An Industry Oriented Mini Project Report On

## DESIGN OF CARRY SELECT ADDER WITH ONLINE TESTABILITY USING REVERSIBLE GATES

## Submitted to

## Partial fulfillment of the requirements of the degree of Bachelor of Technology

## In

**ELECTRONICS AND COMMUNICATION ENGINEERING**

## By

### B.CHARAN KUMAR 23RA5A0404

### Under the Guidance of

### Mrs. Y. VISHWA SRI

### Associate Professor

### 

### 

### KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY

### (UGC Autonomous Institution, Govt. of India)

### (Affiliated to JNTUH, Approved by AICTE, NBA &NAAC with ‘A’ Grade)

### Ghanpur(V) , Ghatkesar(M), Medchal Dist-500088

### A.Y: 2024-2025

**KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

## Department of Electronics and Communication Engineering



### CERTIFICATE

This is to certify that the Industry Oriented Mini Project work entitled **“DESIGN OF CARRY SELECT ADDER WITH ONLINE TESTABILITY USING REVERSIBLE GATES ”** is submitted by **BHOGA CHARAN KUMAR-23RA5A0404** in partial fulfillment of the degree of Bachelor of Technology From the Department of Electronics and Communication Engineering during the academic year 2024-25.I certify further to the best of our knowledge the work reported is not part of any other project on the basis of which a degree or an award has been given on an earlier occasion to any other candidate. The result has been verified and found to be satisfactory.

Shaik Imamvali Y. Vishwa Sri Y. Vishwa Sri

Project Supervisor Internal Guide Head of the Department

Examiner/ Faculty Incharge Dr. S. Sreenath Kashyap

Principal

**ACKNOWLEDGEMENT**

I hereby express our sincere thanks and gratitude to Almighty, Parents, family members and friends. Without their un-sustained support, we could not have done this project.

My sincerest thanks to the **Shri. Kommuri Pratap Reddy,** Chairman, KPRES, Mr. **K. Prashanth Reddy,** Vice – Chairman, KPRES for their support. We are very much thankful to our Director **Prof . B. Sudheer Prem Kumar**, Principal **Dr. S Sreenath Kashyap** for supporting us in all aspects and guiding in all walks of life.

I profoundly thank Mrs. **Y. Vishwa Sri,** Associate Professor & Head of the Department of Electronics and Communication Engineering who has been an excellent guide and a great source of inspiration to our work.

I thank our Coordinator **Mr Shaik Imamvali** Associate Professor, ECE and Internal Guide **Mrs. Y. Vishwa sri**, Associate Professor for their valuable suggestions and technical Guidance and constant encouragement and support in carrying out our project. The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success. In this context, we would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased our task.

**B.CHARAN KUMAR (23RA5A0404)**

**ABSTRACT**

Now a days, Reversible Logic is winding up increasingly more Conspicuous innovation having its applications in different fields like Nanotechnology, Low Power CMOS, Optical Computing. Reversibility plays prominent role for building energy efficient circuits. In this paper, reversible carry select adder is proposed and the performance analysis is verified. Online testing feature is added to reversible carry select adder which detects single stuck-at faults in a circuit. The proposed design is better as compared to basic carry select adder.

we propose a novel design of a reversible carry select adder (CSA) integrated with online testability features to enhance fault detection and reliability in low-power digital circuits. Reversible logic is gaining significant attention due to its potential to reduce power dissipation by eliminating information loss, which is critical for emerging computing technologies.

The carry select adder, known for its fast addition by precomputing carry signals, is redesigned using reversible gates to minimize power and hardware complexity. Moreover, we incorporate an online testability scheme that enables real-time fault detection during normal operation without interrupting the functionality of the adder. The proposed design achieves a balance between speed, fault coverage, and resource utilization. Simulation results demonstrate that the reversible CSA with online testability significantly reduces power consumption while providing robust fault detection capabilities compared to conventional irreversible and offline testable designs.

#### **KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

**Department of Electronics & Communication Engineering**

**Vision of the Institute**

To emerge as a premier institute of high-quality professional graduates who can contribute to economic and social developments of the Nation.

### Mission of the Institute

|  |  |  |  |
| --- | --- | --- | --- |
| **Mission** | | **Statement** | |
| **IM1** | | To have holistic approach in curriculum and pedagogy through industry interface to meet the needs of global competency. | |
| **IM2** | | To develop student with knowledge, attitude, employability skills, entrepreneurship, research potential and professionally ethical citizens. | |
| **IM3** | | To contribute to advancement of engineering and technology that would help to satisfy societal needs | |
| **IM4** | | To preserve, promote cultural heritage, humanistic values and spiritual values thus helping in peace and harmony in the society. | |

#### **KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

**Department of Electronics & Communication Engineering**

### Vision of the Department

To impart quality technical education in Electronics & Communication with accent on creativity, innovation and research thereby producing competent engineers who can meet global challenges with societal commitment.

### Mission of the Department

|  |  |
| --- | --- |
| **Mission** | **Statement** |
| **DM1** | To impart quality education to students in Basic Sciences, Mathematics, Electronics and communication Engineering through innovative teaching- learning process. |
| **DM2** | To facilitate students to define, design, and solve engineering problems in the field of Electronics and Communications Engineering using various Electronic Design Automation (EDA) tools. |
| **DM3** | To encourage research culture among faculty and students thereby facilitating them to be creative and innovative through constant interaction with R & D organizations and Industry. |
| **DM4** | To include qualities, professional ethics and social responsibilities in students and faculty |

#### **KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

**Department of Electronics & Communication Engineering**

**Program Educational Objectives**

|  |  |
| --- | --- |
| **PEO1** | **PEO1:** Graduates with fundamental and advanced knowledge in Engineering Sciences, Mathematics and Engineering course of Electronics and Communication Engineering along with allied domains of Engineering will be able to exhibit the contemporary skills and become globally competent with a flair for life-long learning. |
| **PEO2** | **PEO2:** Graduates capable in design, develop creative and innovative technologies in the field of Electronics and Communication Engineering, enabling them to work in multi-disciplinary teams to meet the societal needs. |
| **PEO3** | **PEO3:** Graduates with professional ethics and values, positive attitude, effective communication skills and knowledge in cutting edge technologies will ensure sustainable engineering and able to demonstrate top notch technology in their professional practices. |

 **KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

**Department of Electronics & Communication Engineering**

**Program Outcomes (POs)**

**PO1. Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**PO2. Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**PO3. Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4. Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5. Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

**PO6. The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**PO7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

 **KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

**Department of Electronics & Communication Engineering**

**PO8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**PO9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**PO10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**PO12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change

******

#### **KOMMURI PRATAP REDDY INSTITUTE OF TECHNOLOGY**

**Department of Electronics & Communication Engineering**

### Program Specific Outcomes

### PSO1. Apply the knowledge of mathematics, basic sciences and Engineering Sciences with a multidisciplinary approach to analyze and design electronic circuits, communication systems using contemporary skills.

### PSO2. Conduct experiments with the state-of-the-art technology and carryout investigation in research as a team or individual and interpretant the data to synthesize various electronic and communication systems in the perspective of societal concerns, environment and economy with sustainable engineering practices.

### PSO3. Demonstrate professional ethics and implement the principles of project management, business and public policy in the design of a system and carrying out simulation studies, and eventually communicate with effective presentation for its implementation with a due note for the scope of further studies to continue life-long learning.

|  |  |
| --- | --- |
|  |  |

**CONTENTS PAGE NO.**

**Certificate** i

**Acknowledgement** iii

**Abstract** iv

**CHAPTER 1**

Introduction 1

**CHAPTER 2**

Literature Survey 2

**CHAPTER 3**

Analysis 5

Reversible gates 5

**CHAPTER 4**

Existing system 10

Programming language 10

**CHAPTER 5**

Proposed system 15

**CHAPTER 6**

Source code 19

**CHAPTER 7**

Results 23

**CHAPTER 8**

Conclusion 29

**CHAPTER 9**

Self-evaluation 30

Reference 32

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.No** | **TITLE** | **Page No.** |
| 1 | CNOT gate block diagram | 6 |
| 2 | CNOT gate circuit diagram | 6 |
| 3 | Toffoli gate block diagram | 7 |
| 4 | Toffoli gate circuit diagram | 7 |
| 5 | Fredkin gate block diagram | 8 |
| 6 | Fredkin gate circuit diagram | 8 |
| 7 | Peres gate block diagram | 9 |
| 8 | Peres gate circuit diagram | 9 |
| 9 | CSA Design 1 | 15 |
| 10 | CSA Design 2 | 16 |
| 11 | Design 3 | 17 |
| 12 | Design 4 | 17 |
| 13 | Design 5 | 18 |
| 14 | RCSA Design 2 | 22 |
| 15 | Circuit diagram of RCSA design 1 | 25 |
| 16 | Physical design of RCSA design 1 | 25 |
| 17 | Circuit diagram of RCSA design 2 | 26 |
| 18 | Physical design of RCSA design 2 | 26 |
| 19 | Output power | 27 |

**C H A P T E R**

1

**INTRODUCTION**

*This chapter discusses about the objectives, research gap, background and identification of problem*

Advanced scaling of metal-oxide-semiconductor (MOS) is required for the IC Industry. Due to this advancement in scaling. integration density increases and reliability of integrated circuit becomes an important issue.

Scaling also results in noise problems, which affects the performance of chips. In this competitive world, reliability of the product, especially the manufactured integrated circuits, plays a major role in the survival in the electronics industry. More research has been done by the industry to produce more reliable product.

Even after a lot of testing on the product to avoid the faulty product from reaching the consumer, there are still many marginally good products go to the consumer that may introduce malfunction. Concurrent detection of faults is required to keep their reliability in high level.

More precisely, it masks the fault and prevents. it from occurring in a subsequent clock cycle. Thus, the effect of the faults will not be carried to the next level of the logic circuit.

**C H A P T E R**

2

**LITERATURE SURVEY**

*This chapter discusses about a concise review of existing research, technologies, and methods relevant to the project topic.*

The advancement of arithmetic logic circuits has always been a vital area of research in the field of VLSI (Very Large-Scale Integration) and digital system design. Adders play a fundamental role in all computing and signal processing systems. In this context, optimizing the **Carry Select Adder (CSLA)** for speed, power, and fault detection is crucial. This literature survey explores key developments in CSLA architectures, **reversible computing**, and **online testability** to understand the background and motivation behind the proposed design.

**2.1 Carry Select Adder (CSLA)**

The Carry Select Adder is known for its **high-speed performance** among classical adder architectures. It achieves this by precomputing sum outputs for both possible carry-in values (0 and 1) and then selecting the correct result using a multiplexer. While this improves performance, it increases area and power consumption due to duplication of logic blocks.

**B. Ramkumar and Harish M Kittur (2011)** proposed an area-efficient CSLA using a **binary to excess-1 converter (BEC)** to reduce logic duplication. This design showed a balance between delay and area but lacked mechanisms for fault detection or energy recovery.

**2.2 Reversible Logic Gates**

Reversible computing has emerged as a promising solution to the growing concern of **power dissipation** in digital circuits, especially in quantum and low-power applications. In reversible logic, the number of inputs equals the number of outputs, and there is a one-to-one mapping, ensuring no information is lost—hence, no heat is generated as per **Landauer’s principle**.

Researchers like **R. Landauer (1961)** and **C.H. Bennett (1973)** laid the theoretical foundation for reversible computation. Reversible gates like **Peres gate, Fredkin gate, Toffoli gate**, and **Feynman gate (CNOT)** are widely used to build power-efficient arithmetic circuits.

Incorporating reversible logic into adder designs reduces energy loss and makes them suitable for **quantum computing**, **nanotechnology**, and **low-power VLSI systems**.

**2.3 Online Testability in Arithmetic Circuits**

Online testability refers to the ability of a circuit to **detect faults during normal operation** without interrupting functionality. This is essential for high-reliability systems such as aerospace, medical, and military applications.

Many prior works focused on **offline testing techniques**, requiring external hardware or pre-processing, which are not suitable for real-time applications. Recent efforts have introduced **built-in self-test (BIST)** and **parity-based checkers** for online fault detection in arithmetic units. However, these methods often increase circuit complexity and energy consumption.

**P. Balasubramanian (2017)** proposed online testable designs using reversible logic gates, highlighting that reversible circuits can detect specific faults like **bit-flip errors** using duplicated signals or parity checks without major overhead.

**2.4 Reversible Adders with Fault Detection**

Researchers have designed **fault-tolerant reversible adders** by combining standard reversible gates in innovative ways. Many of these designs aim to achieve a balance between:

* **Area** (gate count),
* **Power** (energy loss),
* **Speed** (propagation delay), and
* **Fault coverage** (detection of stuck-at, bridging, or transient faults).

The concept of **online testability using XOR-based error signals (E0, E1)**, as used in the proposed design, has shown promise in real-time fault detection.

**2.5 Summary of Literature Survey**

From the survey, it is evident that:

* Traditional CSLAs offer high speed but consume more power and area.
* Reversible logic reduces energy loss and is suitable for future computation models.
* Integrating online testability in arithmetic circuits enhances system reliability.
* There is a growing interest in **hybrid designs** that combine **reversible gates** with **online error detection mechanisms**.

Thus, the proposed project aims to combine these advantages by designing a **carry select adder with online testability using reversible gates**, offering **high-speed operation**, **low power dissipation**, and **built-in fault detection**.

**C H A P T E R**

3

**ANALYSIS**

*This chapter provides a brief review of existing methods, technologies, and logic designs related to the proposed project, focusing on their relevance to performance, efficiency, and fault detection.*

In concurrent error detectable adders, multi block carry select adder is implemented with half adders and increment circuit to execute a full adder circuit in block0. Let n be the length of the blocks.

From block 1 to block n-1, mux is used to select the correct result. In block 0, one row is for half adder and one row is for binary incrementer .

From block 1 to block n-1 blocks, there are 16 blocks because we implemented for 64-bit. From block 1 to block n-1, there will be one row of half adder, two rows of binary incrementer for taking 0 as the carry input and taking 1 as the carry input, and one row of mux for selecting a correct sum result. In our paper, half adder, increment and mux circuit is replaced.

We proposed an alternative circuit for half adder, incrementor, and mux circuit using reversible gates.

A 64-bit carry select adder and 32-bit carry select adder with reversible gate are designed in VLSI technology. This proposed work proves that reversible gates can also be used for concurrent detection of faults and masking the faults.

**CNOT GATE**

Controlled NOT gate (also C-NOT or CNOT), controlled-*X* gate, controlled-bit-flip gate, Feynman gate or controlled Pauli-X is a [quantum logic gate](https://en.wikipedia.org/wiki/Quantum_logic_gate) that is an essential component in the construction of a [gate-based](https://en.wikipedia.org/wiki/Quantum_circuit) [quantum computer](https://en.wikipedia.org/wiki/Quantum_computer). It can be used to [entangle](https://en.wikipedia.org/wiki/Quantum_entanglement) and disentangle [Bell states](https://en.wikipedia.org/wiki/Bell_state). Any quantum circuit can be simulated to an arbitrary degree of accuracy using a combination of CNOT gates and single [qubit](https://en.wikipedia.org/wiki/Qubit) rotations.

The first experimental realization of a CNOT gate was accomplished in 1995. Here, a single [Beryllium](https://en.wikipedia.org/wiki/Beryllium) ion in a [trap](https://en.wikipedia.org/wiki/Ion_trap#Trapped_ion_quantum_computer) was used. The two qubits were encoded into an optical state and into the vibrational state of the ion within the trap.

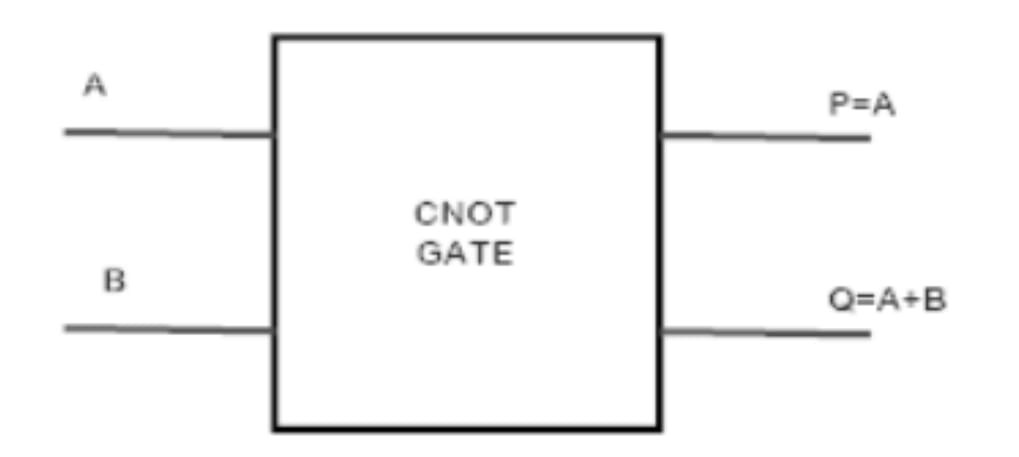
****

FIG 1: CNOT GATE BLOCK DIAGREM

Viewing CNOT in this basis, the state of the second qubit remains unchanged, and the state of the first qubit is flipped, according to the state of the second bit. (For details see below.) "Thus, in this basis the sense of which bit is the *control bit* and which the *target bit* has reversed. But we have not changed the transformation at all, only the way we are thinking about it.

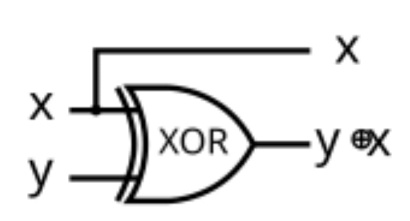


FIG 2 : CNOT GATE CIRCUIT DIAGRAM

**TOFFOLI GATE**

In [logic circuits](https://en.wikipedia.org/wiki/Logic_circuits), the Toffoli gate, also known as the CCNOT gate (“controlled-controlled-not”), invented by [Tommaso Toffoli](https://en.wikipedia.org/wiki/Tommaso_Toffoli) in 1980[[1]](https://en.wikipedia.org/wiki/Toffoli_gate#cite_note-Toffoli1980-1) is a [CNOT](https://en.wikipedia.org/wiki/CNOT) gate with two control bits and one target bit. That is, the target bit (third bit) will be inverted if the first and second bits are both 1. It is a universal reversible logic gate, which means that any classical [reversible circuit](https://en.wikipedia.org/wiki/Reversible_Circuit) can be constructed from Toffoli gates. There is also a [quantum-computing](https://en.wikipedia.org/wiki/Quantum_computing) version where the bits are replaced by [qubits](https://en.wikipedia.org/wiki/Qubit).

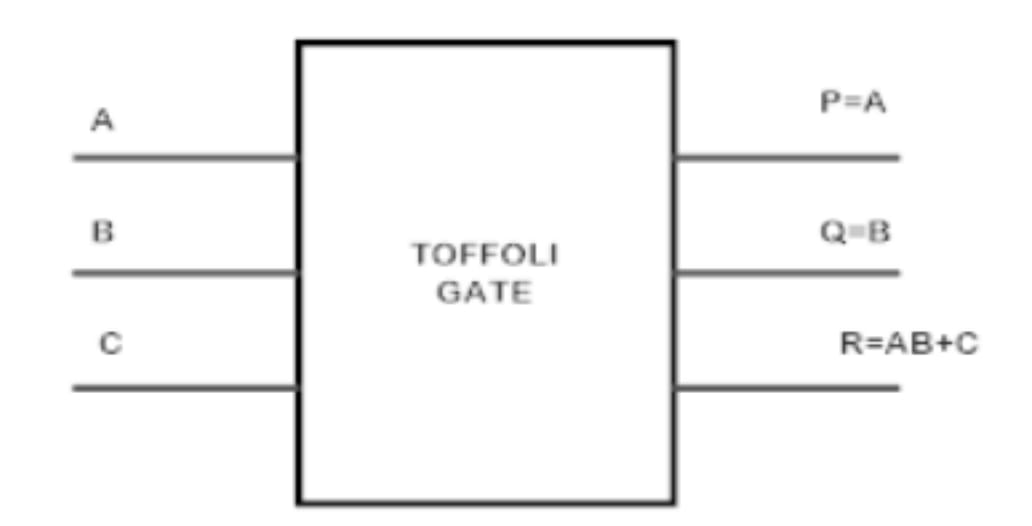
****

FIG 3: TOFFOLI GATE BLOCK DIGARAM

Any reversible gate that consumes its inputs and allows all input computations must have no more input bits than output bits, by the [pigeonhole principle](https://en.wikipedia.org/wiki/Pigeonhole_principle). For one input bit, there are two possible [reversible](https://en.wikipedia.org/wiki/Reversible_computing) gates. One of them is NOT. The other is the identity gate, which maps its input to the output unchanged. For two input bits, the only non-trivial gate (up to symmetry) is the [controlled NOT gate](https://en.wikipedia.org/wiki/Controlled_NOT_gate) (CNOT), which [XORs](https://en.wikipedia.org/wiki/XOR_gate) the first bit to the second bit and leaves the first bit unchanged.

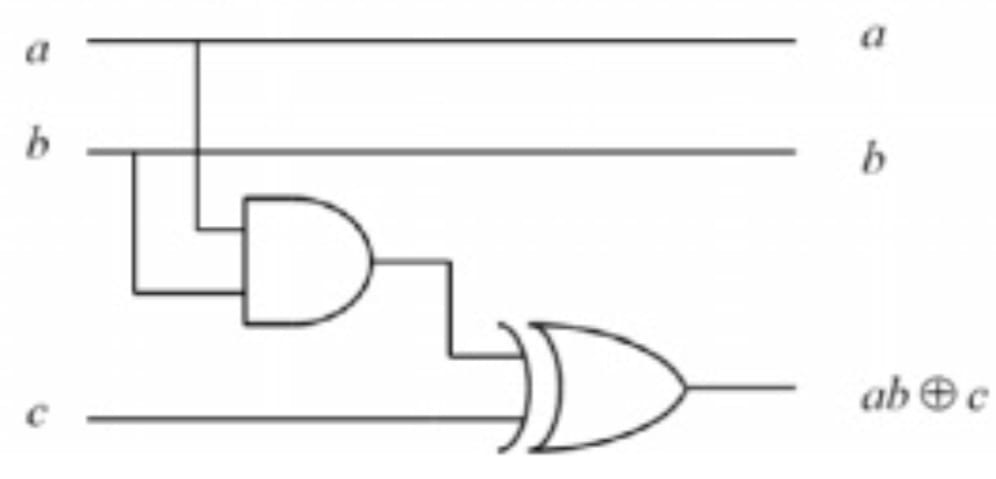


FIG 4: TOFFOLI GATE CIRCUIT DIAGRAM

**FREDKIN GATE**

The Fredkin is a computational circuit suitable for [reversible computing](https://en.wikipedia.org/wiki/Reversible_computing), invented by [Edward Fredkin](https://en.wikipedia.org/wiki/Edward_Fredkin). The Fredkin gate is a circuit or device with three inputs and three outputs that transmits the first bit unchanged and swaps the last two bits if, and only if, the first bit is 1.

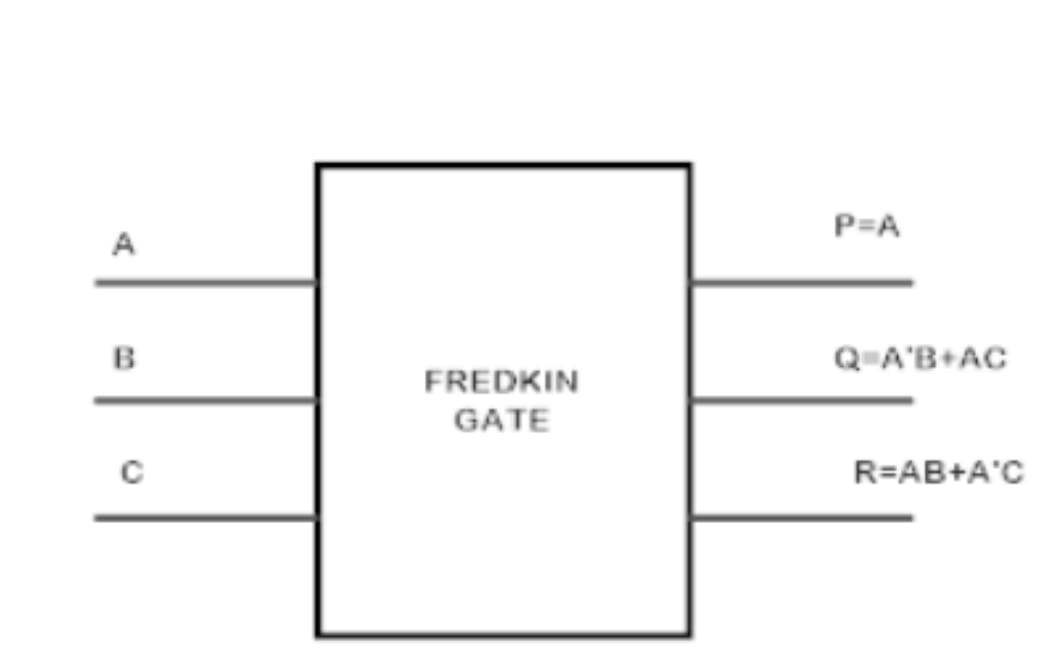
****

FIG 5: FREDKIN GATE BLOCK DIAGRAM

The basic Fredkin gate[[3]](https://en.wikipedia.org/wiki/Fredkin_gate#cite_note-3) is a [controlled](https://en.wikipedia.org/wiki/Quantum_gate#Controlled_gates) [swap gate](https://en.wikipedia.org/wiki/Quantum_gate#Swap_gate) (CSWAP gate) that [maps](https://en.wikipedia.org/wiki/Function_(mathematics)) three inputs (*C*, *I*1, *I*2) onto three outputs (*C*, *O*1, *O*2). The *C* input is mapped directly to the *C* output. If *C* = 0, no swap is performed; *I*1 maps to *O*1, and *I*2 maps to *O*2It is easy to see that this circuit is reversible, i.e., "undoes" itself when run backwards.

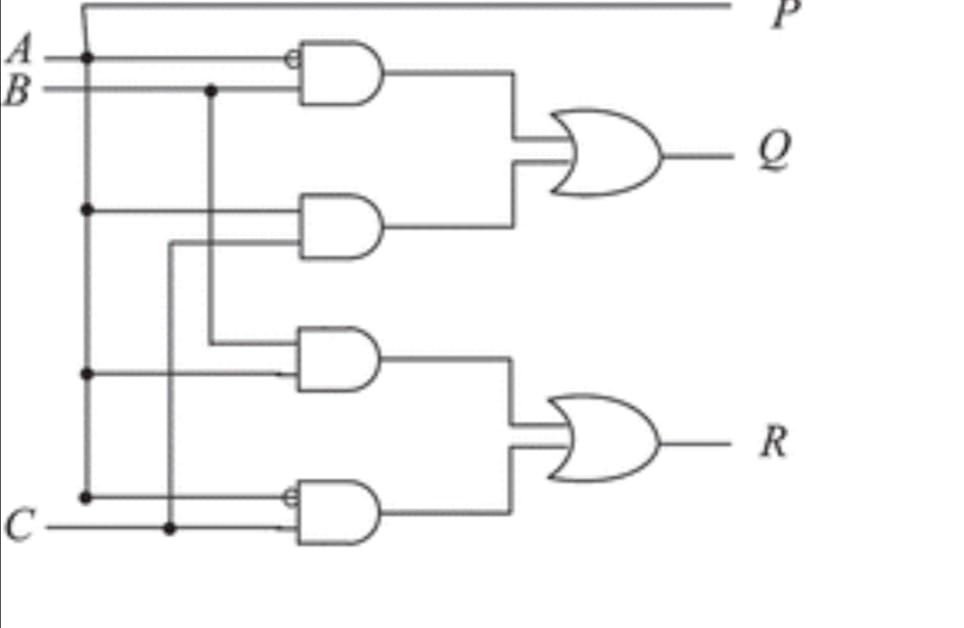
****

FIG 6 : FREDKIN GATE CIRCUIT DIAGRAM

**PERES GATE**

The Peres gate is a 3x3 reversible logic gate, meaning it has three inputs and three outputs with a one-to-one mapping between them. The gate's outputs are defined as P = A, Q = A XOR B, and R = (A AND B) XOR C, where A, B, and C are the inputs.

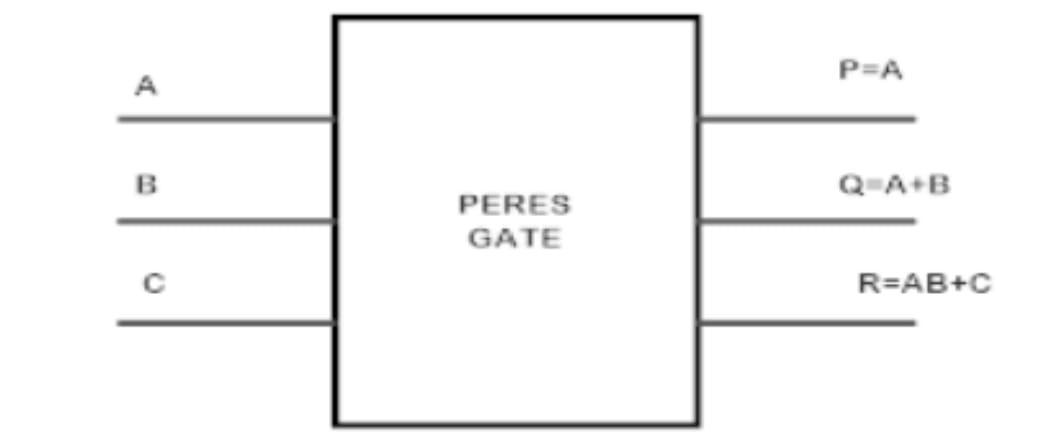


FIG 7 : PERES GATE BLOCK DIAGRAM

In early applications using the Peres gate a symbolic representation was done comprising a Toffoli gate and an additional CNOT gate, meant to cancel the last CNOT of the corresponding Barenco et al. model This representation is also used in Fig. 5. Since this representation seemed to be possibly misleading, because the symbolic Peres gate looked more complex than the Toffoli gate instead of being simpler, some authors started to use another symbol for the Peres gate, including a double circle on the bit that should output the sum of the control signals

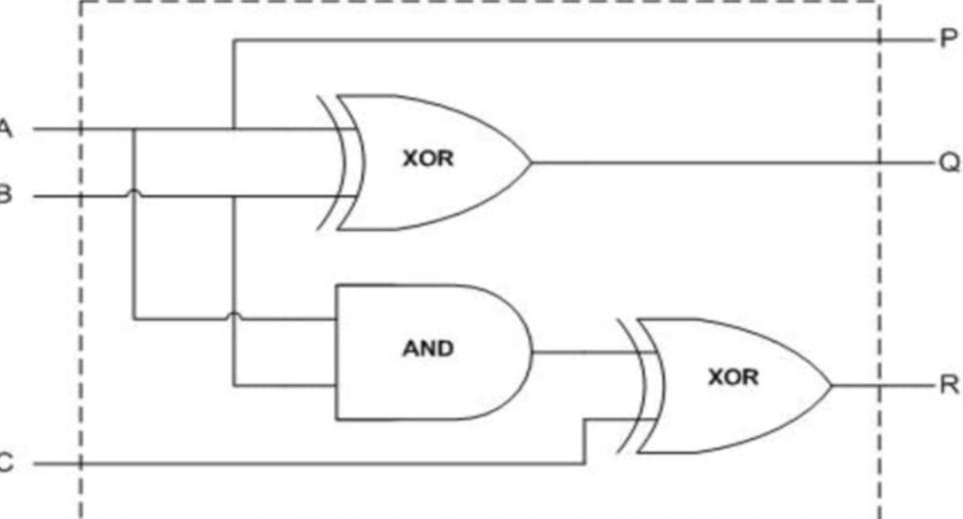


FIG 8 :PERES GATE CIRCUIT DIAGRAM

**C H A P T E R**

4

**EXISTING SYSTEM**

*This chapter outlines the current existence of related systems, tools, and technologies. It highlights the availability of reversible logic components, simulation environments, and testability frameworks that support the development of the proposed design.*

Reversible Logic Gates – such as Peres, Fredkin, and CNOT gates – to minimize energy dissipation, quantum cost, and garbage outputs.

Carry Select Adder (CSA) – an inherently fast arithmetic unit that pre-computes outputs for both possible input carry conditions (0 and 1).

Online Testability – through integration of a Two-Pair Two-Rail Checker circuit to detect single stuck-at faults during circuit operation, ensuring error detection on-the-fly.

**Two designs of reversible CSAs:**

Design 1: Based on direct gate-level mapping of traditional CSA using multiple Peres and Fredkin gates.

Design 2: Based on simplified logic equations (using XOR, XNOR, AND, OR) for sum and carry outputs, reducing quantum cost and ancilla.

Fault Detection Integration: By including a configurable logic block that feeds complementary outputs to the rail checker, you implemented self-checking capabilities in the adder.

**SOFTWARE SPECIFICATION**

**XILINX**

Xilinx ISE[[2]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-xilinx_com-xug-2) (short for *Integrated Synthesis Environment*)[[3]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b1-3) is a discontinued software tool from [Xilinx](https://en.wikipedia.org/wiki/Xilinx) for [synthesis](https://en.wikipedia.org/wiki/Logic_synthesis) and analysis of [HDL](https://en.wikipedia.org/wiki/Hardware_description_language) designs, which primarily targets development of  [embedded](https://en.wikipedia.org/wiki/Embedded_system) [firmware](https://en.wikipedia.org/wiki/Firmware) for Xilinx [FPGA](https://en.wikipedia.org/wiki/FPGA) and [CPLD](https://en.wikipedia.org/wiki/CPLD) integrated circuit (IC) product families. It was succeeded by [Xilinx Vivado](https://en.wikipedia.org/wiki/Xilinx_Vivado). Use of the last released edition from October 2013 continues for [in-system programming](https://en.wikipedia.org/wiki/In-system_programming) of [legacy hardware](https://en.wikipedia.org/wiki/Legacy_system) designs containing older FPGAs and CPLDs otherwise orphaned by the replacement design tool, [Vivado Design Suite](https://en.wikipedia.org/wiki/Xilinx_Vivado" \o "Xilinx Vivado).

ISE enables the developer to synthesize ("compile") their designs, perform [timing analysis](https://en.wikipedia.org/wiki/Static_timing_analysis), examine [Register transfer level](https://en.wikipedia.org/wiki/Register_transfer_level) (RTL) diagrams, simulate a design's reaction to different stimuli, and configure the target device with the [programmer](https://en.wikipedia.org/wiki/Programmer_(hardware)). Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and chip Scope Pro.[[4]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b4-4) The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the [ModelSim](https://en.wikipedia.org/wiki/ModelSim" \o "ModelSim) logic simulator is used for system-level testing.[[5]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b2-5)[[6]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b6-6)

As commonly practiced in the commercial [electronic design automation](https://en.wikipedia.org/wiki/Electronic_design_automation) sector, Xilinx ISE is [tightly-coupled](https://en.wikipedia.org/wiki/Vendor_lock-in) to the architecture of Xilinx's own chips (the internals of which are highly proprietary) and cannot be used with FPGA products from other vendors.[[3]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b1-3) Given the highly proprietary nature of the Xilinx hardware product lines, it is rarely possible to use open source alternatives to tooling provided directly from Xilinx, although as of 2020, some exploratory attempts are being made.

The primary user interface of the ISE is the Project Navigator, which includes the design hierarchy (Sources), a [source code](https://en.wikipedia.org/wiki/Source_code) editor (Workplace), an output console (Transcript), and a processes tree (Processes).

The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a [tree structure](https://en.wikipedia.org/wiki/Tree_structure).[[3]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b1-3) For single-chip designs there may be one main module, with other modules included by the main module, similar to the main() subroutine in [C++](https://en.wikipedia.org/wiki/C%2B%2B) programs.[[3]](https://en.wikipedia.org/wiki/Xilinx_ISE#cite_note-b1-3) Design constraints are specified in modules, which include pin configuration and mapping.

**CADENCE**

Cadence Design Systems is a leading provider of electronic design automation (EDA) software used for designing and verifying integrated circuits (ICs), systems on chip (SoCs), and printed circuit boards (PCBs). It offers a comprehensive suite of tools for both analog and digital circuit design. For this project, Cadence tools were utilized to model, simulate, and analyze the **Carry Select Adder (CSLA)** using **reversible logic gates**, and to implement **online testability** features.

### ****Cadence Tools Used****

* ***Cadence Virtuoso****:* Used for schematic design and transistor-level implementation of reversible logic gates such as Toffoli, Fredkin, and Peres gates. Virtuoso allows precise control of layout and parasitic analysis.
* ***Cadence Spectre****:* Used for analog and digital simulation to test the performance of the designed CSLA. It provides accurate timing, delay, and power analysis.
* ***Cadence Genus Synthesis Solution****:* A digital synthesis tool used to convert RTL-level designs into gate-level representations.
* ***Cadence Innovus****:* Used for physical implementation (place and route) and for analyzing timing, area, and power in complex designs.
* ***Cadence NC-Sim / Xcelium****:* These tools provide functional simulation for RTL designs, which is useful in verifying logical correctness and online testability.

### ****Design Flow Using Cadence****

1. ***RTL Design****:*
   * The CSLA architecture was first modeled using reversible logic gate modules in VHDL/Verilog.
   * These modules were synthesized using **Cadence Genus**.
2. ***Functional Simulation****:*
   * Using **Cadence Xcelium** , the design was simulated for functional correctness.
   * Testbenches were created to verify online testability features (e.g., parity detection).
3. ***Schematic and Layout****:*
   * The reversible gates were designed at the transistor level in **Virtuoso**.
   * Layout vs. Schematic (LVS) and Design Rule Checks (DRC) ensured correctness and manufacturability.
4. ***Timing and Power Analysis****:*
   * **Cadence Spectre** and **Innovus** were used to analyze the timing delays and power consumption of the final design.
   * The reversible design showed reduced power dissipation compared to conventional logic due to minimal information loss.

### ****Benefits of Using Cadence****

* **Accuracy:** Industry-standard tools for precise power and timing analysis.
* **Integration:** Seamless integration across design and verification phases.
* **Advanced Visualization:** Easy debugging and waveform analysis for RTL and layout simulations.
* **Support for Reversible Logic:** Custom gate libraries can be created for reversible logic simulation and layout.

Cadence software played a crucial role in the design, simulation, and verification of the Carry Select Adder with online testability using reversible logic. It enabled efficient implementation of complex gates, accurate timing and power analysis, and verification of testability features. The use of Cadence tools ensured that the design met all the performance, power, and area requirements while exploring the emerging field of reversible computing.

**PROGRAMMING LANGUAGE**

Verilog, [standardized](https://en.wikipedia.org/wiki/Standardized) as IEEE 1364, is a [hardware description language](https://en.wikipedia.org/wiki/Hardware_description_language) (HDL) used to model [electronic systems](https://en.wikipedia.org/wiki/Electronic_system). It is most commonly used in the design and verification of digital circuits, with the highest level of abstraction being at the [register-transfer level](https://en.wikipedia.org/wiki/Register-transfer_level). It is also used in the verification of [analog circuits](https://en.wikipedia.org/wiki/Analogue_electronics) and [mixed-signal circuits](https://en.wikipedia.org/wiki/Mixed-signal_integrated_circuit), as well as in the design of [genetic circuits](https://en.wikipedia.org/wiki/Synthetic_biological_circuit)

In 2009, the Verilog standard (IEEE 1364-2005) was merged into the [SystemVerilog](https://en.wikipedia.org/wiki/SystemVerilog" \o "SystemVerilog) standard, creating IEEE Standard 1800-2009. Since then, Verilog has been officially part of the System Verilog language. The current version is IEEE standard 1800-2023

Hardware description languages such as Verilog are similar to [software](https://en.wikipedia.org/wiki/Software) [programming languages](https://en.wikipedia.org/wiki/Programming_language) because they include ways of describing the propagation time and signal strengths (sensitivity). There are two types of [assignment operators](https://en.wikipedia.org/wiki/Assignment_operator); a blocking assignment (=), and a non-blocking (<=) assignment. The non-blocking assignment allows designers to describe a state-machine update without needing to declare and use [temporary storage variables](https://en.wikipedia.org/wiki/Temporary_storage_variable). Since these concepts are part of Verilog's language semantics, designers could quickly write descriptions.

Now we need to set the work directory and create a work so go to File -> Set Design Directory from the File menu of the Menu Bar. First set the design directory to the tutorial directory that you just created

To compile you first need to select the various files in the File Browser (left side) by clicking on the mouse left button (simultaneously press on shift for multiple selections), then click on the corresponding buttons in the Menu Bar or explicitly go to Tools -> Verilog Compiler.... Now is actually a good time to look at these files as well, you can do that for example by clicking on the file, then going to File -> Edit, you should see the actual file in a text editor:

Finally, we can simulate! Click on the + sign in front of the Snapshots library to expand its contents, then select worklib. stimulus: module and click on or go to Tools -> Simulator.... This should launch the Simvision Design Browser and Console windows: Click on stimulus on the left window, then on the Waveform button (the one that looks like a set of white digital waveforms on a black background, sixth from the right), this should open a new Waveform window. Now press on the Run button on the Design Browser window

**C H A P T E R**

5

**PROPOSED SYSTEM**

*This chapter describes the proposed design of a carry select adder with online testability using reversible gates. It explains the architecture, working principles, and integration of reversible logic to achieve low power consumption, high speed, and fault detection during operation.*

CSA is one of the quickest adders for arithmetic functions. Because this adder pre calculates the sum by considering two possibilities with carry in as '0' and carry in as '1'. So, this adder just to select the required sum from available sums.

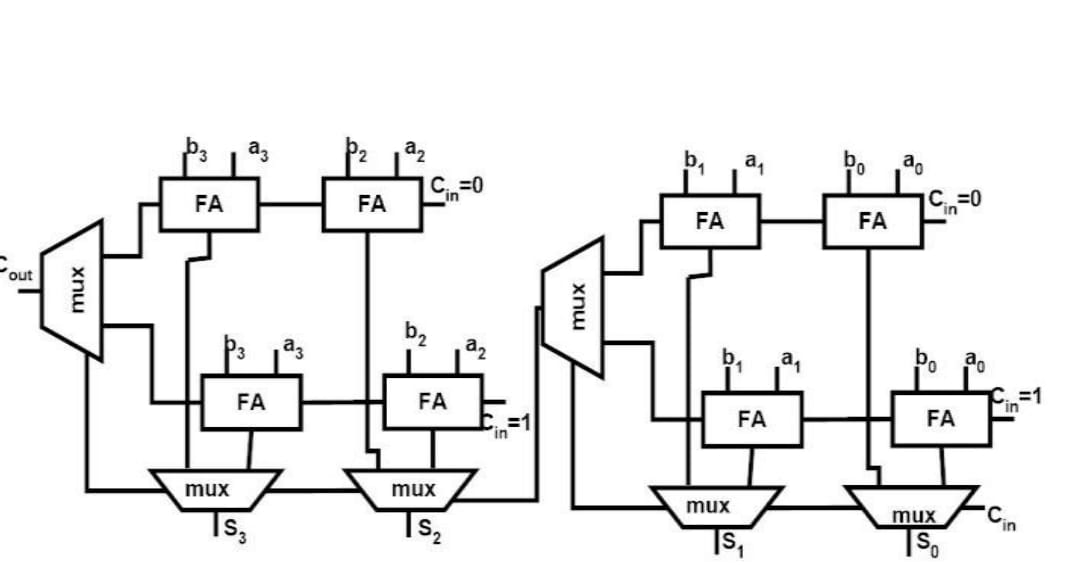


FIG 9: CSA DESIGN 1

A. CSA-design 1:

Reversible CSA design 1 is implemented directly from conventional method. For implementing 1-bit adder this design requires four PERES gates and two FREDKIN gates. 4-bit carry select adder which is build using reversible gates can be seen in Fig. 7. Where, RFA is Reversible full adder which can be seen in Fig. 5.

At each stage two RFA's are used, one is for whenever carry in is '0' and another for carry in is '1'. FG (FREDKIN) gates in the circuits are used as multiplexers, CNOT gates at bottom used as buffers. The number of garbage outputs for each stage are eight. This design is having a Quantum cost of twenty six for each stage, as it requires four Peres gates which costs four for each and two Fredkin gates which costs five for each.

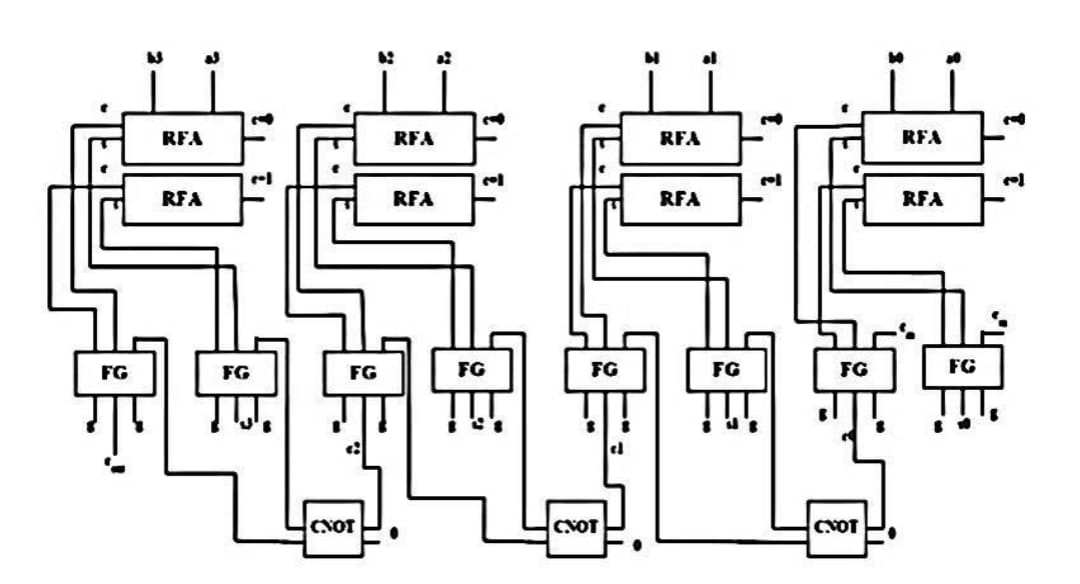


FIG 10: CSA DESIGN 2

Reversible carry select adder- design 2 is implemented from the logic which is given in Table 1. This design efficient and faster as it need to perform logical operations two bits only [15]. This circuit is designed using Verilog and implemented in Xilinx ISE 14.1 [16] and simulation results are shown in Fig. 9. This design requires two CNOT gates, two PERES gates and two FREDKIN gates for single stage.

2-bit CSA which is built from reversible gates using design-2 is shown in Fig. 8. Where CNOT gates at top is used as EXOR and EX-NOR gates (from right side) for calculating sum whenever carry in is '0' and '1'. PERES gates are used as AND and OR gates (from right side) for calculating carry out when carry in is '0' and '1'. The CNOT gate at bottom is used as a buffer. The total number of garbage outputs for each stage are ten. This design is having a Quantum cost of twenty for each stage, as it requires two PERES gates which costs four for each, two CNOT gates which costs one for each and two FREDKIN gates which costs five for each.

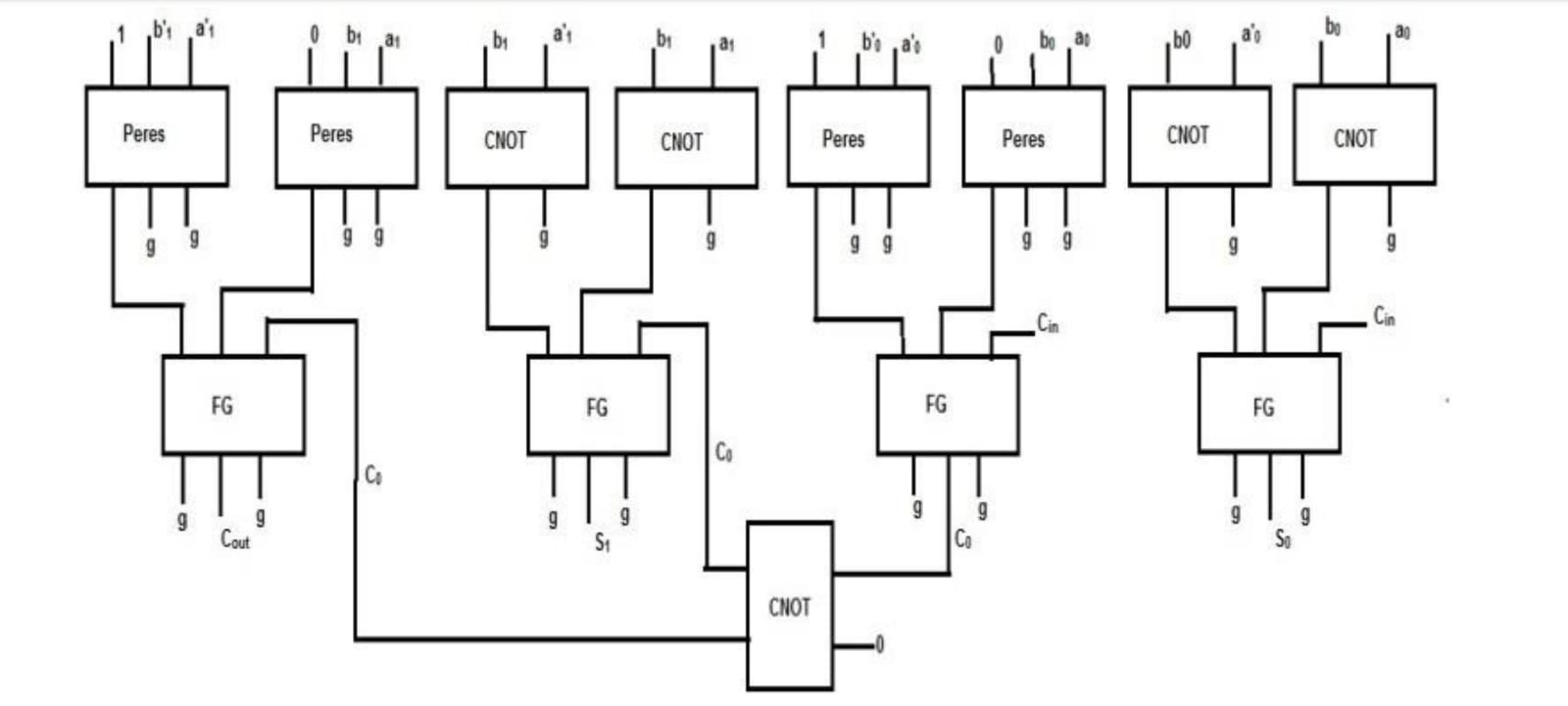


FIG 11: DESIGN 3

C. Carry select adder with online testability:

Online testability means adding extra circuit to existing circuit to detect faults in the circuits [17]. Reversible circuits testing is in existence since last few years and the scientists are demonstrating their interest in testing of reversible logic.

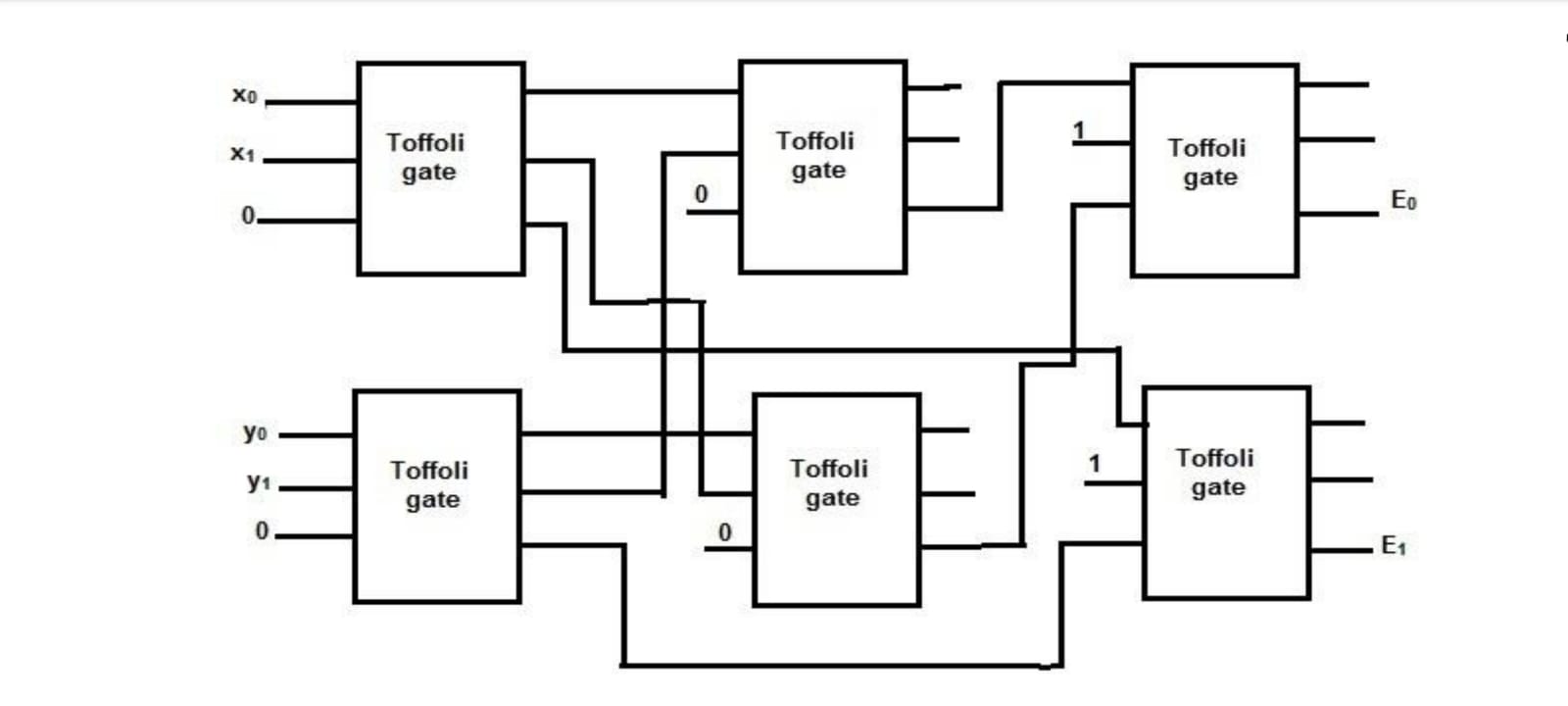


FIG 12: DESIGN 4

To design online testability to carry select adder we have to design a configurable block such that the inputs to the two pair rail checker are complementary pairs. Assume S\_{0} S\_{1} are full-adder sum outputs whenever carry in '0' with inputs as (a\_{0}, b\_{0}) & (a\_{1}, b\_{1}) S 10, S\_{11} are full-adder outputs whenever carry-in is '1' with inputs as (a\_{0}, b\_{0}) & (a\_{1}, b\_{1}) So, we need to design a configurable block for carry-select adder such that inputs to the rail checker circuit are complementary pairs

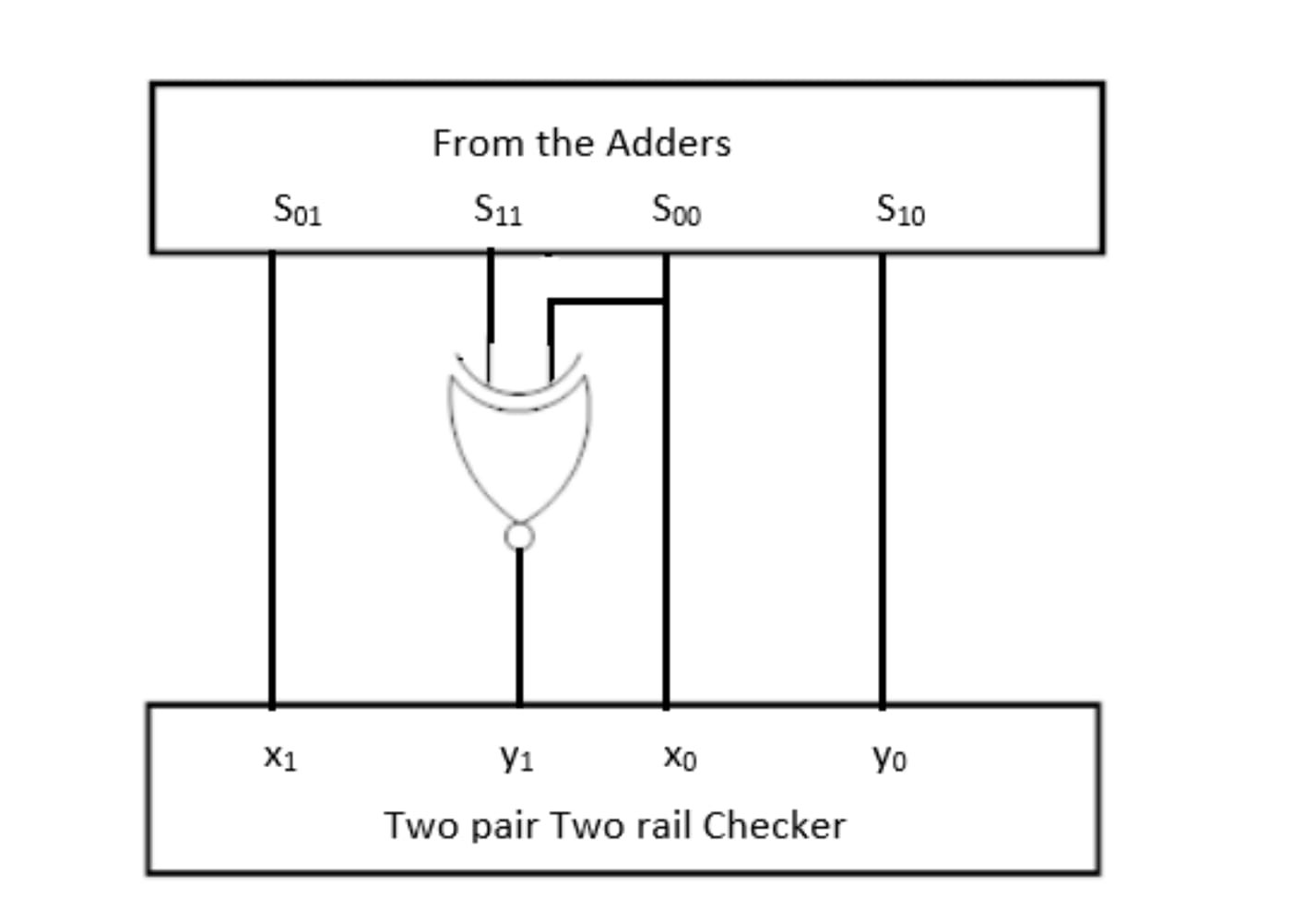


FIG 13: DESIGN 5

Consider inputs of 2-bit adder are (a\_{0}, b\_{0}) = (0, 1) and (a\_{1}, b\_{1}) = (1, 1) S\_{0} S\_{1} , S\_{10} and S\_{11} become 1,0,0 and 1. Assume S\_{11} is stuck-at 0 then S\_{11} becomes 0. So, inputs to the rail checker circuits become (x\_{1}, y\_{1}, x\_{0}, y\_{0}) = (0, 0, 1, 0) (x\_{1}, y\_{1}) pair is not complemented. This condition results both outputs of rail checker circuit will become 0 and 0 which is not a valid out code. So, fault is detected in the circuit.

**C H A P T E R**

6

**SOURCE CODE**

*This chapter presents the Verilog source code developed for simulating and testing the proposed carry select adder with online testability using reversible logic gates. The code includes the main design modules, testbench, and necessary logic for fault detection and waveform generation.*

module Peres(input A, B, C, output P, Q, R);

wire t1;

assign P = A;

assign t1 = A ^ B;

assign Q = t1;

assign R = (A & B) ^ C;

endmodule

module FG(input A, B, C, output P, Q);

assign P = A;

assign Q = A ? C : B;

endmodule

module CNOT(input A, B, output Y);

assign Y = A ^ B;

endmodule

module carry\_select\_online\_testable(

input a1, a0, b1, b0, a1\_not, a0\_not, b1\_not, b0\_not, Cin,

output Cout, x0, y0, x1, y1, E0, E1);

wire w1, w2, w3, w4, w5, w6, w7, w8;

wire s0, s1, s00, s01, s10, s11;

wire p1\_1, p1\_2, p1\_3;

wire p2\_1, p2\_2, p2\_3;

Peres per1(1'b1, b0\_not, a0\_not, p1\_1, p1\_2, p1\_3);

Peres per2(1'b0, b0, a0, p2\_1, p2\_2, p2\_3);

FG fg0(p1\_3, p2\_3, Cin, s0, w1); // C0

wire p3\_1, p3\_2, p3\_3;

wire p4\_1, p4\_2, p4\_3;

Peres per3(1'b1, b1\_not, a1\_not, p3\_1, p3\_2, p3\_3);

Peres per4(1'b0, b1, a1, p4\_1, p4\_2, p4\_3);

FG fg1(p3\_3, p4\_3, w1, s1, Cout);

CNOT c1(b0, a0, w2);

CNOT c2(a0\_not, b0\_not, w3);

CNOT c3(w2, 1'b0, s00);

CNOT c4(w3, 1'b0, s01);

CNOT c5(b1, a1, w4);

CNOT c6(a1\_not, b1\_not, w5);

CNOT c7(w4, 1'b0, s10);

CNOT c8(w5, 1'b0, s11);

CNOT c9(s00, s00, x0);

CNOT c10(s00, s0, y0);

CNOT c11(s01, s01, x1);

CNOT c12(s01, s1, y1);

assign E0 = x0 ^ y0;

assign E1 = x1 ^ y1;

endmodule

module testbench;

reg a0, a1, b0, b1, Cin;

wire a0\_not, a1\_not, b0\_not, b1\_not;

wire Cout, x0, y0, x1, y1, E0, E1;

assign a0\_not = ~a0;

assign a1\_not = ~a1;

assign b0\_not = ~b0;

assign b1\_not = ~b1;

carry\_select\_online\_testable uut(

a1, a0, b1, b0, a1\_not, a0\_not, b1\_not, b0\_not, Cin,

Cout, x0, y0, x1, y1, E0, E1 );

initial begin

$dumpfile("test.vcd");

$dumpvars(0, testbench);

$display("Time | a1 a0 b1 b0 Cin | Cout x0 y0 x1 y1 E0 E1");

for (integer i = 0; i < 32; i = i + 1) begin

{a1, a0, b1, b0, Cin} = i; #10;

$display("%4t | %b %b %b %b %b | %b %b %b %b %b %b %b", $time, a1, a0, b1, b0, Cin, Cout, x0, y0, x1, y1, E0, E1);

end

$finish;

end

endmodule

**CIRCUIT DIAGRAM**

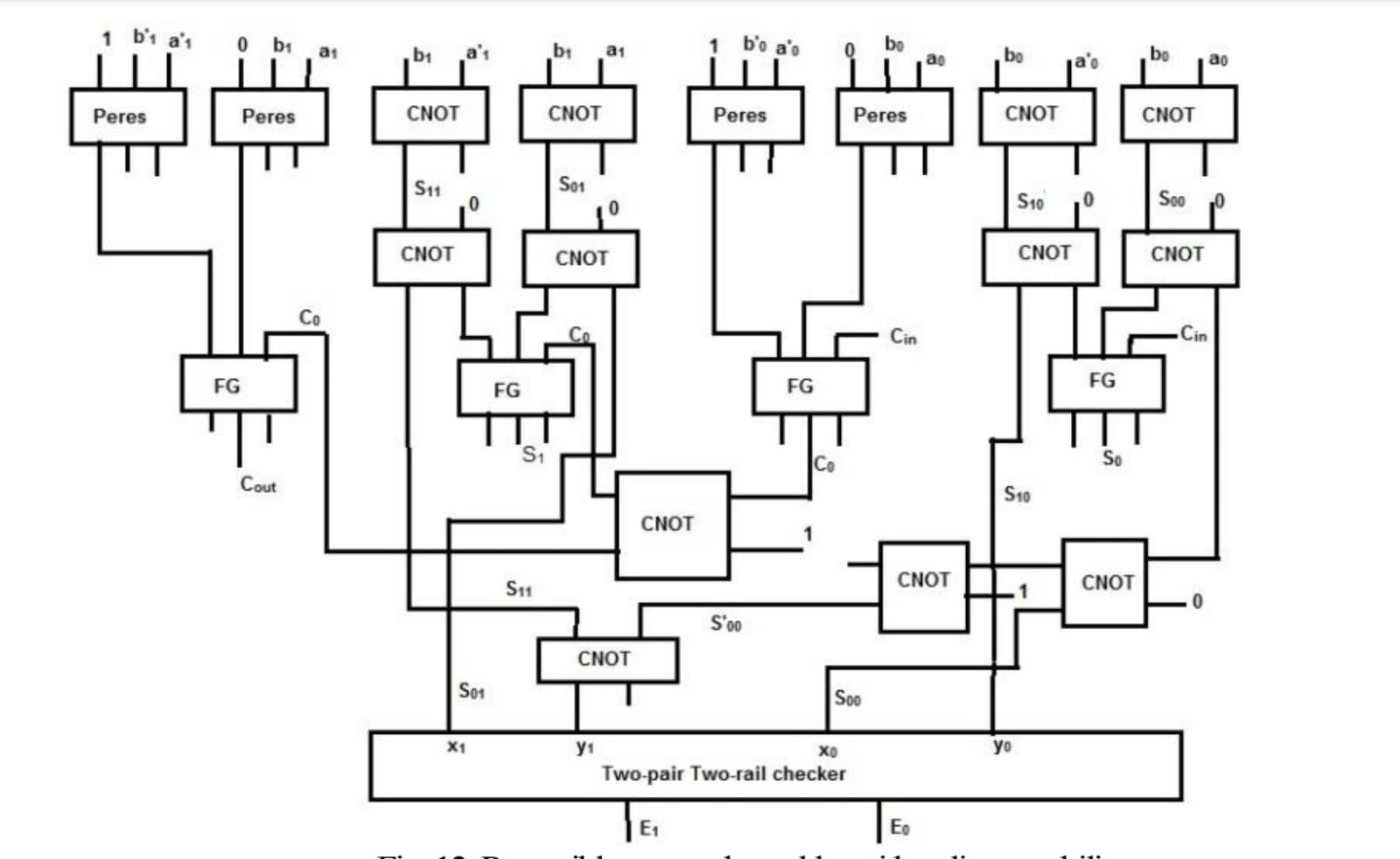
****

FIG 14: RCSA DESIGN 2

**C H A P T E R**

7

**RESULTS**

*This chapter presents the gate-level implementation of the proposed design, showing how the circuit is built, simulated, and tested using Verilog and related tools.*

VLSI (Very Large Scale Integration) refers to the process of creating integrated circuits by combining millions or billions of transistors onto a single chip. The design of such chips can be broadly divided into two main phases: **Logical Design** and **Physical Design**. While logical design defines the function of the chip (through logic gates and circuits), **Physical Design** translates this functionality into geometric representations (layouts) that can be manufactured using photolithography.

**Physical Design** is the process of transforming a logical description of a circuit (usually in the form of a netlist) into a physical layout. This layout must satisfy a number of constraints, including area, timing, power, and manufacturability. The end goal is to generate GDSII (Graphic Data System) or OASIS files, which are used in the fabrication of chips.

**Main Steps of Physical Design**

**1. Partitioning**

Partitioning involves dividing the chip into smaller, manageable blocks or modules. This helps reduce the complexity of subsequent steps and allows for parallel processing during layout. The goal is to minimize the number of interconnections between partitions (cut-size) while balancing their sizes.

**2. Floorplanning**

In this stage, each partition or module is assigned a specific area on the chip. The arrangement must consider aspect ratios, I/O pin placements, power distribution, and inter-module communication. Floorplanning affects performance, power consumption, and routing complexity.

**3. Placement**

Placement assigns specific locations to individual standard cells (logic gates). Good placement minimizes wirelength, delays, and power consumption while maximizing routability. Modern tools use algorithms like simulated annealing, force-directed placement, or analytical placement.

**4. Clock Tree Synthesis (CTS)**

The clock signal must reach all sequential elements (like flip-flops) simultaneously. CTS builds a balanced tree structure to distribute the clock with minimal skew and low power. It is a crucial step for ensuring timing accuracy.

**5. Routing**

Routing connects all the placed components (cells and blocks) using metal wires. It’s done in two stages:

* **Global Routing**: Determines approximate paths for each net.
* **Detailed Routing**: Assigns exact tracks and layers to complete connections.

Routing must avoid violations like shorts and spacing errors and adhere to design rules.

**6. Physical Verification**

This stage ensures that the layout is manufacturable and meets all design rules. It includes:

* **DRC (Design Rule Checking)**: Verifies compliance with manufacturing constraints.
* **LVS (Layout Versus Schematic)**: Checks if the layout matches the original schematic/netlist.
* **ERC (Electrical Rule Check)**: Ensures proper electrical functioning (e.g., no floating gates).

**7. Timing Analysis and Optimization**

Static Timing Analysis (STA) checks if all paths in the design meet their timing constraints. Designers use techniques like buffer insertion, gate sizing, and logic restructuring to optimize timing.-

**CIRCUIT DIAGRAM :**

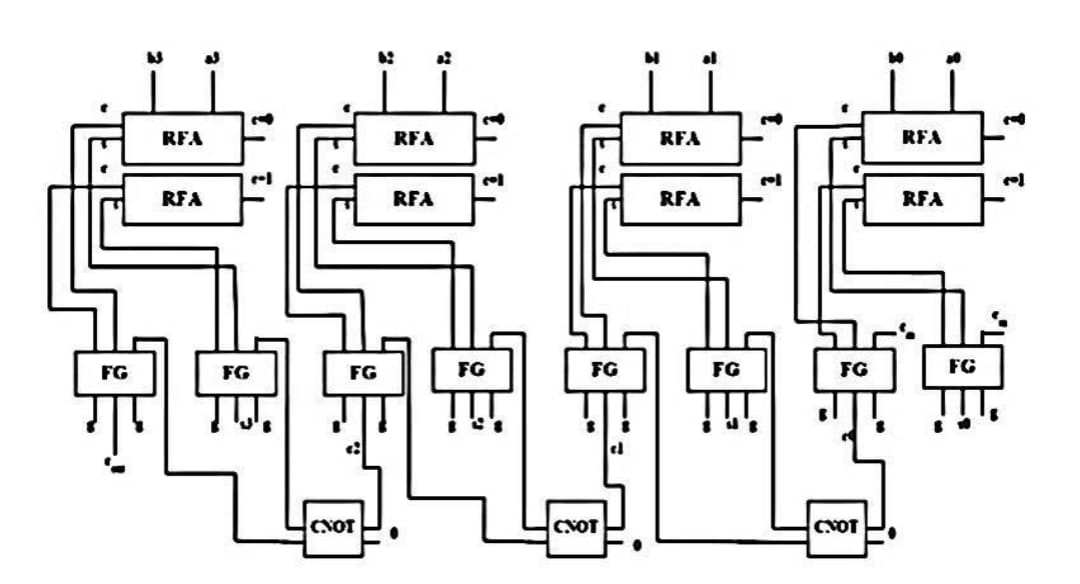


FIG 15 : CIRCUIT DIAGRAM OF RCSA DESIGN -1

**PHYSICAL DESIGN:**

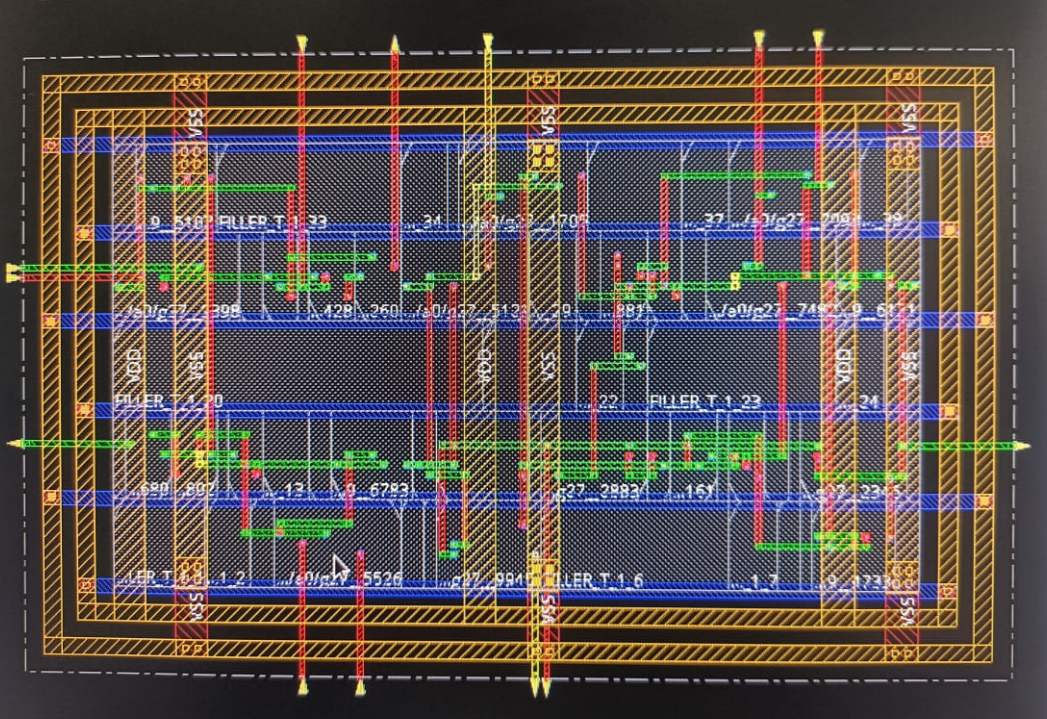
****

FIG 16: PHYSICAL DESIGN OF RCSA DESIGN -1

**CIRCUIT DIAGRAM:**

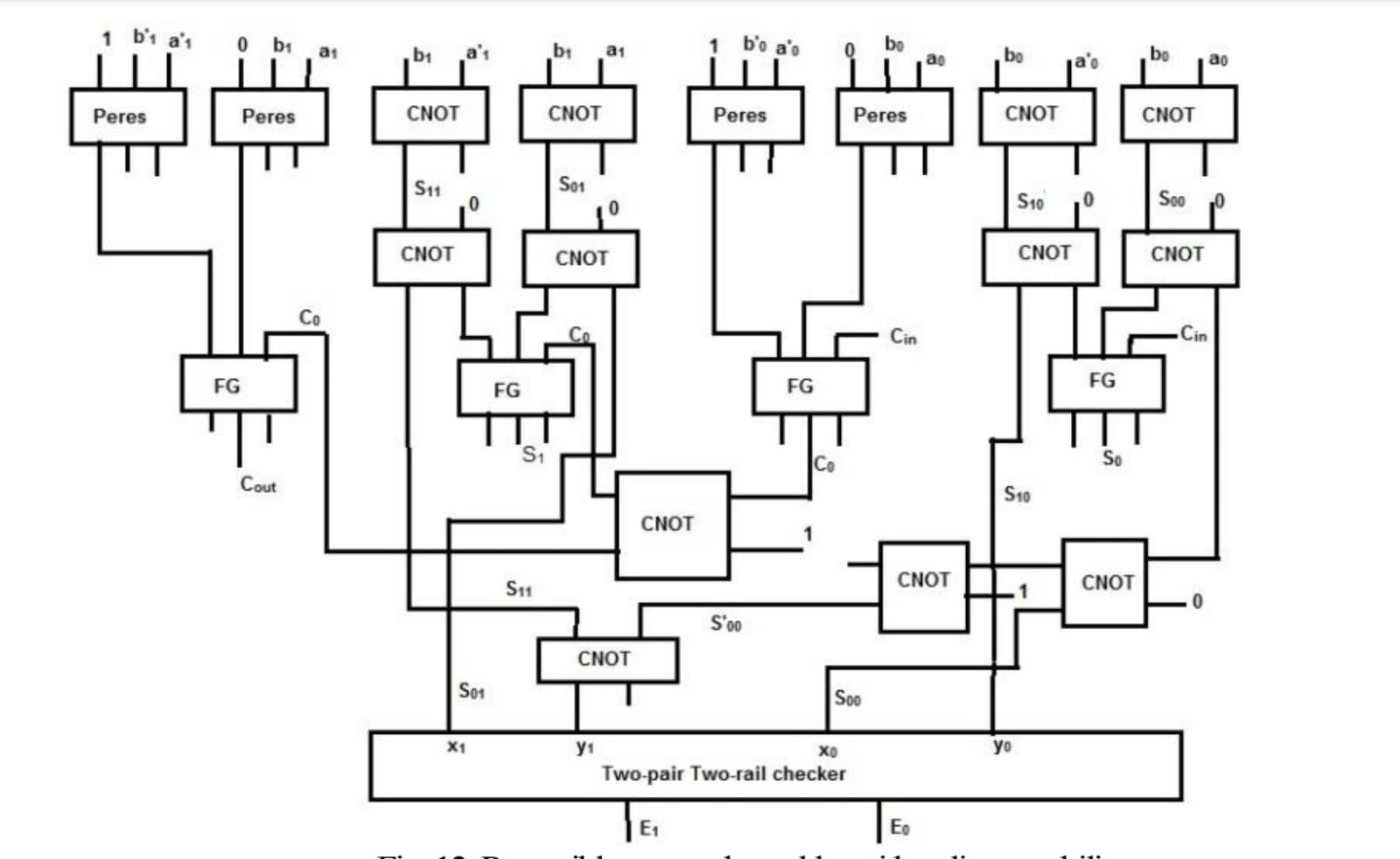
****

FIG 17:CIRCUIT DIAGRAM OF RCSA DESIGN -2

**PHYSICAL DESIGN:**

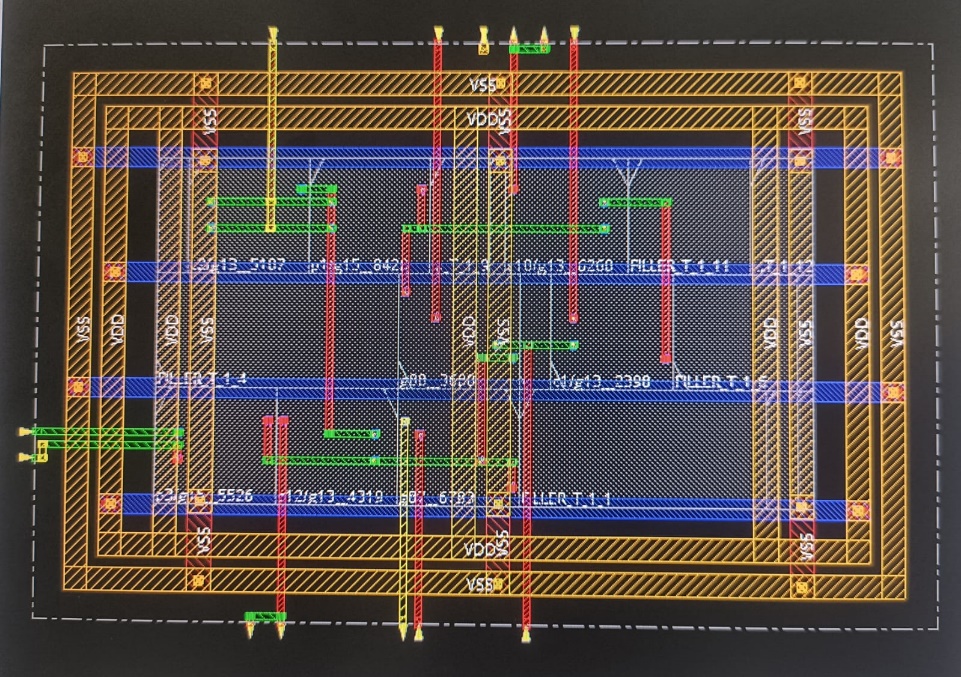
****

FIG 18: PHYSICAL DESIGN OF RCSA DESIGN -2

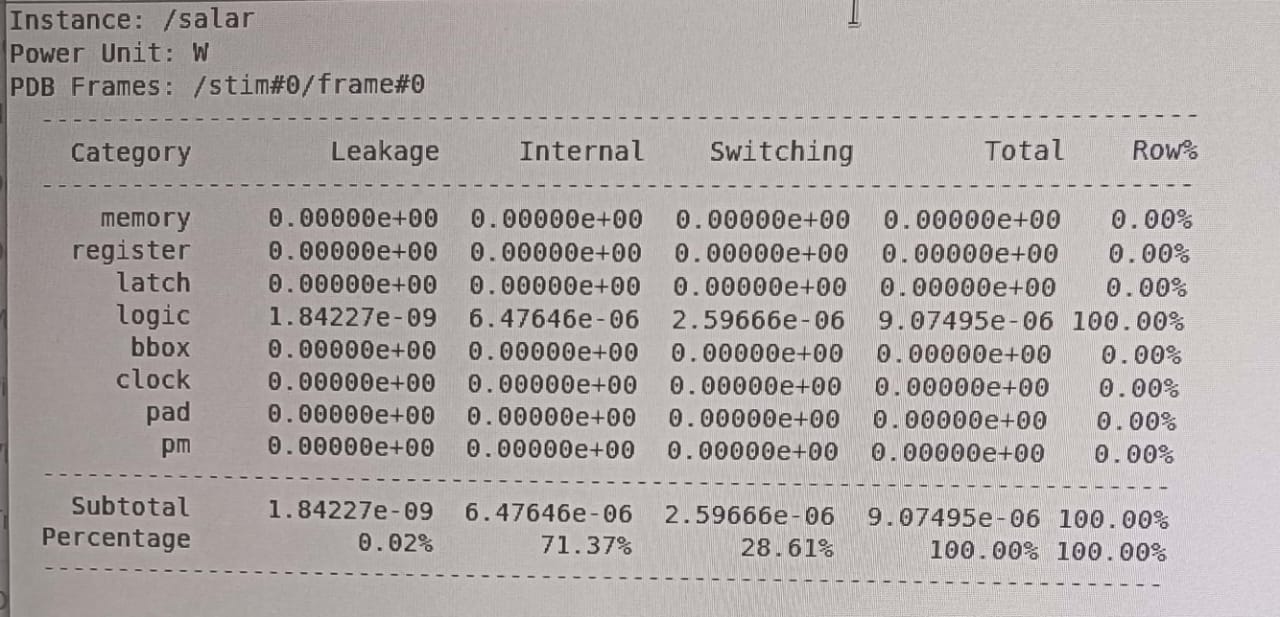


FIG 19: OUTPUT POWER

**COMPARISON TABLE:**

|  |  |  |
| --- | --- | --- |
| **PARAMETER** | **DESIGN 2** | **DESIGN 1** |
| Power | 9.07495e.06 | 15.0893e-09 |
| Area | 71.37%(6.4764e-06) | 93.43% |
| Delay | 28.61%(2.59666e-06) | 51.65% |
| leakage | 0.02%(1.84227e-09) | 0.3% |

**APPLICATIONS:**

* Biomedical devices
* Cryptography
* Fault tolerance
* Quantum computing

**LIMITATIONS**

* Increased Circuit Complexity
* Hardware Overhead
* Timing and Synchronization Issues
* Garbage Outputs and Constant Inputs
* Limited Tool Support
* Scalability Issues
* Design Trade-offs

**C H A P T E R**

8

**CONCLUSION**

*This chapter highlights the project's outcomes and impact, while the future scope suggests possible enhancements and extended applications*

In this project, a novel technique for implementing a reversible carry select adder (CSA) was proposed and evaluated through two distinct designs using reversible logic gates. The use of reversible logic significantly reduces power dissipation and makes the design suitable for future technologies like quantum computing and low-power VLSI systems.

Between the two architectures developed, **Design-2** demonstrated superior efficiency over **Design-1**, particularly in terms of **Ancilla input optimization** and **quantum cost reduction**—two key metrics in reversible logic design. Design-2 not only uses fewer resources but also simplifies the logical structure while preserving performance.

Furthermore, to ensure online testability, a **self-checking mechanism** was integrated into Design-2 using a **two-pair two-rail checker circuit**. This circuit effectively detects **single stuck-at faults** by monitoring the outputs of EX-NOR gates connected to the sum outputs of full adders. In the presence of a fault, the circuit generates invalid input combinations that trigger incorrect output codes, allowing for real-time fault detection without interrupting normal operations.

The simulation and analysis clearly show that Design-2 not only optimizes hardware parameters but also improves fault coverage and reliability—making it a more practical solution for fault-tolerant arithmetic circuits.

In conclusion, the comparison of Design-1 and Design-2 across various parameters (gate count, ancilla bits, garbage outputs, and fault detection capability) confirms that **Design-2 is the better candidate for reversible and fault-tolerant carry-select adder implementation**, laying the groundwork for more energy-efficient and reliable digital systems.

**C H A P T E R**

9

**SELF EVALUATION**

*This chapter reflects on the overall project experience, including the knowledge gained, challenges faced, skills developed, and the effectiveness of the proposed design in meeting the project objectives*

In this project, our achievements were both technical and personal. Undertaking this project during our 5th semester provided us with a valuable opportunity to apply theoretical knowledge to practical implementation. Although the project focused on designing a reversible carry select adder with online testability, it also enhanced our understanding of digital logic design, fault detection, and the fundamentals of hardware simulation.

Throughout the project, we encountered several challenges—ranging from logic errors to simulation mismatches. To overcome them, we collaborated as a team, consulted various reference books, discussed concepts with faculty members, explored multiple online resources, and performed several experiments and observations. This problem-solving process taught us the importance of teamwork, persistence, and continuous learning.

One key realization was that while we may eventually forget some technical details, the hands-on experience and lessons learned from building and troubleshooting our own design will stay with us. This project has not only deepened our knowledge of digital electronics and Verilog coding but also improved our confidence in handling complex design tasks independently.

In summary, this project was an enriching experience that helped shape both our technical skills and our ability to work collaboratively under real-world constraints.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | **Project Details** | | |
| **Academic Year** | | 2024-25 | | |
| **Title of the Project** | | Design of carry select adder with online testability using reversible gates | | |
| **Name of the Students and Hall Ticket No.** | | K. Akhila (23RA5A0402)  B. Charan kumar (23RA5A0404)  K . Trividha (23RA5A0407) | | |
| **Name of the Guide(s)** | | Mrs. Y. Vishwa sri | | |
|  | | **Project PO Mapping** | | |
| **Name of**  **Course from which**  **Principles are applied in this project** | **Related**  **Course**  **Outcome**  **Number** | **Description of the application** | **Page Number** | **Attained PO** |
| DLD(C31) | C31.2 | Introduction | 1-3 | PO1, PO3, PO4, PO11, PO12 |
|  |  | Literature survey | 4-7 | PO1, PO2, PO3, PO4, PO6, PO9, PO11, PO12 |
| DLDC31) | C31.2 | Design considerations | 8-11 | PO1, PO3, PO4, PO5, PO6, PO9, PO11, PO12 |
|  |  | Hardware description | 12-59 | PO1, PO3, PO5, PO6, PO9, PO11, PO12, PO12 |
| VLSI(C31) | C31.4 | Software simulation | 60-64 | PO1, PO3, PO5, PO6, PO9, PO11, PO12, PSO2 |
| VLSI(C31) | C31.3 | Implementation | 65-72 | PO1, PO3, PO4, PO5, PO6, PO9, PO11, PO12 |
|  |  | Result | 73-74 | PO1, PO2, PO3, PO5, PO6, PO9, PO11, PSO1, PSO2, PSO3 |

**Signature of the Students:**

K. Akhila (23RA5A0402)

B. Charan kumar (23RA5A0404)

K . Trividha (23RA5A0407) **Signature of the Internal Guide**

**REFERENCES**

1. R. Landauer, *"Irreversibility and Heat Generation in the Computational Process"*, IBM Journal of Research and Development, Vol. 5, Issue 3, 1961, pp. 183–191, IBM.
2. C. H. Bennett, *"Logical Reversibility of Computation"*, IBM Journal of Research and Development, Vol. 17, Issue 6, 1973, pp. 525–532, IBM.
3. H. Thapliyal and N. Ranganathan, *"Design of Efficient Reversible Binary Subtractors Based on a New Reversible Gate"*, Proceedings of the IEEE Computer Society Annual Symposium on VLSI, 2009, pp. 229–234, IEEE.
4. H. G. Rangaraju, U. Venugopal, K. N. Muralidhara, and K. B. Raja, *"Low Power Reversible Parallel Binary Adder/Subtractor"*, International Journal of VLSI Design & Communication Systems (VLSICS), Vol. 1, No. 3, 2010, pp. 23–34. [arXiv].
5. R. Garipelly, P. M. Kiran, and A. S. Kumar, *"A Review on Reversible Logic Gates and Their Implementation"*, International Journal of Emerging Technology and Advanced Engineering (IJETAE), Vol. 3, No. 3, 2013, pp. 417–423.
6. V. Shukla, O. P. Singh and G. R. Mishra, "A novel approach for reversible realization of 3:8 decoder circuit with optimized performance parameters," 2018 IEEE International Conference on Computational and Characterization Techniques in Engineering & Sciences (CCTES), 2018, pp. 286-290, IEEE.
7. Gaur H.M., Singh A.K., Ghanekar U, “In-depth comparative analysis of reversible gates for designing logic circuits”, Procedia Comput. Sci., vol. 125, 2018, pp.810–817, Elsevier.
8. P. Khatter, N. Pandey and K. Gupta, "An Arithmetic and Logical Unit using Reversible Gates," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 2018, pp. 476-480, IEEE.
9. H. Thapliyal and N. Ranganathan, “Design of reversible sequential circuits optimizing quantum cost, delay and garbage outputs,” ACM Journal of Emerging Technologies in Computing Systems, vol. 6, no. 4, 2010, pp. 14:1–14:35
10. R. P. Feynman, “Quantum mechanical computers,” Optics News, vol. 11, pp. 11–20, 1985.
11. T. Toffoli, “Reversible computing,” Tech. Memo MIT/LCS/TM-151, MIT Lab. for Com. Sci. (1980).