

PROJECT 1

COMMUNICATION CHIP

COMPLETE DFT DOCUMENTATION

Presented by

A. Siva Nagi Reddy

CONTENTS

1. Introduction
2. Problem Statement and Motivation
3. Design Specifications
4. Functional Architecture Description
5. Hierarchical Design Breakdown
 - 5.1 CoreA Description
 - 5.2 CoreB Description
6. Flip-Flop Count Estimation Methodology
7. Clock Architecture and Constraints
 - 7.1 Functional Clocks
 - 7.2 Scan Clock
8. Scan Insertion Strategy
9. Scan Chain Architecture
 - 9.1 CoreA Scan Chain Configuration
 - 9.2 CoreB Scan Chain Configuration
10. Scan Compression (Embedded Deterministic Test – EDT)
11. Automatic Test Pattern Generation (ATPG) Flow
 - 11.1 Fault Models Targeted
 - 11.2 ATPG Modes
 - 11.3 Fault Coverage Analysis
12. Memory Built-In Self-Test (MBIST)
13. Verification and Simulation
 - 13.1 Scan Design Rule Checks (DRC)
 - 13.2 Scan Pattern Simulation
14. Challenges and Debugging
15. Tools Used and Their Purpose
16. Final Outcome and Conclusions

1. Introduction

The Level-3 Communication Chip project represents a mid-complexity digital integrated circuit designed to emulate real-world communication subsystems commonly used in networking equipment, data transmission interfaces, and embedded control applications. Communication chips of this category typically consist of multiple reusable intellectual property (IP) blocks integrated hierarchically to perform data processing, control, and coordination functions. As device complexity increases, ensuring manufacturability and testability becomes a critical design requirement rather than an afterthought.

The primary objective of this project was to implement a complete end-to-end Design for Testability (DFT) methodology that transforms a purely functional design into a production-ready, testable silicon implementation. This involved incorporating scan-based testing techniques, managing multiple clock domains, applying scan compression to reduce test cost, and validating the design through industry-standard verification flows.

This project closely simulates a production-like semiconductor environment where multiple IP cores operate under different functional clocks and are integrated into a parent block. The DFT solution had to address real engineering challenges such as hierarchical scan insertion without breaking IP boundaries, safe interaction between asynchronous clock domains, efficient ATPG pattern generation, and robust verification through simulation. As a result, this project provides hands-on exposure to the same DFT challenges faced in real semiconductor companies.

2. Problem Statement and Motivation

Modern communication chips contain thousands to millions of transistors and a significant number of sequential elements such as flip-flops and registers. These sequential elements store state information that cannot be directly controlled or observed from primary inputs and outputs. Without proper test structures, internal defects introduced during fabrication—such as stuck-at faults, transition delay faults, or bridging faults—remain undetected, leading to low manufacturing yield and unreliable products.

The motivation behind this project stems from the fundamental limitations of functional testing. Functional tests are insufficient to cover all internal fault scenarios, especially in deeply sequential designs with limited controllability and observability. DFT techniques, particularly scan-based testing, overcome these limitations by converting internal storage elements into controllable and observable structures.

Additionally, hierarchical chip designs introduce further complexity. IP reuse is common in modern SoCs, and each IP block may be developed independently. The DFT solution must therefore support hierarchical integration without violating IP encapsulation. Another key motivation was scalability—ensuring that the DFT architecture could support future design growth without major rework.

By addressing these challenges, this project demonstrates how DFT enables high fault coverage, reduces test application time, lowers tester memory requirements, and ensures long-term scalability of the design.

3. Design Specifications

The Level-3 Communication Chip was designed as a hierarchical digital system with moderate complexity, making it ideal for demonstrating practical DFT methodologies without oversimplification. The design follows a parent–child core structure, where a top-level integration block manages multiple identical functional sub-blocks.

The approximate flip-flop count of the design is around 3720, which places it in the mid-range complexity category. This size is large enough to necessitate scan compression but small enough to allow clear analysis of scan architecture and ATPG behavior. The design operates under three independent functional clock domains, reflecting real communication systems where different modules operate at different frequencies. In addition, a dedicated scan clock is used exclusively for scan shift operations.

The chosen DFT methodology includes full scan insertion combined with scan compression using Embedded Deterministic Test (EDT). This approach balances test coverage, test time, and resource usage, aligning with industry-standard practices for communication-class chips.

- Design Type: Communication Chip (Level-3)
- Architecture: Hierarchical (Parent–Child Core structure)
- Approximate Flip-Flop Count: ~3720
- Clock Domains: 3 functional + 1 scan clock
- DFT Methodology: Full Scan + Scan Compression

4. Functional Architecture Description

The functional architecture of the design is organized hierarchically to reflect modular design principles used in real semiconductor products. At the top level, the design consists of a parent block named CoreB. CoreB is responsible for system-level coordination, control logic, and integration of child blocks. Within CoreB, four identical instances of a child block named CoreA are instantiated.

Each CoreA block represents a reusable communication sub-module that performs specific data processing or control functions. By replicating CoreA blocks, the design demonstrates IP reuse, which is a common strategy to reduce development time and improve reliability in large-scale designs.

CoreB manages arbitration, routing, and additional control functions required to coordinate the operation of multiple CoreA instances. This hierarchical organization introduces challenges in DFT implementation, particularly in maintaining scan chain continuity while preserving module boundaries. The architecture therefore serves as an excellent platform to demonstrate hierarchical scan insertion and stitching techniques.

5. Hierarchical Design Breakdown

5.1 CoreA Description

CoreA is a medium-sized functional block containing a mix of combinational and sequential logic. It includes data path components responsible for processing signals, finite state machines (FSMs) that control operational behavior, internal registers that store intermediate values, and interface logic for communication with CoreB. From a DFT perspective, CoreA represents a reusable IP block that must remain largely self-contained.

Each CoreA block includes its own scan chains, allowing it to be tested independently or as part of the larger system. This modular scan design ensures that CoreA can be reused in other designs with minimal modification, which is a critical requirement in industrial IP development.

5.2 CoreB Description

CoreB functions as the integration layer of the design. It contains control logic that coordinates the operation of the four CoreA instances, additional registers and pipeline stages, and scan stitching logic that connects individual CoreA scan chains into a coherent top-level scan architecture. CoreB also plays a key role in clock and reset distribution.

From a DFT standpoint, CoreB is responsible for managing scan enable signals, scan clock distribution, and ensuring that scan chains across different hierarchical levels operate correctly during shift and capture phases.

6. Flip-Flop Count Estimation Methodology

Accurate estimation of the number of flip-flops in a design is essential for planning the scan architecture, determining scan chain length, and selecting an appropriate compression ratio. In this project, flip-flop counts were estimated based on scan chain configuration rather than purely RTL statistics, providing a more test-centric perspective.

Each CoreA block contains approximately 675 flip-flops, distributed across internal scan chains. With four CoreA instances, the total contribution from CoreA blocks is approximately 2700 flip-flops. CoreB adds an additional 1020 flip-flops, primarily associated with control logic and integration functionality.

The total flip-flop count of approximately 3720 informed key DFT decisions, including the number of scan chains, scan chain length balancing, and the level of compression required to achieve efficient testing without excessive complexity.

- CoreA: ~675 flops each
- CoreB: ~1020 flops
- Total flops: ~3720

7. Clock Architecture and Constraints

7.1 Functional Clocks

The design operates using three independent functional clocks: CLKA, CLKB, and CLKC. CLKA operates at 100 MHz, CLKB at approximately 75 MHz, and CLKC at approximately 71 MHz. These clocks drive different functional domains within the design and are asynchronous with respect to each other.

From a DFT and timing analysis perspective, it is essential to properly constrain these clocks to prevent incorrect timing analysis across unrelated domains. Special care was taken to avoid unintended timing paths between asynchronous clock domains during both functional and scan operation.

- CLKA: 10 ns (100 MHz)
- CLKB: 13.2 ns (~75 MHz)
- CLKC: 14 ns (~71 MHz)

7.2 Scan Clock

A dedicated scan clock operating at 25 MHz was introduced exclusively for scan shift operations. This lower frequency reduces power consumption during scan shifting and ensures reliable operation of long scan chains. The scan clock is isolated from functional clocks using clock multiplexing controlled by scan enable signals.

- Scan Clock: 25 MHz

8. Scan Insertion Strategy

A full scan methodology was adopted in this project, meaning that all functional flip-flops were converted into scan-enabled flip-flops. This approach maximizes controllability and observability, enabling high fault coverage during ATPG.

Scan insertion was performed hierarchically to preserve CoreA and CoreB boundaries. Scan enable signals, scan clock routing, and reset behavior were carefully designed to ensure that scan operation does not interfere with functional behavior. This hierarchical approach also simplifies debug and reuse of individual IP blocks.

9. Scan Chain Architecture

The scan chain architecture was carefully designed to balance chain length, minimize routing complexity, and support efficient compression.

Within CoreA, 15 internal scan chains were created, each with a length of 45 flip-flops. In addition, 5 external scan chains with 27 flip-flops each were implemented. This configuration supports an 8× compression ratio.

Within CoreB, 15 internal scan chains with 68 flip-flops each and 5 external scan chains with 27 flip-flops each were implemented, enabling a 10× compression ratio. These chains were stitched at the top level to form the complete scan architecture.

10. Scan Compression (EDT)

Embedded Deterministic Test (EDT) was implemented to reduce test data volume and overall test application time. Without compression, the number of ATPG patterns and the amount of tester memory required would be significantly higher.

EDT works by compressing test stimuli at the tester interface and decompressing them on-chip, while compacting response data before sending it back to the tester. In this project, EDT logic was inserted at the top level, ensuring compatibility with the hierarchical scan architecture while maintaining high fault coverage.

11. ATPG Flow

Automatic Test Pattern Generation (ATPG) was performed targeting both stuck-at faults and transition delay faults. Patterns were generated in compressed mode to take advantage of EDT and in bypass mode for debug and validation.

The ATPG process achieved fault coverage greater than 95%, which meets industry expectations for communication-class designs. Coverage analysis helped identify and resolve any remaining testability issues.

12. MBIST

Memory Built-In Self-Test (MBIST) was implemented for all embedded memory blocks in the design. MBIST allows memories to be tested independently of logic scan, ensuring thorough coverage of memory-specific faults.

March-based test algorithms were used to detect common memory faults such as stuck-at, transition, and coupling faults. MBIST operation was isolated from normal scan operation to avoid interference.

13. Verification and Simulation

Verification played a critical role in validating the DFT implementation. Scan Design Rule Checks (DRC) were performed to ensure proper scan connectivity, clocking, and reset behavior. Any violations were analyzed and corrected.

Scan pattern simulations were performed using QuestaSim to validate scan shift, capture, and response behavior. This step ensured that ATPG patterns function correctly in a real simulation environment.

14. Challenges and Debugging

Several challenges were encountered during the project, including managing multiple asynchronous clock domains, balancing scan chain lengths, and resolving scan DRC violations. These challenges were addressed through careful constraint definition, scan architecture refinement, and iterative verification.

Debugging these issues provided valuable insight into real-world DFT problem-solving and reinforced the importance of planning and verification.

15.TOOLS USED AND THEIR PURPOSE

The following Electronic Design Automation (EDA) tools were used during the Design for Testability (DFT) implementation of the Level-3 Communication Chip project.

1. Tessent Scan (Mentor Graphics)

Purpose:

- Insertion of scan-enabled flip-flops
- Definition and creation of scan chains
- Hierarchical scan insertion at CoreA and CoreB levels

- Scan stitching across hierarchical boundaries
- Scan Design Rule Checks (DRC)
- Verification of scan enable, scan clock, and reset behavior

2. Tessent TestKompress (EDT)

Purpose:

- Implementation of scan compression using Embedded Deterministic Test
- Reduction of test data volume and tester memory usage
- Generation of compressed scan chains
- Support for high fault coverage with fewer ATPG patterns
- Decompression of scan stimuli and compaction of scan responses

3. Tessent ATPG

Purpose:

- Automatic Test Pattern Generation for scan-based testing
- Generation of test patterns for stuck-at faults
- Generation of test patterns for transition delay faults
- Support for both compressed and bypass ATPG modes
- Fault simulation and coverage analysis

4. Tessent MBIST

Purpose:

- Insertion of Memory Built-In Self-Test logic
- Testing of embedded SRAM and memory blocks
- Execution of March-based memory test algorithms
- Detection of memory-specific faults such as stuck-at and coupling faults
- Isolation of memory testing from logic scan testing

5. QuestaSim (Mentor Graphics)

Purpose:

- Functional simulation of RTL and gate-level netlists
- Verification of scan shift and capture operations
- Simulation of ATPG patterns
- Debugging scan-related issues
- Validation of DFT correctness before sign-off

6. Static Timing Analysis (STA) Tool (PrimeTime – basic exposure)

Purpose:

- Verification of timing constraints for functional clocks
- Validation of scan clock timing
- Identification of setup and hold violations
- Ensuring safe timing across clock domains during scan operation

7. Scripting Languages (TCL / Python)

Purpose:

- Automation of DFT flows
- Scan chain analysis and reporting
- Pattern handling and log analysis
- Reducing manual effort and improving repeatability

16. Final Outcome

The Level-3 Communication Chip design was successfully transformed into a fully testable, industry-ready implementation. The final design achieved high fault coverage, efficient test compression, and robust verification results. This project demonstrates practical expertise in hierarchical DFT implementation and serves as a strong foundation for real-world DFT engineering roles.