Research Gaps	- Limited support for input languages (only ADA and ANSI-C)	- Handling pointers, castings, and structures in C programs	- Complexity of BPMN models for large robotic workflows - Need for better integration with	- Dependency on the quality of test data - Computatio	- Generalizing methodology to other arithmetic operations

	<ul style="list-style-type: none"> - Need for more input and output verification formats - Handling complex hierarchical designs efficiently 	<ul style="list-style-type: none"> - Extending methodology to cover more complex types - Automating the preprocessing step for C code 	<ul style="list-style-type: none"> domain-specific tools - Handling dynamic and adaptive robotic behaviors 	<ul style="list-style-type: none"> nal resources required for large programs - Enhancing automation and efficiency of fault localization 	<ul style="list-style-type: none"> - Handling special cases and exceptions in floating-point arithmetic - Improving scalability for larger designs
Applications	<ul style="list-style-type: none"> - Digital designs - SoC and ASIC development 	<ul style="list-style-type: none"> - C programs - Embedded systems 	<ul style="list-style-type: none"> - Robotic workflows - Manufacturing processes 	<ul style="list-style-type: none"> - Safety-critical systems - Software debugging 	<ul style="list-style-type: none"> - Floating-point division units - Datapath circuits
Key Contributions	<ul style="list-style-type: none"> - Formal verification integrated within HLS Ensures functional equivalence between high-level code and synthesized HDL - Automatic generation of verification artifacts 	<ul style="list-style-type: none"> Transformational approach using CDFG - Automatic Maude code generation - Comprehensive verification using Maude tools 	<ul style="list-style-type: none"> - Layer-based verification for robotic workflows - BPMN-based modelling - Integration with multiple verification tools 	<ul style="list-style-type: none"> - Effective fault localization using TBFV-SE - Combines formal verification and testing - Algorithm for pinpointing faulty statements 	<ul style="list-style-type: none"> - Two-step verification for floating-point division - Combines fixed-point and floating-point verification - Uses equivalence checking for accuracy

Chapter 3

Objectives

Develop a Comprehensive UVM Testbench for I3C Protocol:

Objective: Design and implement a modular, reusable and scalable UVM testbench tailored to the I3C protocol.

Tasks:

- Create UVM components such as agents, drivers, monitors, scoreboards and sequencers.
- Ensure the testbench architecture supports various I3C features and scenarios.

Verify Basic I3C Protocol Operations:

Objective: Validate the fundamental operations of the I3C protocol, including data transfer, addressing and bus arbitration.

Tasks:

- Develop test cases for SDR and HDR modes.
- Verify correct data transfer between master and slave devices.
- Ensure proper handling of start, stop and repeated start conditions.

Test Dynamic Addressing and Hot-Join Mechanisms:

Objective: Validate the dynamic addressing and hot-join capabilities of the I3C protocol.

Tasks:

- Create test scenarios for dynamic address assignment and resolution.
- Verify the hotjoin mechanism ensuring that devices can join bus without disrupting ongoing communication.

Verify In-Band Interrupts and Error Handling:

Objective: Ensure the I3C protocol correctly handles in-band interrupts and error conditions.

Tasks:

- Develop test cases for in-band interrupt generation and handling.
- Verify error detection and recovery mechanisms, including parity errors and bus contention.

Ensure Backward Compatibility with I2C Devices:

Objective: Verify that the I3C protocol maintains backward compatibility with legacy I2C devices.

Tasks:

- Develop test cases to ensure I3C master can communicate with I2C slave devices.
- Validate that I2C devices can coexist on the same bus with I3C devices without interference.

Achieve High Functional and Code Coverage:

Objective: Ensure comprehensive verification by achieving high functional and code coverage.

Tasks:

- Implement coverage models to track functional and code coverage metrics.
- Identify and address coverage gaps by developing additional test cases.

Perform Regression Testing:

Objective: Conduct regression testing to ensure that changes and updates do not introduce new issues.

Tasks:

- Develop a regression test suite covering all critical aspects of the I3C protocol.
- Automate the execution of regression tests and monitor results over multiple iterations.

Document Verification Methodology and Results:

Objective: Provide comprehensive documentation of the verification methodology, testbench architecture and results.

Tasks:

- Document the design and implementation of the UVM testbench.
- Compile test plans, test cases, and coverage reports.
- Summarize verification findings and provide recommendations for further improvements.

Chapter 4

Work done and Results

4.1 Objective : Send dynamic address to I3c Slave

Create a UVM Sequence to assign dynamic address to all I3c slaves connect to master using ENTDAAs (CCC command to enter dynamic address) and read it successfully. The flow of test is given in fig 4.1 below.

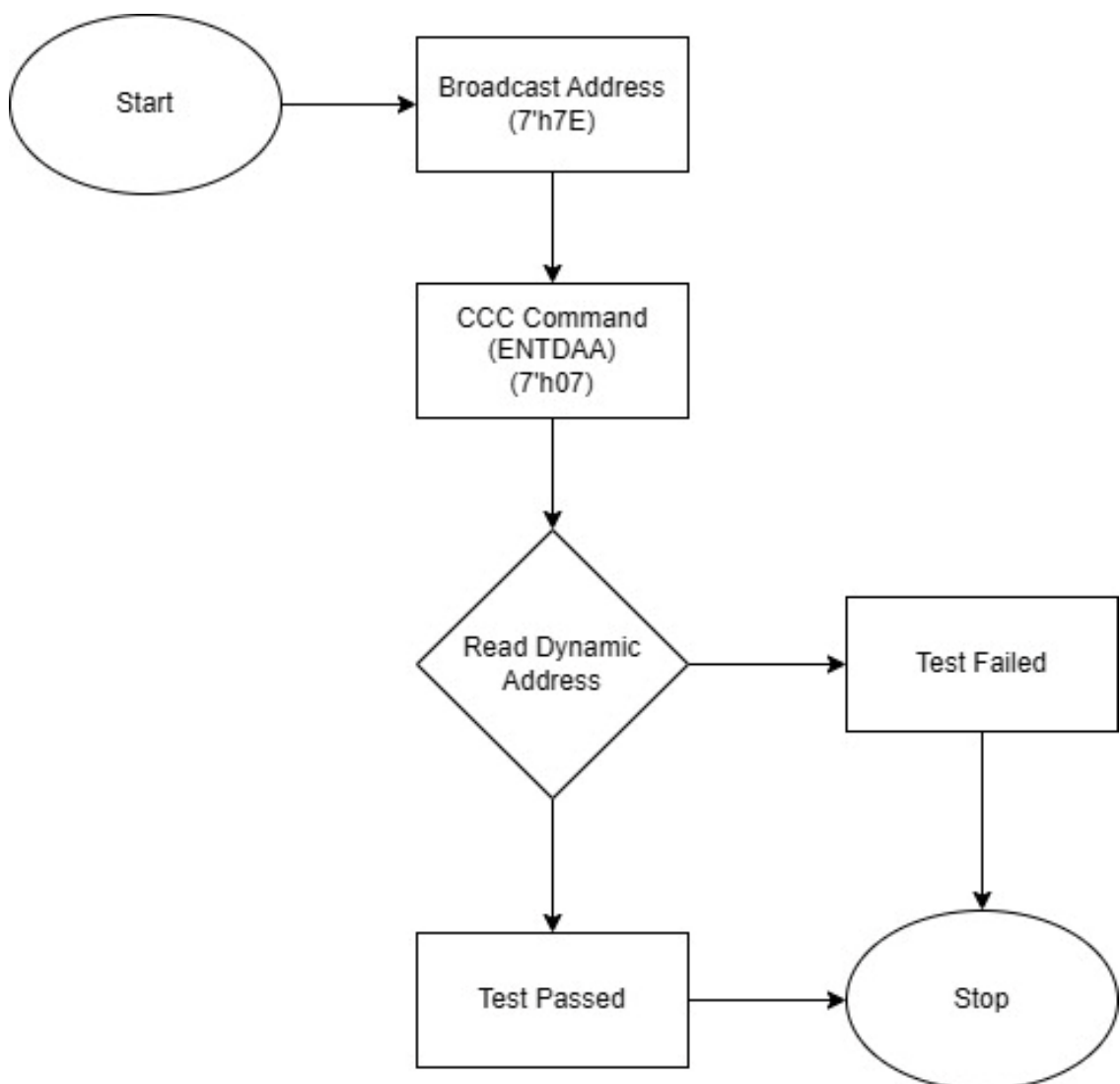


Fig 4.1: Flow of Test 1

Results:

```
UVM_INFO : 85
UVM_WARNING : 2
UVM_ERROR : 0
UVM_FATAL : 0
```

reading dynamic address successfully

| Simulation status : Pass

Fig 4.2: Output Log

Prot	Kind	RW	DevID	Status	DataLen	Data

					CCC^M	

SDR	CCC	-----	0X04	PASS	ENTDAA	0208010100002626
SDR	CCC	-----	0X05	PASS	ENTDAA	07f8000000cc1603
SDR	CCC	-----	0X06	PASS	ENTDAA	07f8000000cd1603

Fig4.3: Tracker

From the above fig 4.2 we can observe that UVM is being used to develop the test bench and required test is passed successfully. We can observe from fig 4.3 that dynamic address is allotted to all slaves successfully.

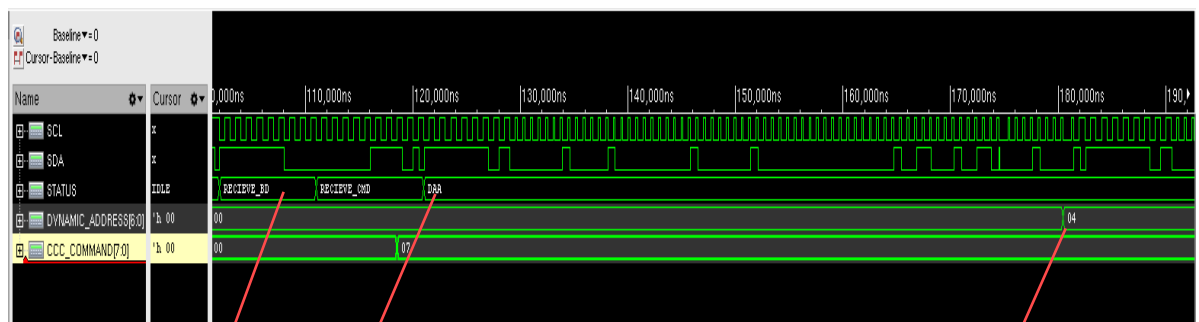


Fig4.4: Waveform

we can observe from fig 4.4 that at point

- (a) we are sending broadcast address
- (b) we are send ENTDAA CCC command to enter dynamic addressing mode
- (c) we can observe that dynamic address 04 is allotted to our slave.

4.2 Objective : Perform I2c write and read operations.

Create a UVM Sequence to assign different (A5 & 5A) Static addresses to I2c Slave and perform read write operations on it. The flow of test is shown in fig 4.5 below.

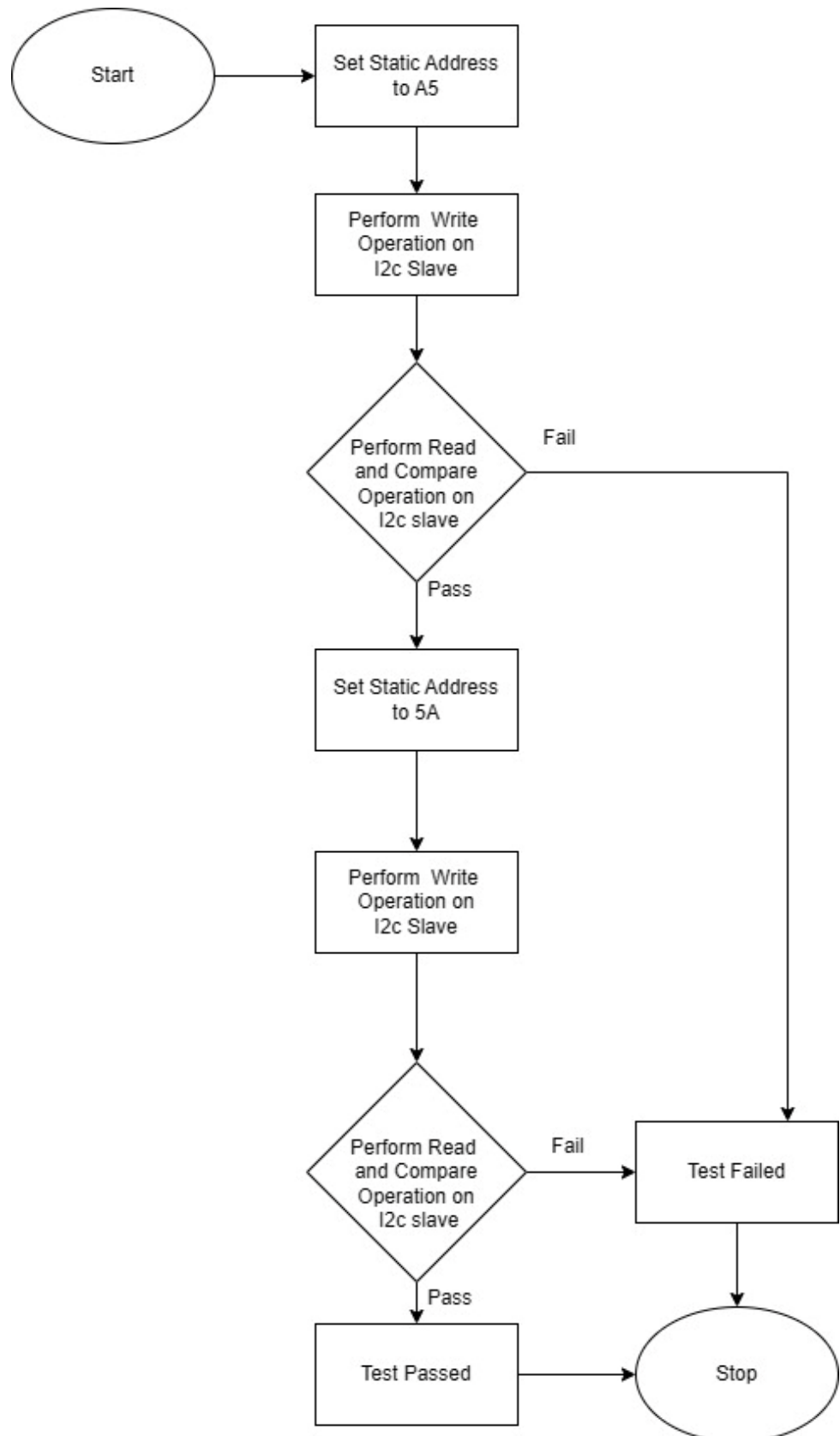


Fig4.5: Flow of Test 2

Results:

```

UVM_INFO : 77
UVM_WARNING : 0
UVM_ERROR : 0
UVM_FATAL : 0 | Simulation status : Pass
    
```

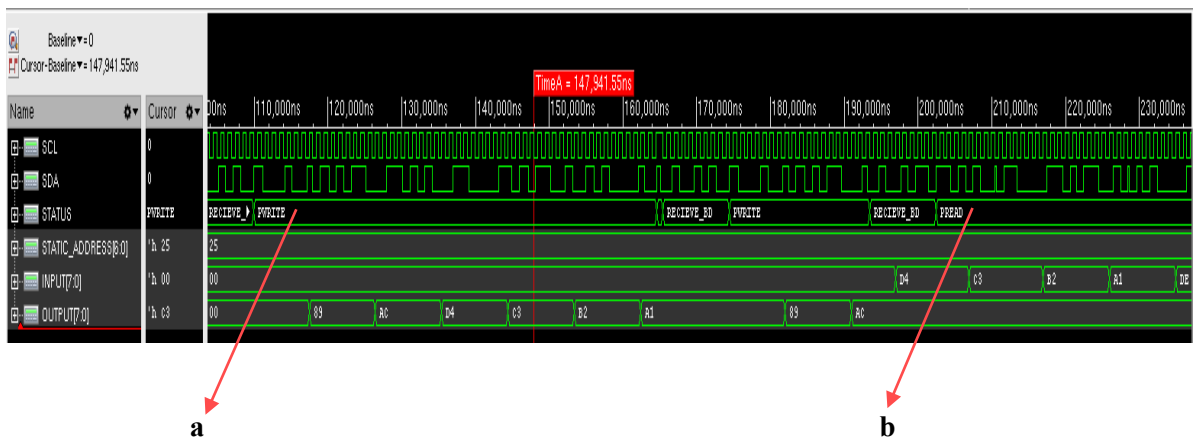
Fig 4.6: Output Log

Prot	Kind	RW	DevID	Status	DataLen	Data

					CCC^M	
I2C	PRV	WRITE	0X25	PASS	6	89acd4c3b2a1
I2C	PRV	WRITE	0X25	PASS	2	89ac
I2C	PRV	READ	0X25	PASS	4	d4c3b2a1
I2C	PRV	WRITE	0X5A	PASS	6	89ac88776655
I2C	PRV	WRITE	0X5A	PASS	2	89ac
I2C	PRV	READ	0X5A	PASS	4	88776655
I2C	PRV	WRITE	0X25	PASS	6	1200efcdab89
I2C	PRV	WRITE	0X25	PASS	2	1200
I2C	PRV	READ	0X25	PASS	4	efcdab89

Fig4.7: Tracker

From the above fig 4.6 we can observe that UVM is being used to develop the test bench and required test is passed successfully. We can observe from fig 4.7 that I2c write and operations are performed at different device ID.



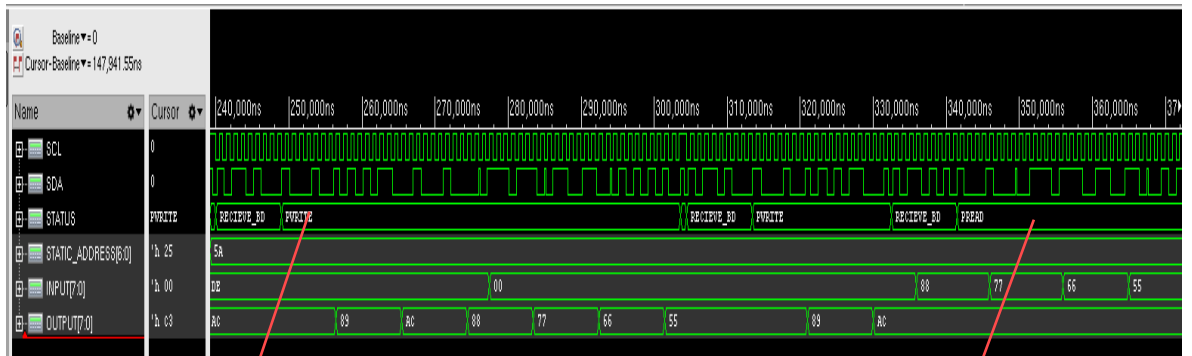


Fig4.8: Waveform

c

d

we can observe from fig 4.8 that at point

- (a) Private Write transaction is happening at device address 25
- (b) private read transaction is happening at device address 25
- (c) Private Write transaction is happening at device address 5A
- (d) private read transaction is happening at device address 5A.

4.3 Objective : Perform SETAASA CCC command.

Create a UVM Sequence to assign Dynamic Address to all I3c slaves, then use SETAASA (CCC Command to Set All Addresses to Static Addresses). The flow of test is shown in fig 4.9 below.

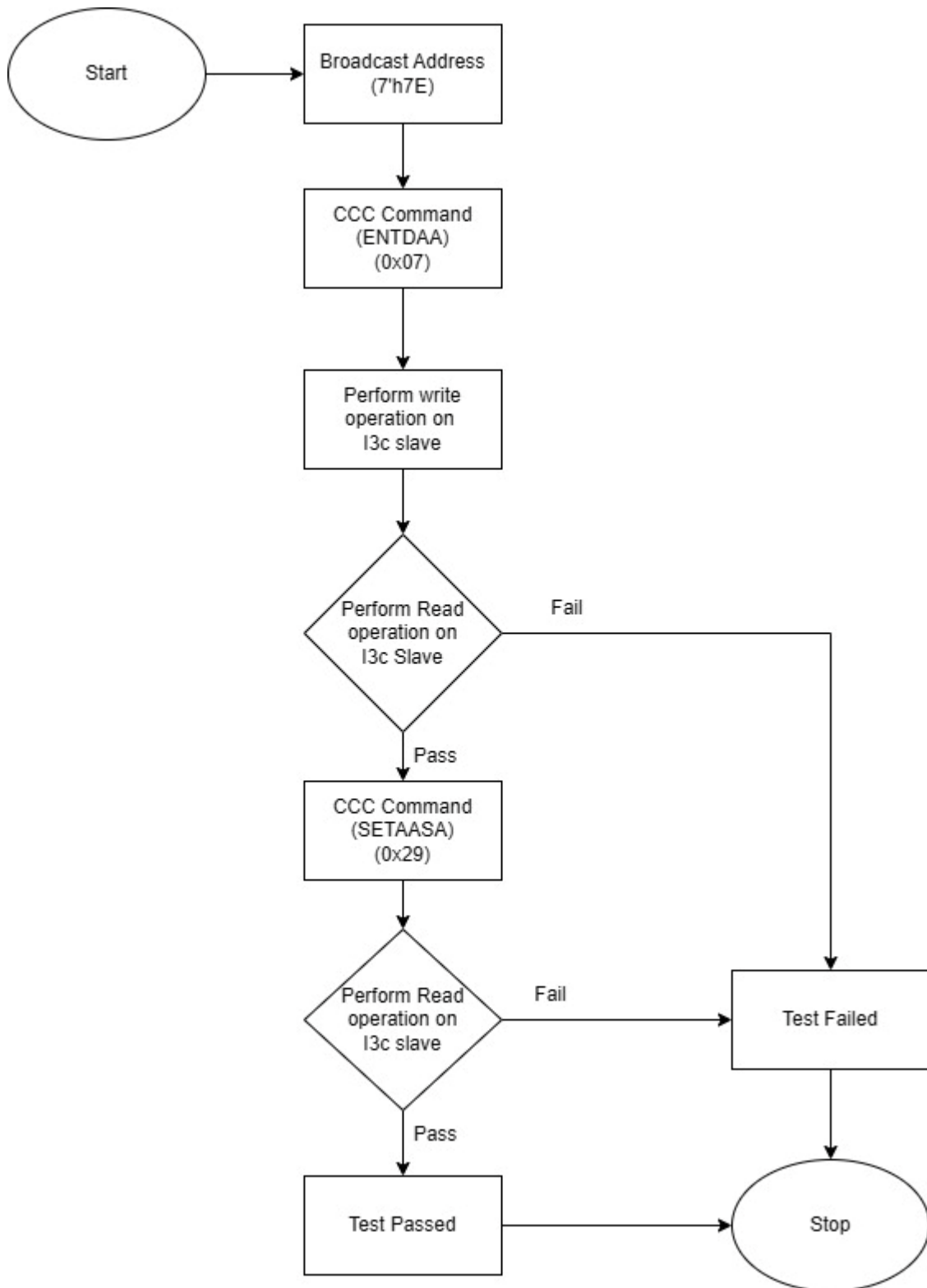


Fig4.9: Flow of Test 3

Results:

```

UVM_INFO : 92
UVM_WARNING : 1
UVM_ERROR : 0
UVM_FATAL : 0
Simulation status : Pass
  
```

Fig 4.10: Output Log

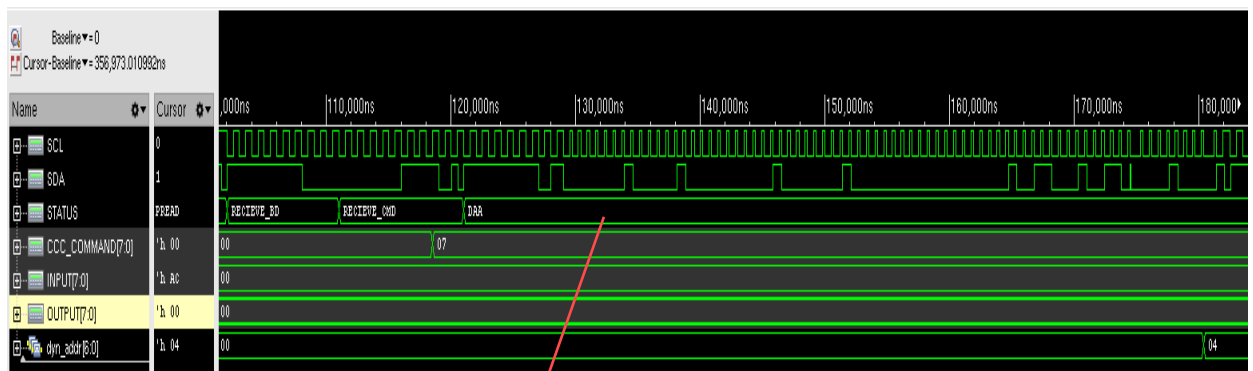
Prot	Kind	RW	DevID	Status	DataLen	Data

					CCC^M	

SDR	CCC	----	0X04	PASS	ENTDAA	0208010100002626
SDR	CCC	----	0X05	PASS	ENTDAA	07f800000cc1603
SDR	CCC	----	0X06	PASS	ENTDAA	07f800000cd1603
SDR	PRV	WRITE	0X04	PASS	6	05ac3412cdab
SDR	PRV	WRITE	0X04	PASS	2	05ac
SDR	PRV	READ	0X04	ABORT	4	3412cdab
SDR	CCC	WRITE	BCAST	PASS	SETAASA	
SDR	PRV	WRITE	0X04	PASS	2	05ac

Fig4.11: Tracker

From the above fig 4.10 we can observe that UVM is being used to develop the test bench and required test is passed successfully. We can observe from fig 4.11 that dynamic address is allotted to all slaves successfully, then I3c write and read operations are performed, then we are sending SETAASA CCC command and finally we are performing read operation again.



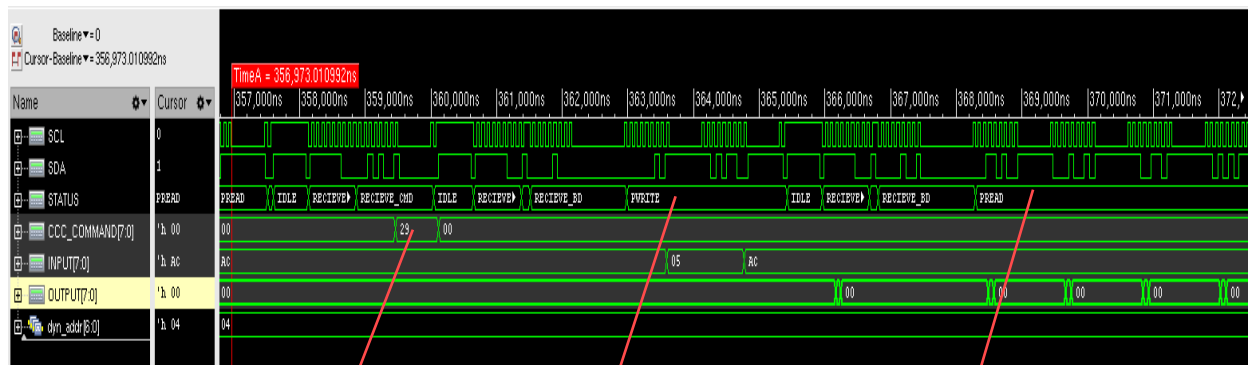


Fig4.12: Waveform

b

c

d

From fig 4.12 we can observe that at point
 (a) Dynamic Address Addressing is Happening
 (b) we are send CCC Command for SETAASA after that
 (c) and (d) we are performing write and read operations.

4.4 Objective :Perform I3c Write and Read operations

Create a UVM Sequence to assign Dynamic Address to all I3c slaves, then perform READ Write operations on i3c Slave. Flow of test is shown in Fig 4.13 below.

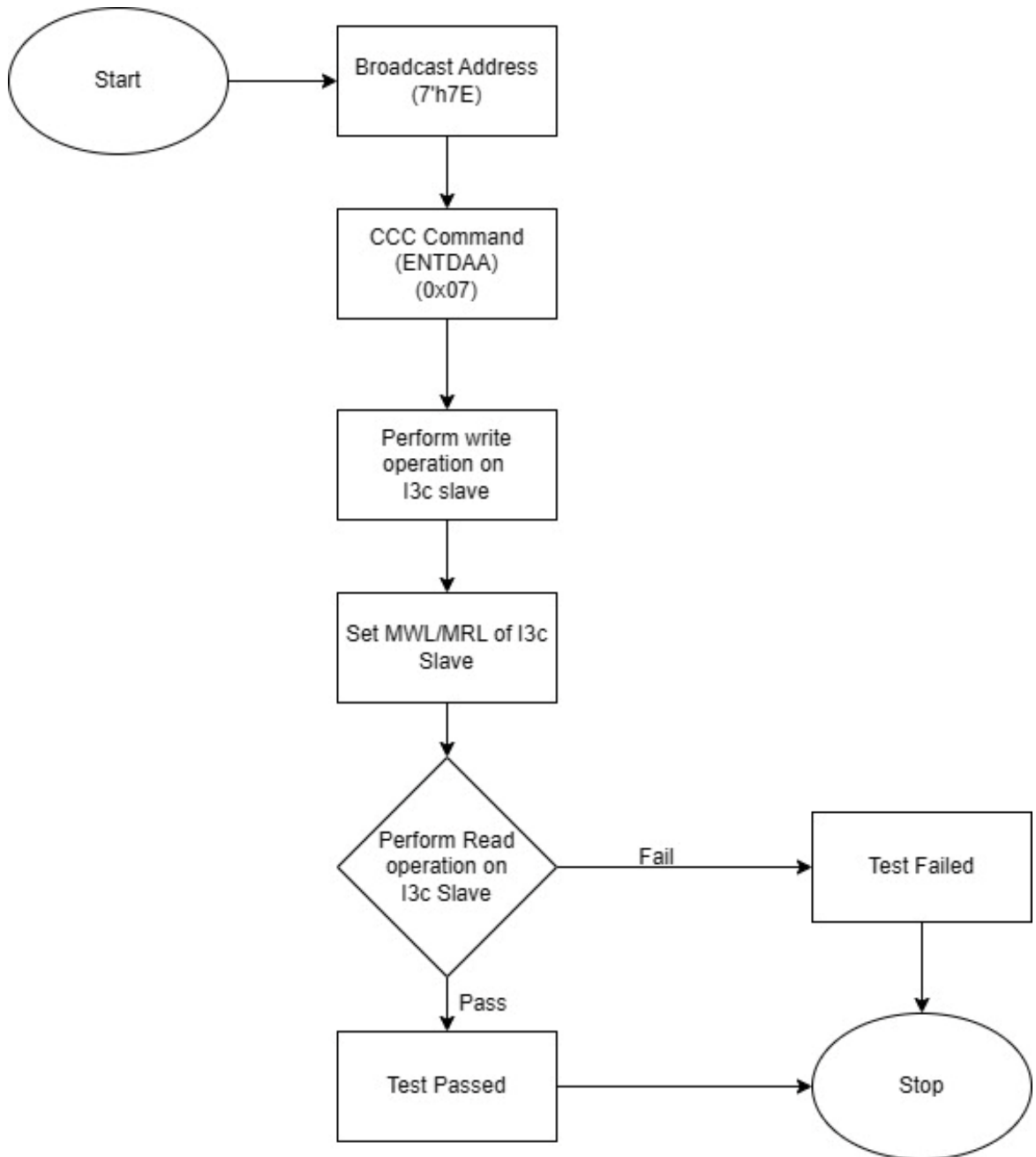


Fig4.13: Flow of Test 4

Results:

```

UVM_INFO : 91
UVM_WARNING : 1
UVM_ERROR : 0
UVM_FATAL : 0
Simulation status : Pass
  
```

Fig 4.14: Output Log

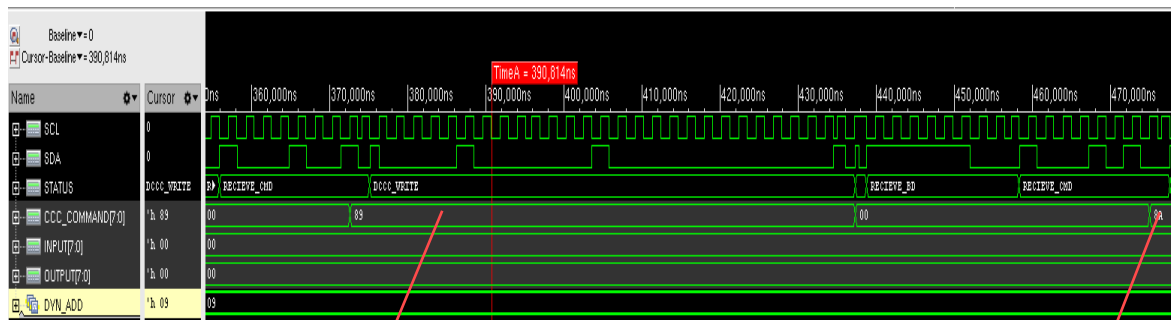
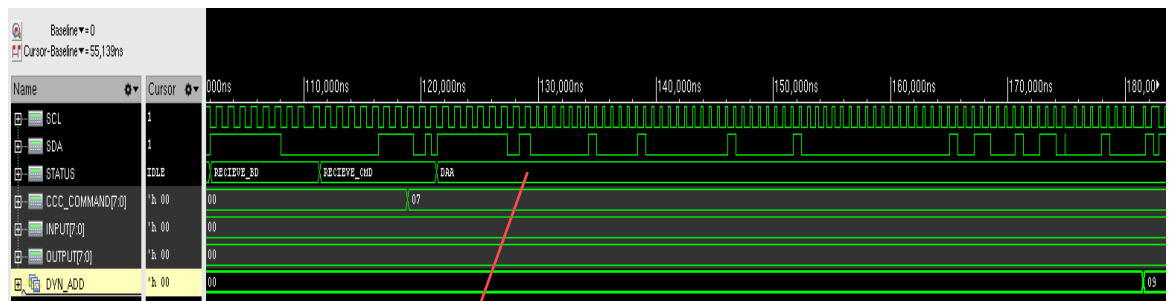
Prot	Kind	RW	DevID	Status	DataLen	Data

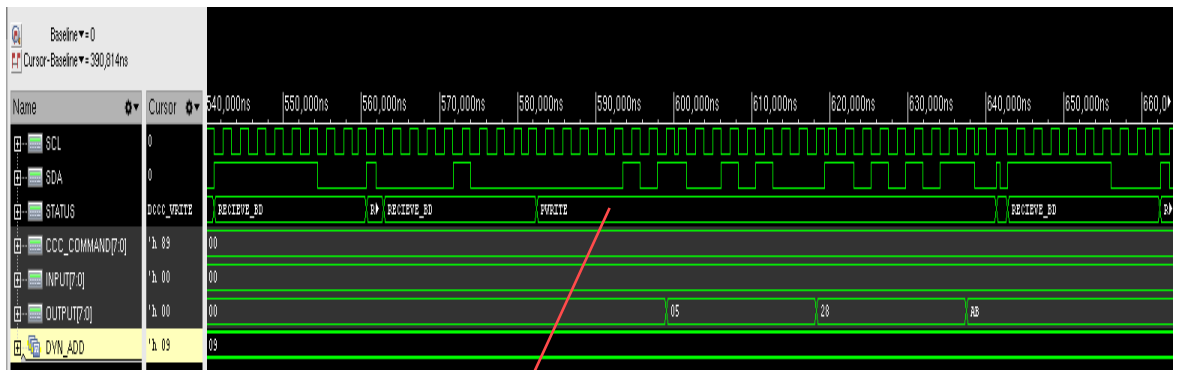
CCC^M						

SDR	CCC	----	0X04	PASS	ENTDAA	0208010100002626
SDR	CCC	----	0X05	PASS	ENTDAA	07f8000000cc1603
SDR	CCC	----	0X06	PASS	ENTDAA	07f8000000cd1603
SDR	CCC	----	----	----	SETMML^M	
SDR	CCC	WRITE	0X04	PASS	SETMML	1000
SDR	CCC	----	----	----	SETMRL^M	
SDR	CCC	WRITE	0X04	PASS	SETMRL	2000
SDR	PRV	WRITE	0X04	PASS	3	0528ab
SDR	PRV	WRITE	0X04	PASS	2	0528

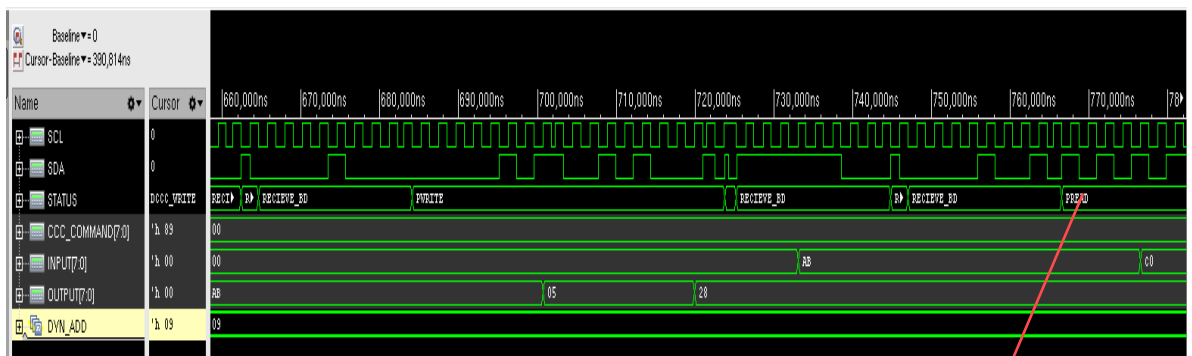
Fig4.15: Tracker

From the above fig 4.14 we can observe that UVM is being used to develop the test bench and required test is passed successfully. We can observe from fig 4.15 that dynamic address is allotted to all slaves successfully, then Max Write Length and Max Read Length is set to 1000 and 2000 respectively. After that write and Read Operations are Performed on I3c Slave controller.





d



e

Fig4.16: Waveform

- We can observe from fig 4.16 that at point
- (a) Dynamic Address Addressing is Happening
 - (b) Max Write Length CCC command was sent
 - (c) Max Read Length CCC command was sent
 - (d) write operation is happening
 - (e) read operation is happening.

4.5 Objective : Perform IBI functionality

Create a UVM Sequence to check In Band Interrupt Function. Flow of test is shown in Fig 4.17 below.

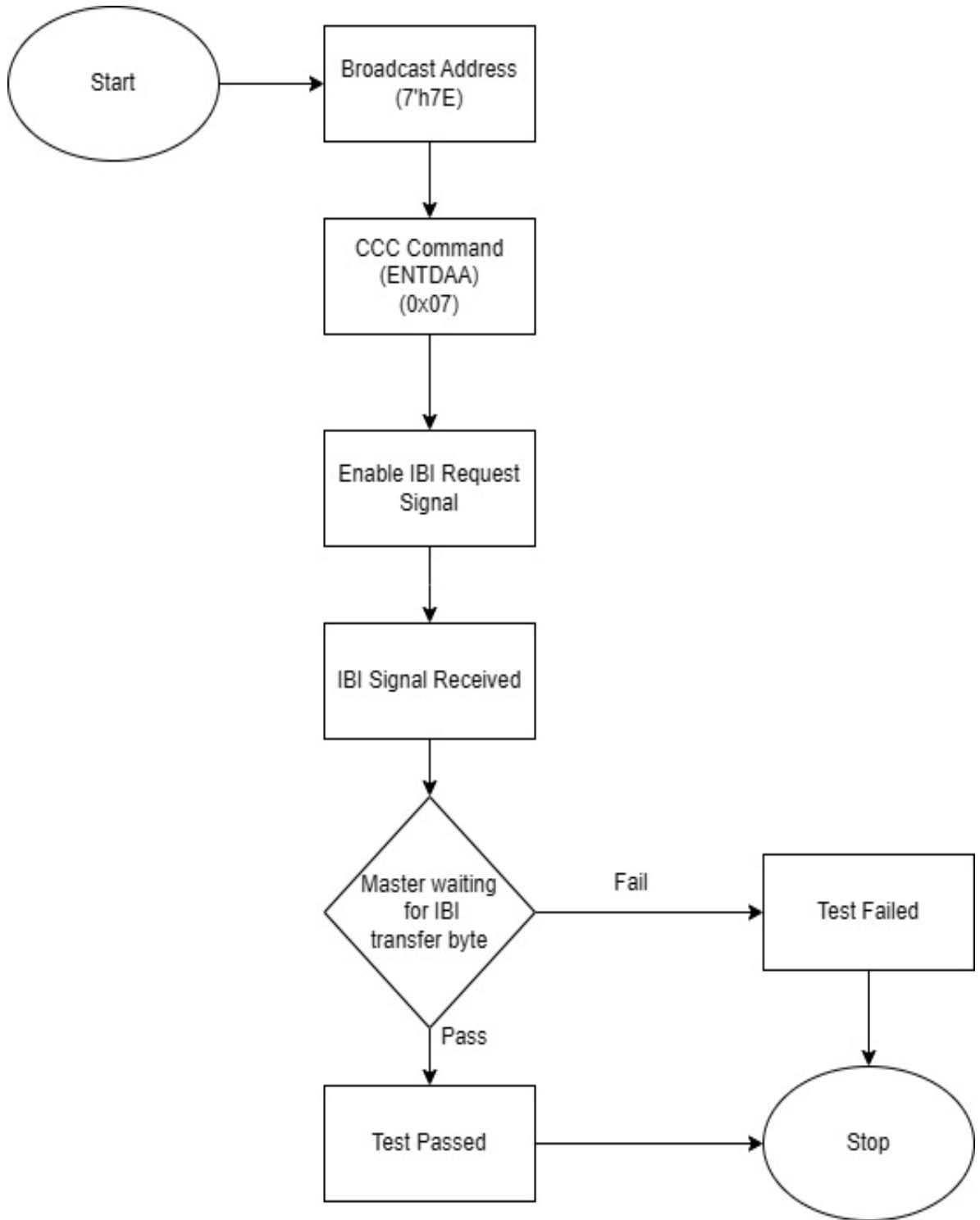


Fig4.17: Flow of Test 5

Results:

```

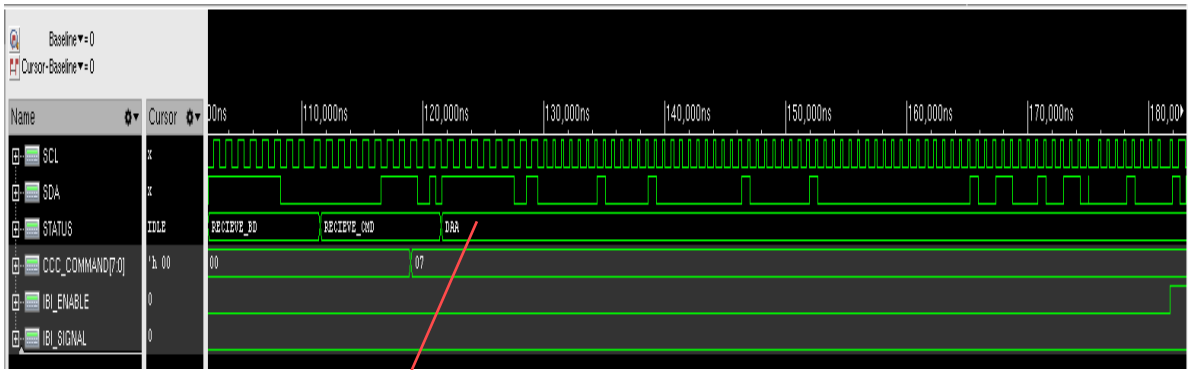
report counters by bus
UVM_INFO : 90
UVM_WARNING : 2
UVM_ERROR : 0
UVM_FATAL : 0
Simulation status : Pass
    
```

Fig 4.18: Output Log

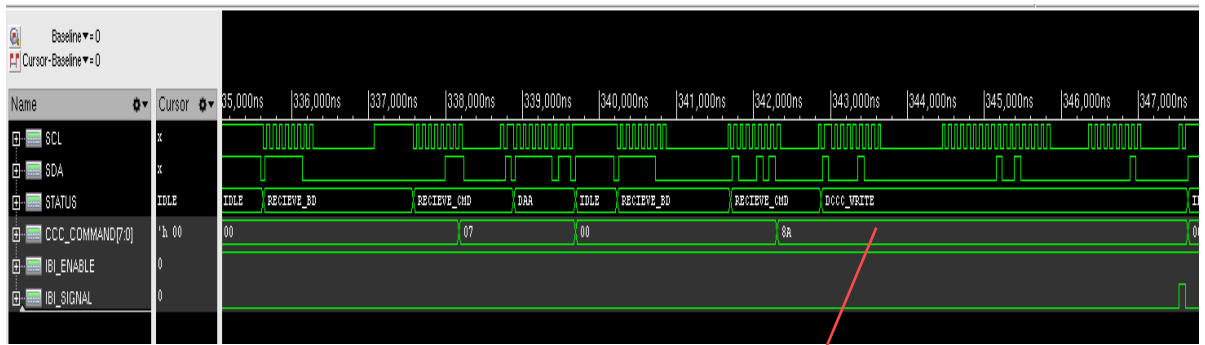
Prot	Kind	RW	DevID	Status	DataLen	Data
CCC^M						
SDR	CCC	----	0X04	PASS	ENTDAA	0208010500002626
SDR	CCC	----	0X05	PASS	ENTDAA	07f8000000cc1603
SDR	CCC	----	0X06	PASS	ENTDAA	07f8000000cd1603
SDR	CCC	----	----	----	SETMRL^M	
SDR	CCC	WRITE	0X04	PASS	SETMRL	222222
SDR	IBI	READ	0X04	PASS	2	3623

Fig4.19: Tracker

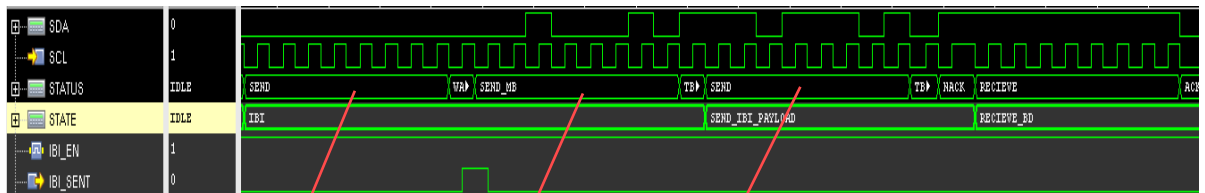
From the above fig 4.18 we can observe that UVM is being used to develop the test bench and required test is passed successfully. We can observe from fig 4.19 that dynamic address is allotted to all slaves successfully, then Max Read Length is sent to limit the payload value, after that IBI is sent.



a



b



c

d

e

Fig4.20: Waveform

We can observe from fig 4.20 that at point
 (a) Dynamic Address Addressing is Happening
 (b) Max Read Length CCC command was sent
 (c) IBI is sent , at point (d) IBI MDB is sent
 (e) IBI payload is sent.

4.6 Objective : Perform Superlint Design Checks

Perform Superlint Formal checks on Verification environment.

Results:

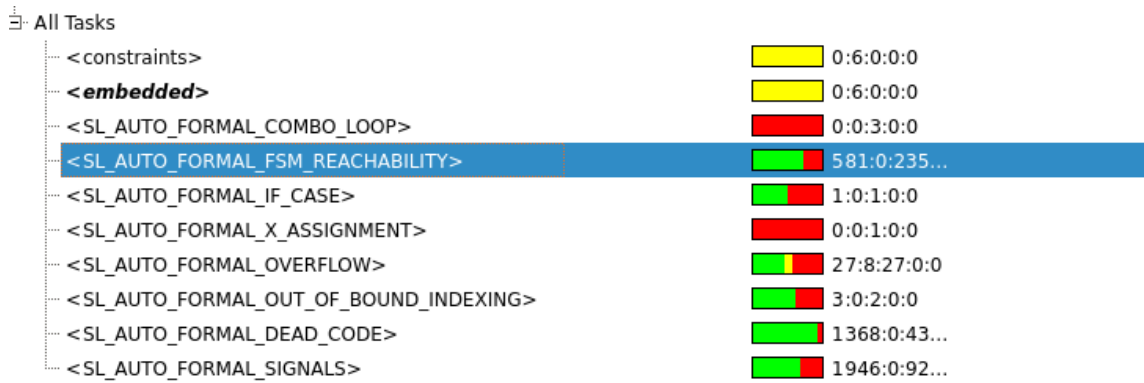


Fig4.21: Superlint Reports

4.7 Objective : Perform Connectivity Checks

Perform Connectivity Formal checks on Verification environment.

Results:

```
-----  
Summary  
-----  
Generated 7 connections  
  straightforward : 5  
  simple          : 0  
  conditional     : 0  
  complex        : 2
```

CO ▾	Connection Name	Source	Destination	Classification
	c_0	1'b0	ibi_sent_pulse_sync.ste_sync_...	tied to 0
	c_1	1'b0	ibi_sent_pulse_sync.ste_sync_...	tied to 0
	c_2	ibi_sent_pulse_sync....	ibi_sent_pulse_sync.ste_sync_...	none
	c_3	ibi_sent_pulse_sync....	ibi_sent_pulse_sync.ste_sync_...	none
	c_4	ibi_sent_pulse_sync....	ibi_sent_pulse_sync.ste_sync_...	none
	c_5	ibi_sent_pulse_sync....	ibi_sent_pulse_sync.out	none
	c_6:1_1	ibi_sent_pulse_sync....	ibi_sent_pulse_sync.out	none

Fig4.22: Connectivity Reports

Chapter 5

Conclusion

- The I3C protocol's robustness and dependability have been demonstrated by the successful completion of a thorough verification that included I2C write and read operations, SETAASA (Set All Addresses to Static Address), and ENTDAAs (Enter Dynamic Address Assignment).
- Each device on the I3C bus was accurately assigned a unique dynamic address, which is essential for preserving orderly communication and addressing within the network, according to the ENTDAAs test, which validated the dynamic address assignment mechanism.
- In a variety of applications, this dynamic addressing feature improves the I3C protocol's scalability and flexibility.
- The SETAASA test also verified that the protocol can return all devices to their static addresses, which is a necessary feature for preserving a reliable and consistent addressing scheme, especially in settings where changing the address dynamically is undesirable.
- The protocol's dependability in handling address configurations is demonstrated by the fact that all devices successfully executed the SETAASA command, returning them to their initial static addresses.
- To make sure the I3C protocol is backward compatible with older I2C devices, I2C write and read tests were also carried out.
- The results of these tests, which used the I3C master to write and read data to and from I2C devices, verified precise data transfers and appropriate acknowledgement signals, indicating smooth support for I2C transactions.
- In order to ensure seamless transitions and interoperability, this compatibility is essential for integrating I3C into current systems that use I2C devices. All things considered, the verification results show that the I3C implementation complies with protocol requirements and operates dependably across a range of functionalities.
- The I3C protocol's robustness and interoperability are confirmed by the successful completion of these tests, which makes it appropriate for integration into intricate systems that demand sophisticated communication capabilities.
- In the future, the protocol's robustness under harsh circumstances may be ensured by additional stress testing and performance optimisation. Its applicability may also be expanded by investigating new features and improvements.

REFERENCES

- [1] NXP, "I3C Interface Devices," *NXP Semiconductors*. [Online]. Available: <https://www.nxp.com/products/interfaces/ic-spi-i3c-interface-devices/i3c-interface-devices:I3C>.
- [2] Design and Reuse, "The Importance of VLSI Design Verification and Its Methodologies," *Design & Reuse*, Sep. 2023. [Online]. Available: <https://www.design-reuse.com/articles/54702/importance-of-vlsi-design-verification-and-its-methodologies.html>.
- [3] Truscova, "How to Use BugHunter Effectively: The Ultimate Guide," *Truscova Blog*. [Online]. Available: https://truscova.com/blog_article_21.php.
- [4] Cadence, "Cadence Design Systems," *Cadence Design Systems*. [Online]. Available: https://www.cadence.com/en_US/home.html.
- [5] S. N. Chauhan and G. K. Andurkar, "Development of UVM Testbench for I3C protocol," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India, July 6-8, 2023
- [6] A. Mahale and K. B. S., "Architecture Analysis and Verification of I3C Protocol," in *2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, 2019
- [7] Y. K. S. and R. Bhakthavatchalu, "Design and Implementation of MIPI I3C Master Controller SubSystems," in *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Karnataka, India, June 23-25, 2023
- [8] M. Golubic, J. Kundrata, and A. Baric, "Verification of the Legacy Compatibility of the MIPI I3C Master," in *2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO)*, Opatija, Croatia, Sep. 27 - Oct. 01, 2021
- [9] N. B. Harshitha, P. Kumar Y. G., and M. Z. Kurian, "An Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC's): A Review," in *Proceedings of the International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021)*, Tumkur, India, 2021
- [10] S. Bhattacharjee, D. Shah, and D. Pal, "Deploying Artificial Intelligence in Design Verification to Accelerate IP/SoC Sign-off with Zero Escape," in *Proceedings of the 2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS)*, Goa, India, 2024
- [11] G. M., G. Gopakumar, and D. D. S., "Formal and Simulation Verification: Comparing and Contrasting the Two Verification Approaches," in *Proceedings of the 2021 2nd International Conference on Advances in Computing, Communication, Embedded and Secure Systems (ACCESS)*, Ernakulam, India, 2021
- [12] A. O. Naik, E. Kuruvilla, and A. P. Chavan, "Integration and Verification of IP Cores on SoC," in *Proceedings of the 2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, Bengaluru, India, 2021

- [13] P. Gao, N. Xu, X. Zheng, J. Gong, and X. Zhong, "Design Implementation and Verification of a Flexible I3C Hardware Architecture," in Proceedings of the 2024 13th International Conference on Communications, Circuits and Systems (ICCCAS), Zhuhai, China, 2024
- [14] Purvi D. Mulani, "SoC Level Verification Using SystemVerilog," in Second International Conference on Emerging Trends in Engineering and Technology (ICETET-09), 2009
- [15] N. Sahay and S. Gajjar, "Design and UVM based Verification of UART, SPI, and I2C Protocols," in 2024 5th International Conference on Smart Electronics and Communication (ICOSEC), 2024
- [16] N. Bhuvaneshwary, J. Deny, and A. Lakshmi, "Exploration on Reusability of Universal Verification Methodology," in 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022
- [17] M. Sukhanya and K. Gavaskar, "Functional Verification Environment for I2C Master Controller using System Verilog," in 2017 4th International Conference on Signal Processing, Communications and Networking (ICSCN-2017), Chennai, India, March 2017
- [18] X. Wang, H. Ruan, and L. Zou, "Exploration of Using Direct Programming Interface to Improve the Reusability of Verification IP," in 2022 7th International Conference on Integrated Circuits and Microsystems (ICICM), 2022
- [19] W. Huang and G. Sheng, "Analysis and Research on UART Communication Protocol," in 2024 4th Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), 2024
- [20] J. Gao, J. Pang, G. Zhang, C. Huo, J. Yuan, and Z. Yin, "Research on Communication Protocol Adaptability of Power Internet of Things Application Layer," in 2022 5th International Conference on Energy, Electrical and Power Engineering (CEEPE), 2022
- [21] L. M. Kappaganthu, D. P. M, and A. Yadlapati, "I2C Protocol and its Clock Stretching Verification using System Verilog and UVM," in 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017
- [22] A. Kapoor, W. Ferguson, H. Jain, and S. Kundu, "Formal Verification of Floating-Point Division," in *2023 IEEE 30th Symposium on Computer Arithmetic (ARITH)*, 2023
- [23] M. Dossis, "Converging Formal Verification in a High-Level Synthesis Environment," in *2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, 2021
- [24] M. Rathmair, T. Haspl, T. Komenda, B. Reiterer, and M. Hofbauer, "A Formal Verification Approach for Robotic Workflows," in *2021 20th International Conference on Advanced Robotics (ICAR)*, Ljubljana, Slovenia, 2021
- [25] R. Wang, S. Liu, and Y. Sato, "A Fault Localization Approach Derived From Testing-based Formal Verification," in *2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS)*, Singapore, 2020

[26] F. Boutekkouk, "C Programs Formal Verification Using Maude," in *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Oum El Bouaghi, Algeria, 2024

I3C

ORIGINALITY REPORT

12%

SIMILARITY INDEX

4%

INTERNET SOURCES

10%

PUBLICATIONS

3%

STUDENT PAPERS

PRIMARY SOURCES

- 1** Shreyash Naresh Chauhan, Ganesh K Andurkar. "Development of UVM Testbench for I3C protocol", 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2023
Publication 1%
- 2** Jorge André Gastmaier Marques, Sergiu Arpadi, Maximilian Luppe. "Open-Source FPGA Implementation of an I3C Controller", Chips, 2025
Publication 1%
- 3** researchr.org
Internet Source 1%
- 4** "Table of Contents", 2022 7th International Conference on Integrated Circuits and Microsystems (ICICM), 2022
Publication <1%
- 5** Yadhu Krishnan S, Ramesh Bhakthavatchalu. "Design and Implementation of MIPI I3C master controller SubSystems", 2023 3rd International Conference on Intelligent Technologies (CONIT), 2023
<1%

Publication

6	www.ijraset.com Internet Source	<1%
7	Mario Golubic, Jurica Kundrata, Adrijan Baric. "Verification of the Legacy Compatibility of the MIPI I3C Master", 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), 2021 Publication	<1%
8	eprints.usm.my Internet Source	<1%
9	toc.proceedings.com Internet Source	<1%
10	"Table of contents", 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017 Publication	<1%
11	Jian Gao, Jianmin Pang, Ganghong Zhang, Chao Huo, Jianan Yuan, Zhibin Yin. "Research on Communication Protocol Adaptability of Power Internet of Things Application Layer", 2022 5th International Conference on Energy, Electrical and Power Engineering (CEEPE), 2022 Publication	<1%
12	Submitted to Heriot-Watt University Student Paper	<1%

13 Chin-Wen Liao, Hsiu-Chou Yu, Yu-Cheng Liao. "Verification of SPI Protocol Using Universal Verification Methodology for Modern IoT and Wearable Devices", Electronics, 2025

Publication

<1%

14 Soares, Mariana Catarina Pereira. "An Investigation Into User Expectations for Formal Verification", Universidade do Porto (Portugal), 2024

Publication

<1%

15 N. Bhuvaneswary, Muvvala Venkata Sai Syam Sundar, Gaddam Gangaiah, Penumuchu Rahul, Pichhapati Vishnu Vardhan Reddy. "10GB MAC CORE Verification - Scoreboard Module", 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 2023

Publication

<1%

16 Submitted to Fakultet elektrotehnike i računarstva / Faculty of Electrical Engineering and Computing

Student Paper

<1%

17 www.researchgate.net

Internet Source

<1%

18 Submitted to Concordia University

Student Paper

<1%

- | | | |
|----|---|------|
| 19 | Michael Rathmair, Thomas Haspl, Titanilla Komenda, Bernhard Reiterer, Michael Hofbauer. "A Formal Verification Approach for Robotic Workflows", 2021 20th International Conference on Advanced Robotics (ICAR), 2021
Publication | <1 % |
| 20 | Submitted to Colorado Technical University Online
Student Paper | <1 % |
| 21 | ieeexplore.ieee.org
Internet Source | <1 % |
| 22 | "Table of contents", 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019
Publication | <1 % |
| 23 | Submitted to Asia Pacific Institute of Information Technology
Student Paper | <1 % |
| 24 | www.freepatentsonline.com
Internet Source | <1 % |
| 25 | Dae-Won Moon, Seung-Hyun Pyo, Dae-Ki Hong, Otgonbayar Bataa, Erdenekhuu Norinpel. "Assertion-Based Verification of I2C Module Using SystemVerilog", Electronics, 2025
Publication | <1 % |

- 26 Submitted to Rochester Institute of Technology <1 %
Student Paper
-
- 27 1adfb5a7-a038-4413-bbae-af4ce0f91acc.filesusr.com <1 %
Internet Source
-
- 28 Gokulnath G, Jubilant J Kizhakkethottam. "A Study on Formal Verification of Smart Contracts in Distributed Ledger Technology", 2023 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), 2023 <1 %
Publication
-
- 29 Hongda Zhao, Zhe Wang, Mingxia Zhu, Xia Zhou, Yichen Li, Tengfei Zhang, Hangtao Yuan. "Application of 5G Communication Technology in Ubiquitous Power Internet of Things", 2020 Asia Energy and Electrical Engineering Symposium (AEEES), 2020 <1 %
Publication
-
- 30 Rong Wang, Shaoying Liu. "Branch Sequence Coverage Criterion for Testing-Based Formal Verification with Symbolic Execution", 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2019 <1 %
Publication
-

- | | | |
|----|---|------|
| 31 | Rufina Jasni S, Siva Mangai. N. M.. "Design of I2C Master Core with AHB Protocol for High Performance Interface", 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021
Publication | <1 % |
| 32 | polen.itu.edu.tr
Internet Source | <1 % |
| 33 | pure.tue.nl
Internet Source | <1 % |
| 34 | www.coursehero.com
Internet Source | <1 % |
| 35 | Ai Liu, Shaoying Liu. "Enhancing the Capability of Testing-Based Formal Verification by Handling Operations in Software Packages", IEEE Transactions on Software Engineering, 2022
Publication | <1 % |
| 36 | "Advances in Artificial Intelligence and Security", Springer Science and Business Media LLC, 2022
Publication | <1 % |
| 37 | "Ubiquitous Power Internet of Things Architecture Construction Plan", 2019 IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP), 2019
Publication | <1 % |

38 N B Harshitha, Y G Praveen Kumar, M Z Kurian. "An Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC's): A Review", 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021
Publication <1%

39 "Table of Contents", 2021 IEEE Mysore Sub Section International Conference (MysuruCon), 2021
Publication <1%

40 Giacomo Piscitelli. "AnWeb", Proceedings of the 14th international conference on Software engineering and knowledge engineering - SEKE 02 SEKE 02, 2002
Publication <1%

Exclude quotes Off Exclude matches Off
Exclude bibliography Off

