VISVESVARAYA TECHNOLOGICAL UNIVERSITY,

JNANASANGAMA, BELAGAVI- 590018

BLDEA's V.P. Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPUR

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

A Major Project Phase-II report on

"DESIGN OF AN AREA EFFICIENT REVERSIBLE FIR FILTER"

Submitted in partial fulfillment for the award of degree of Bachelor of Engineering in Electronics and Communication Engineering

Submitted by

RANI ANIL PATIL	**2BL22EC074**
SARASWATI KANGAL	**2BL22EC090**
SHREEDEVI G KUMBAR	**2BL22EC096**
SWATI ROOGI	**2BL22EC113**

Under the Guidance of

Dr. DANESHWARI HATTI

2025-26

VISVESVARAYA TECHNOLOGICAL UNIVERSITY,

BELAGAVI

B.L.D.E. Association's

V.P Dr. P.G HALAKATTI COLLEGE OF ENGINEERING AND TECHNOLOGY, VIJAYAPUR

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is Certified that the Major project phase-II work entitled **"DESIGN OF AN AREA EFFICIENT REVERSIBLE FIR FILTER"** carried out by **Rani Anil Patil, Saraswati Kangal, Shreedevi G Kumbar and Swati Roogi** bonafide students of **VP Dr P.G Halakatti College of Engineering and Technology, Vijayapura** in partial fulfillment for the award of **Bachelor of Engineering in Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belgaum** during the year 2025-2026. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The Major project Phase-II report has been approved as it satisfies the academic requirement in respect of Major project phase-II work prescribed for the said degree.

GUIDE	H.O.D	PRINCIPAL
Dr. DANESHWARI HATTI	Dr. J S GONAL	Dr. MANJUNATH P

EXTERNAL VIVA

Name of the Examiner	Signature with Date
1.	
2.	

DECLARATION

We, students of Seventh semester B.E, at the department of Electronics and Communication Engineering, hereby declare that, the Major Project Phase-II entitled **"DESIGN OF AN AREA EFFICIENT REVERSIBLE FIR FILTER"** embodies the report of our Major Project Phase-II work, carried out by us under the guidance of **Dr. DANESHWARI HATTI ,** We also declare that, to the best of our knowledge and belief, the work reported here does not form part of any other report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any student.

Place:- Vijayapura Date:-28/11/2025

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success. We consider it as our privilege to express the gratitude to all those who guided in the completion of our Major Project Phase-II.

First and foremost, We wish to express our profound gratitude to our respected Principal **Dr. MANJUNATH P, B.L.D.E. Association's VACHANA PITAMAHA Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, Vijayapura**, for providing us with a congenial environment to work in.

We would like to express our sincere thanks to **Dr. J.S. GONAL**, the HOD of **Electronics and Communication Engineering**, **B.L.D.E. Association's VACHANA PITAMAHA Dr. P. G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, Vijayapura**, for her continuous support and encouragement.

We are greatly indebted to our guide **Dr. DANESHWARI HATTI**, Department of **Electronics and Communication Engineering**, **B.L.D.E. Association's VACHANA PITAMAHA Dr. P.G. HALAKATTI COLLEGE OF ENGINEERING & TECHNOLOGY, Vijayapura**, who took great interest in our work. She motivated us and guided us throughout the accomplishment of this goal. We express our profound thanks for her meticulous guidance.

RANI ANIL PATIL	**2BL22EC074**
SARASWATI KANGAL	**2BL22EC090**
SHREEDEVI G KUMBAR	**2BL22EC096**
SWATI ROOGI	**2BL22EC113**

ABSTRACT

Reversible computing is a model of computation where every process is reversible, ensuring a one-to-one mapping between input and output states. This property minimizes energy loss and enables the design of low-power digital circuits. In this work, a reversible FIR (Finite Impulse Response) filter architecture is designed and implemented entirely using Cadence digital design tools. The design employs Fredkin, Peres, and PFAG reversible logic gates to realize efficient arithmetic units such as the Vedic Multiplier and Lookahead Carry Select Adder (LCSA). The complete design flow—including schematic design, simulation, synthesis, and layout—was carried out using Cadence Genus 14.25 and Cadence Innovus. Multiple reversible FIR architectures were analyzed and compared based on power, delay, and silicon area. The results show that the architecture designed using the Vedic Multiplier and LCSA with PFAG gates achieves significant reductions in area and cell count, offering superior performance compared to conventional FIR filter designs. This work demonstrates that reversible FIR filter implementation using Cadence tools provides an effective solution for low-power, high performance VLSI design.

Keywords: FIR Filter, Reversible Logic, Vedic Multiplier, LCSA, Fredkin Gate, Peres Gate, PFAG Gate, Cadence

CONTENTS

LIST OF FIGURE

LIST OF TABLE

INTRODUCTION

CHAPTER 1

INTRODUCTION

Finite Impulse Response (FIR) filters are a fundamental component in digital signal processing (DSP), widely used in applications such as audio processing, communication systems, and image filtering due to their inherent stability and linear phase characteristics. With the growing demand for energy-efficient and high-performance digital systems, the concept of reversible computing has gained significant attention. Reversible logic, which allows computations to be performed without information loss, offers promising advantages in reducing power dissipation — a critical consideration in modern VLSI and quantum computing environments.

In this project, we present the design and implementation of a Reversible FIR Filter using reversible logic gates. Unlike conventional designs, reversible circuits do not lose information and theoretically dissipate zero energy, following Landauer's principle. By integrating reversible computing principles into FIR filter architecture, we aim to achieve a low-power, area-efficient digital filter design suitable for next-generation computing platforms, including quantum and nanotechnology-based systems.

This work focuses on synthesizing a FIR filter using commonly known reversible logic gates such as the Peres Gate, Feynman gate, Toffoli gate, and Fredkin gate. We evaluate the design in terms of key metrics such as quantum cost, garbage outputs, delay, and hardware complexity, comparing it with traditional irreversible FIR filter implementations.

This research approach are aligned with prior studies published in **Journal of Advanced Research in Dynamical & Control Systems, Vol. 10, 13-Special Issue, 2018**, which emphasize the importance and feasibility of reversible logic in achieving efficient digital signal processing architectures.

LITERATURE REVIEW

CHAPTER 2

LITERATURE REVIEW

The design of low-power and area-efficient digital circuits has gained significant momentum over the past decades, driven by the increasing demand for energy-efficient and compact devices in portable and embedded systems. The application of reversible logic in VLSI design has emerged as a promising approach to overcome the limitations of conventional logic in terms of power dissipation and heat generation. Prior studies, such as those published in the International Journal of VLSI System Design and Communication Systems Volume.04, IssueNo.11, October-2016, Pages: 1192-1198, highlight the relevance and effectiveness of reversible logic as a foundation for developing energy-aware computational models.

R. Landauer (1961) was the first to highlight the theoretical relationship between information loss and energy dissipation in digital systems. He showed that each bit of information lost during computation results in an energy dissipation of at least kTln2, where k is Boltzmann's constant and T is the temperature. Later, **C.H. Bennett (1973)** proved that computation can be made thermodynamically reversible, thus avoiding this energy loss if the process is logically reversible.

In the context of **Digital Signal Processing (DSP)**, FIR filters are crucial for applications such as audio processing, biomedical signal filtering, and communications. However, traditional FIR filter implementations are power-hungry and occupy significant silicon area. Various researchers have explored techniques to optimize FIR filter design using **reversible logic** and **VLSI techniques**.

Recent research has also explored implementing these designs in older, yet widely used, technologies. While this node does not offer the same density as modern nanometer technologies, it provides a stable platform for academic research and prototyping due to its lower leakage current and mature fabrication process. Studies have shown that integrating reversible logic can lead to area-efficient and energy-saving designs suitable for low-power applications.

METHODOLOGY

CHAPTER 3

METHODOLOGY

The design of an **area-efficient reversible FIR (Finite Impulse Response) filter** follows a systematic and structured methodology. The main objective is to design a filter that minimizes power dissipation and silicon area while maintaining functional accuracy and high performance. The entire process — from design to physical implementation — was carried out using **Cadence digital design tools** for simulation, synthesis, and layout generation.

3.1 FIR Filter Specification

The first step involves defining the design parameters of the FIR filter.

- **Filter order and coefficients** were chosen based on the desired frequency response using basic DSP principles.
- The **number of taps**, **coefficient precision**, and **input bit width** were selected to balance accuracy and hardware efficiency.
- The FIR filter equation implemented is:

$$y(n) = \sum_{k=0}^{N-1} h(k) \cdot x(n-k)$$

where $h(k)$are filter coefficients and $x(n)$are the input samples.

- The FIR architecture is implemented using a **Multiply-Accumulate (MAC)** structure to perform the convolution efficiently.

3.2 Reversible Logic Design

The design employs **reversible logic** to achieve low-power and area-efficient computation. Unlike traditional logic, reversible logic preserves information, minimizing energy loss according to **Landauer's principle**.

- **Gate**

  The design uses a combination of reversible logic gates:

  - **Feynman Gate** – for fan-out and XOR operations.
  - **Toffoli Gate** – for AND and universal logic functions.
  - **Fredkin Gate** – for conditional swapping operations.
  - **Peres Gate** – for efficient full adder design.

- **Reversible Arithmetic Design:**

  - Reversible **adders** and **multipliers** are implemented using these gates.
  - The goal is to minimize **garbage outputs**, **ancilla inputs**, and **quantum cost** to improve circuit efficiency.

  These reversible logic modules form the core arithmetic units of the FIR filter, replacing traditional irreversible components.

3.3 FIR Filter Architecture Design

The **reversible FIR filter architecture** consists of several stages connected in a pipeline structure.

- Each stage performs multiplication of input samples with filter coefficients using reversible multipliers, followed by accumulation using reversible adders.
- The design ensures that no information is lost between stages, preserving reversibility.
- The structure supports scalability for higher-order filters without significant increases in area or delay.

3.4 HDL Modeling and Simulation in Cadence

The complete FIR filter was modeled using **Verilog HDL** and simulated in **Cadence Xcelium**.

- Each reversible gate and arithmetic module (adder, multiplier) was described at the behavioural level in HDL.
- The top-level FIR filter was created by interconnecting these modules.

- Functional simulation was performed in **Cadence Xcelium** to verify correct data flow and filter output response.
- The output values were compared with theoretical results to ensure correct functionality and timing behavior.

3.5 Synthesis Using Cadence Genus

After successful functional verification, the RTL design was synthesized using **Cadence Genus Synthesis Solution** with the **90nm standard cell library**.

- The synthesis process involves:
    - Importing the Verilog files into Cadence Genus.
    - Loading the 90nm technology library (.lib).
    - Defining constraints such as **clock frequency**, **input/output delays**, and **operating conditions**.
    - Running synthesis to convert RTL into a **gate-level netlist** optimized for area and power.
- Post-synthesis analysis includes:
    - **Area Report:** Total cell area used for implementation.
    - **Timing Report:** Verification of setup and hold constraints.
    - **Power Report:** Estimation of dynamic and leakage power.
    - **Gate Report:** A Gate Report summarizes the count and distribution of all logic gates present in the synthesized netlist.

The synthesized netlist and constraint files (.sdc) were then exported to Cadence Innovus for physical implementation.

3.6 Floorplanning and Power Planning Using Cadence Innovus

The next step is **floorplanning**, where the overall chip area and module placement are defined.

- The gate-level netlist is imported into **Cadence Innovus Implementation System**.

- The **core area** is defined based on utilization reported by Genus.
- **I/O pins** are positioned to match data flow between filter stages.
- **Power rings** and **power stripes** (VDD and VSS) are added for uniform power distribution.
- Proper spacing is maintained between standard cells to avoid congestion and ensure reliable routing.

3.7 Placement and Optimization

- Standard cells from the 90nm library are automatically placed using Innovus's **placement engine**.
- **Timing-driven placement** is used to reduce wirelength and ensure timing closure.
- Post-placement **optimization** is performed to eliminate setup and hold violations by adjusting cell sizes and buffer locations.
- **Design Rule Check (DRC)** ensures compliance with spacing and geometry rules.

3.8 Clock Tree Synthesis (CTS)

Clock distribution is implemented through **Clock Tree Synthesis** to ensure all sequential elements receive synchronized clock signals.

- The CTS process minimizes **clock skew** and **insertion delay** using H-tree or balanced tree structures.
- Buffers are inserted to strengthen the clock network and reduce delay variations.
- After CTS, **Static Timing Analysis (STA)** is performed to verify timing integrity.

3.9 Routing

The placed and clocked design is routed to create interconnections between cells.

- **Global routing** determines approximate routing paths.
- **Detailed routing** finalizes metal layer connections according to 90nm technology design rules.
- Routing is optimized to minimize congestion, delay, and crosstalk.

- Post-routing verification includes **DRC** and **LVS (Layout Versus Schematic)** checks to ensure layout correctness.

3.10 Post-Layout Optimization and Signoff

- Perform **parasitic extraction (RC extraction)** to capture wire resistance and capacitance values.
- Conduct **Post-Layout STA** to verify timing performance with real interconnect delays.
- Apply **Engineering Change Orders (ECOs)** if necessary to fix timing violations.
- Run **Power Analysis**, **IR Drop**, and **Electromigration (EM)** checks to ensure reliability.
- Once all checks pass, the final layout is **signed off** and exported as a **GDSII file** for fabrication readiness.

REVERSIBLE LOGIC GATES

CHAPTER 4

REVERSIBLE LOGIC GATES

Reversible logic gates are n-input and n-output logic devices with one-to-one mapping between input and output vectors. These gates ensure that outputs can be uniquely derived from inputs and, conversely, inputs can be perfectly reconstructed from outputs. Direct fan-out is not permitted in reversible circuit synthesis because one-to-many mapping violates reversibility; therefore, fan-out must be implemented using additional reversible gates. As highlighted in prior studies, including the work by Anjulata Choudhary, Nashrah Fatima and Paresh Rawat PhD (2016), "Survey paper on FIR Filter using Programming Reversible Logic Gate", International Journal of Computer Applications, Volume 151 – No.11, reversible logic has emerged as a promising research direction for designing low-power digital circuits.

Reversible logic has found significant applications in low-power CMOS design, cryptography, optical information processing, and nanotechnology. Information loss occurs when input vectors cannot be reconstructed from a circuit's output vectors, which consequently leads to unnecessary energy dissipation. One of the primary design constraints in reversible logic is to minimize the number of gates used and to reduce garbage outputs—outputs that are required for maintaining reversibility but do not contribute to the main computation. Therefore, an effective logic synthesis technique using reversible gates should adhere to the following principles:

- Use minimum number of garbage outputs
- Use minimum input constants
- Keep the length of cascading gates minimum
- Use minimum number of gates

4.1 Feynman Gate

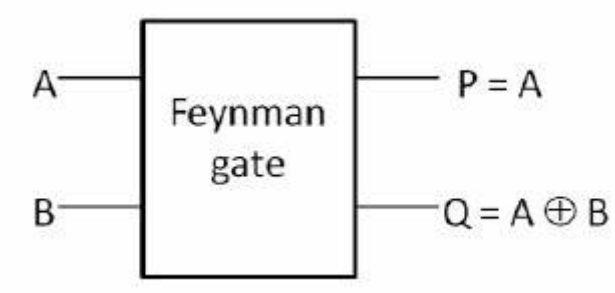

Fig 4.1.1 Feynman Gate

Feynman gate is shown in Fig.4.1.1. The input vector is (A, B) and the output vector is (P, Q). The outputs are P=A, Q=A^B. The quantum cost of a Feynman gate is 1. If A=0 then the output follows the input B. If A=1 then the input follows the output Q. So, the gate is called as copying gate.

4.2 Tofolli Gate

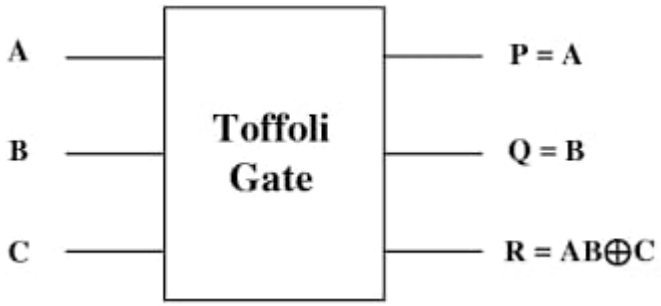

Fig 4.2.1 Tofolli Gate

Tofolli gate is one of the example for (3, 3) reversible gates. Fig.4.2.1 shows the Tofolli gate. This gate is two through gate because two of its outputs are identical with its inputs. Because of this, Tofolli gate is also known as two controlled NOT (2-CNOT). If the first two input bits are one, then the third output bit is the inverse of third input bit i.e., A = B = 1, then R = C. Tofolli gate performs basic AND operation when zero is given as its third input (C = 0; R = AB). Any reversible gate has an inverse or dual. The dual of Tofolli gate is also a Tofolli gate and so it is self-reversible.

4.3 Fredkin Gate

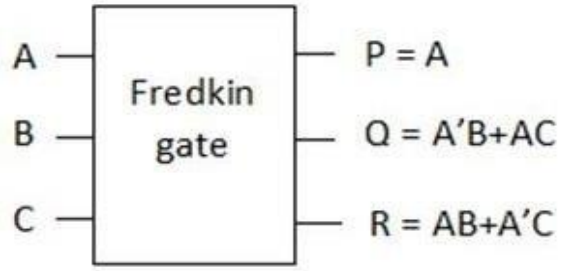

Fig 4.3.1 Fredkin Gate

The Fredkin gate is a type of reversible logic gate that has three inputs and three outputs. It works like a controlled swap gate. One input (A) acts as the control. If A is 0, the other two inputs (B and C) pass through to the outputs without changing. But if A is 1, the gate swaps B and C. So, the output depends on the value of A. This gate is called "reversible" because the outputs can always be used to find the original inputs—no information is lost.

4.4 Peres Gate

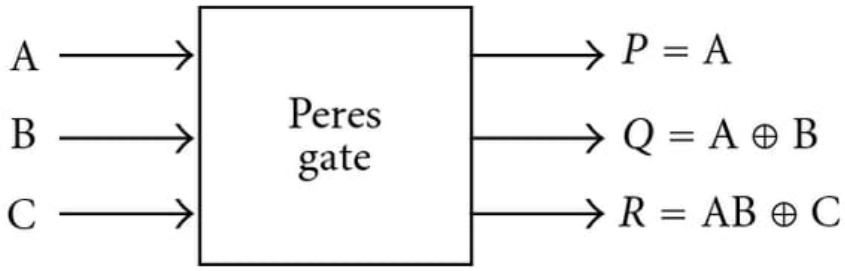

Fig 4.4.1 Peres gate

The Peres gate is a type of reversible logic gate commonly used in quantum computing and low-power VLSI design. It is a 3-input, 3-output gate, often denoted as PG(A, B, C) = (P = A, Q = A ⊕ B, R = AB ⊕ C). The gate combines the functionality of a Feynman gate and a Toffoli gate, making it efficient for circuits where fan-out and control logic are important. The Peres gate is particularly valued for its low quantum cost (typically 4) and minimal garbage outputs, which are critical factors in reversible and quantum circuit design.

4.5 PFAG (Peres Full Adder Gate)

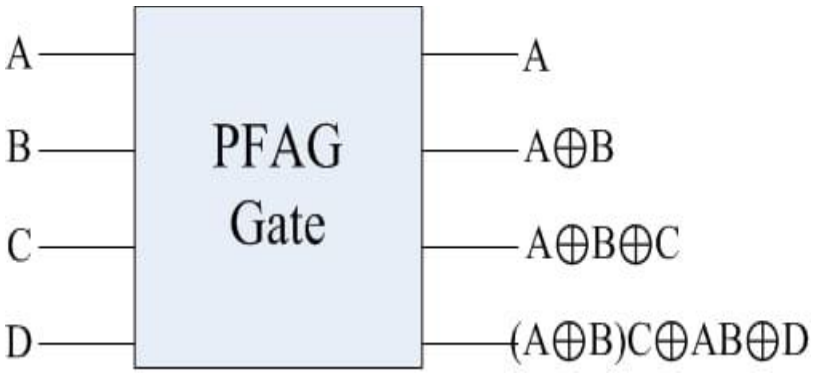

Fig 4.5.1 PFAG

The PFAG gate (Peres Full Adder Gate) is a 4-input, 4-output reversible logic gate commonly used in reversible computing and quantum logic design, particularly in low-power VLSI and nanotechnology circuits. This gate is a reversible full adder because it performs full adder operations with reversibility—no information is lost (important for quantum and low-power circuits) Used in designing low-power arithmetic circuits.

REVERSIBLE LOGIC APPROACH FOR FIR FILTER

CHAPTER 5

REVERSIBLE LOGIC APPROACH FOR FIR FILTER

A Finite Impulse Response (FIR) filter is a type of digital filter widely used in signal processing for applications like noise reduction, signal smoothing, and feature extraction. The term "finite impulse response" refers to the fact that the filter's response to a single input impulse settles to zero in a finite number of steps. FIR filters work by applying a fixed set of coefficients (also called "taps") to the current and past input samples. The output at any time is calculated as a weighted sum of the current and a finite number of previous input values.

Mathematically, an FIR filter of order N computes its output using the formula:

$y[n] = C_0 \cdot x[n] + C_1 \cdot x[n-1] + ... + C_n \cdot x[n-N]$,

where x[n] is the current input, C_0 to C_n are constant coefficients that define the filter's behaviour, and y[n] is the output.

5.1 5-tap FIR Filter

A **5-tap Finite Impulse Response (FIR) filter** is a digital filter used in signal processing applications such as noise reduction, data smoothing, and signal enhancement. The term "5-tap" indicates that the filter uses **five coefficients (or taps)**, each associated with a delayed version of the input signal. Each tap corresponds to a multiplication of the current or past input samples by a predefined coefficient, and the results are summed to generate the output.

FIR filters are preferred for many real-time digital signal processing applications because of their **inherent stability, linear phase response**, and **predictable behaviour**. They do not use feedback, which eliminates the risk of instability common in Infinite Impulse Response (IIR) filters.

In a 5-tap FIR filter, the output at any given time **n**, denoted as **y[n]**, is computed as a weighted sum of the current input and the four previous input values. The general mathematical expression is:

$$y[n] = h[0] \cdot x[n] + h[1] \cdot x[n-1] + h[2] \cdot x[n-2] + h[3] \cdot x[n-3] + h[4] \cdot x[n-4]$$

where:

- **x[n]** represents the current input signal,
- **x[n−1]**, **x[n−2]**, **x[n−3]**, and **x[n−4]** represent delayed input samples,
- **h[0]**, **h[1]**, **h[2]**, **h[3]**, and **h[4]** are the filter coefficients that determine the filter's characteristics, and
- **y[n]** is the output signal(Data out).

Each coefficient defines how much influence a particular input sample has on the output. By carefully selecting these coefficients, the filter can be configured to act as a **low-pass**, **high-pass**, **band-pass**, or **band-stop** filter depending on the application.

BLOCK DIAGRAM

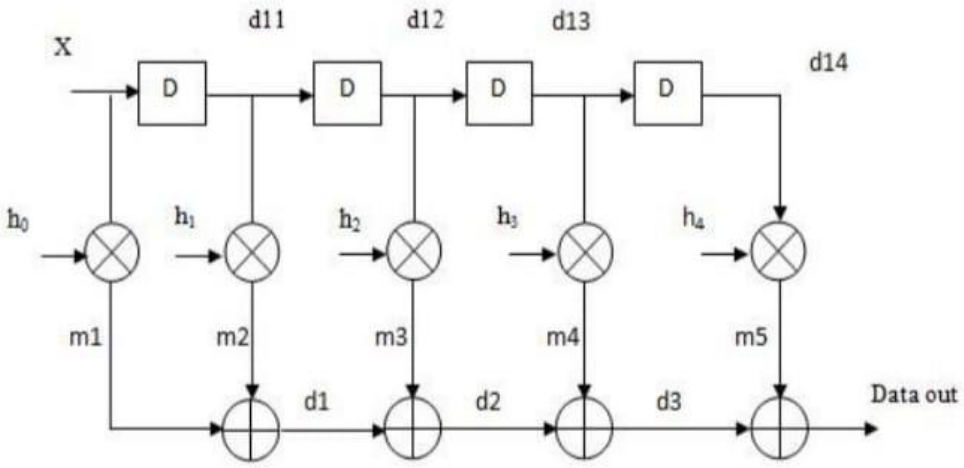

Fig.5.1.1 5-tap FIR Filter

5.2 5-Tap FIR Filter Using Reversible Gates

A 5-tap FIR (Finite Impulse Response) filter using reversible logic combines digital signal processing principles with low-power reversible computing techniques to achieve high-speed and energy-efficient operation. Unlike conventional FIR filters that employ irreversible arithmetic units, this design replaces them with reversible components such as Vedic multipliers and Lookahead Carry Save Adders (LCSA) constructed using reversible logic gates. This methodology aligns with the foundational concepts established Bennett, C.H (1973), "Logical reversibility of Computation", IBM J. Research and Development, 17, pp 525-532, where he demonstrated that logically reversible computation eliminates the intrinsic energy dissipation associated with information loss. By leveraging these principles, the reversible 5-tap FIR filter preserves information and significantly reduces energy consumption, making it highly suitable for next-generation quantum and low-power VLSI systems.

5.2.1 Reversible Design Approach

The reversible design approach of a 5-tap FIR filter integrates energy-efficient and information-preserving computation techniques using reversible logic. The delayed input samples $x[n-1]$to $x[n-4]$are generated through reversible D flip-flops or latches, ensuring that no data is lost between clock cycles and maintaining low power consumption. Each input sample is then multiplied by its corresponding coefficient using a reversible Vedic multiplier, which operates based on the *Urdhva Tiryakbhyam* sutra (vertical and crosswise method) for fast and parallel computation. These multipliers are constructed using Peres, Toffoli, and Feynman gates, providing minimal delay, reduced garbage outputs, and fewer ancilla inputs, thereby enhancing the overall speed and area efficiency of the design compared to traditional reversible multipliers.

The partial products obtained from the five multipliers are combined using a reversible Lookahead Carry Save Adder (LCSA), which performs addition with reduced carry propagation delay by computing carry signals in parallel. This adder is implemented using Fredkin and Peres gates, achieving low quantum cost, high throughput, and scalability for multi-bit operations in FIR filter architectures. Finally, the summed output from the reversible adder network produces the final filter output $y[n]$, which is stable, precise, and consistent with the expected FIR filter response. The entire reversible implementation consumes significantly less power than its irreversible counterpart while maintaining accurate signal processing and adhering to the principles of reversible and low-power VLSI design.

SIMULATION AND SYNTHESIS RESULTS

CHAPTER 6

SIMULATION AND SYNTHESIS RESULTS

6.1 Waveform of 5-tap FIR Filter

The below waveform represents the simulation result of a 5-tap FIR filter designed and implemented using Cadence Virtuoso at the 90 nm technology node. The waveform shows the relationship between the input signal (x_in), the output signal (y_out), and various control signals such as clock (clk), enable, and reset. When the enable signal is active and the clock pulses are applied, the FIR filter processes the sequential input samples to produce the filtered output. The output waveform (y_out) demonstrates the convolution operation, where each output value is obtained as a weighted sum of the current and previous four input samples, corresponding to the five filter taps. The smooth transition and correct timing alignment between x_in and y_out confirm the functional accuracy of the design. The observed output samples reflect the expected filtering behavior, verifying that the multiplier and adder units within the FIR architecture are operating properly. The simulation also validates that the design achieves stable operation, accurate coefficient multiplication, and consistent delay characteristics, indicating a successful implementation of the area-efficient reversible 5-tap FIR filter in the 90 nm CMOS process.

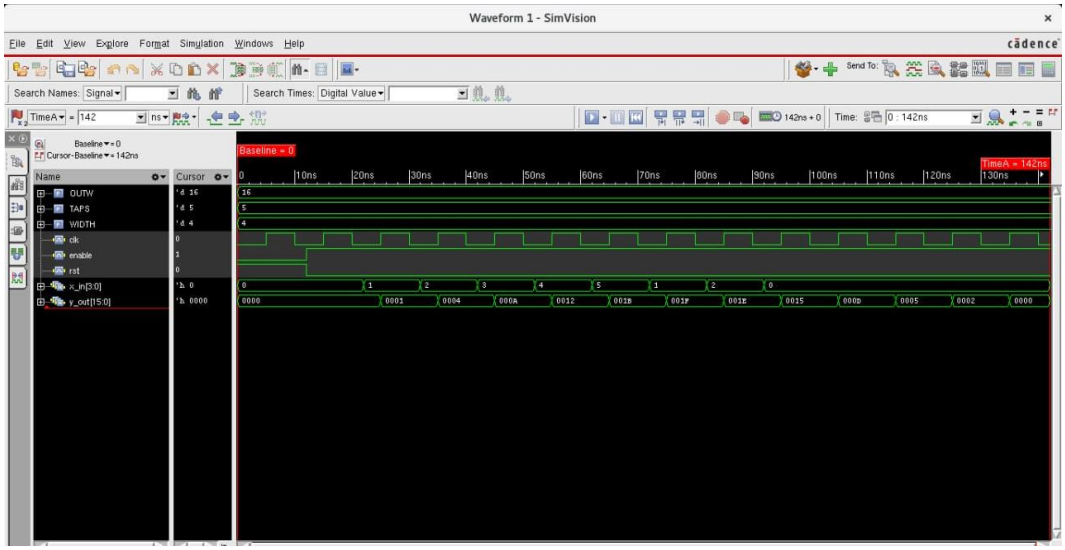

Fig 6.1.1 Waveform for 5-tap FIR Filter using Reversible Logic

6.2 Post-Synthesis Analysis

6.2.1 Area Report

This area report shows the synthesized results of the design, indicating that the reversible FIR filter implemented in 90 nm technology uses a total of 83 standard cells, resulting in a cell area of 969.589 μm² with no additional net area, leading to a total area of **969.589** μm. The compact cell count and relatively low total area demonstrate that the reversible FIR filter architecture is area-efficient, confirming that the optimized reversible logic and multiplier structures reduce hardware overhead while maintaining functional correctness during synthesis.

```
============================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 29 2025  10:19:54 am
  Module:                 fir_filter_5tap
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
============================================================

    Instance    Module  Cell Count  Cell Area  Net Area   Total Area   Wireload
-------------------------------------------------------------------------------
fir_filter_5tap                83    969.589     0.000      969.589 <none> (D)
  (D) = wireload is default in technology library
```

Fig 6.2.1 Area Report of 5 tap FIR Filter using Reversible logic

6.2.2 Timing Report

The images show detailed static timing analysis (STA) reports, which are crucial for verifying that the synthesized digital design meets its timing requirements, particularly the setup time and hold time constraints. The reports focus on critical paths—the signal paths with the tightest timing margins. The first image shows an Late External Delay Assertion check for a path ending at output pin y_out[7], also showing a positive slack of 3417 ps. The second image shows a Setup Check report for a path ending at delay_line_reg[4][2]/SI, indicating a positive slack of 9379 ps, meaning the design comfortably meets this particular setup requirement.

Both reports detail the timing calculation, including the Clock Edge (10000 ps, or 10 ns), the Required Time, the Data Path delay, and a breakdown of the delay contributed by different logic Cells (e.g., SDFFRHQX1, ADDEX1) and their associated Load, Transition, and Delay values in picoseconds (ps) and femtofarads (fF). These positive slack values confirm that the design, at least for the paths shown, is timing-clean at the specified clock frequency.

```
Path 1: MET (3417 ps) Late External Delay Assertion at pin y_out[7]
          Group: clk
     Startpoint: (R) delay_line_reg[2][1]/CK
          Clock: (R) clk
       Endpoint: (R) y_out[7]
          Clock: (R) clk

                    Capture       Launch
       Clock Edge:+   10000            0
       Src Latency:+      0            0
       Net Latency:+      0 (I)        0 (I)
           Arrival:=  10000            0

      Output Delay:-    1000
     Required Time:=    9000
      Launch Clock:-       0
         Data Path:-    5583
             Slack:=    3417

Exceptions/Constraints:
  output_delay             1000            ou_del_12_1

#-------------------------------------------------------------------------------------------------
#      Timing Point       Flags    Arc   Edge   Cell     Fanout  Load Trans Delay Arrival Instance
#                                                           (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------------
  delay_line_reg[2][1]/CK -       -      R     (arrival)     20     -     0     0       0    (-,-)
  delay_line_reg[2][1]/Q  -       CK->Q  R     SDFFRHQX1      6  14.4   147   389     389    (-,-)
  g2/Y                    -       A->Y   F     CLKXOR2X1      2   3.3    65   236     625    (-,-)
  g1789__4733/Y           -       B->Y   R     NAND2XL        1   1.7    73    56     680    (-,-)
  g1788__2802/Y           -       B0->Y  F     OAI21XL        1   4.9   154   131     811    (-,-)
  g1778__5526/S           -       CI->S  R     ADDFX1         1   4.9    99   424    1235    (-,-)
  g1772__5477/S           -       CI->S  F     ADDFX1         1   4.9   100   374    1609    (-,-)
```

```
Path 50: MET (9379 ps) Setup Check with Pin delay_line_reg[4][2]/CK->SI
          Group: clk
     Startpoint: (R) delay_line_reg[3][2]/CK
          Clock: (R) clk
       Endpoint: (R) delay_line_reg[4][2]/SI
          Clock: (R) clk

                    Capture       Launch
       Clock Edge:+   10000            0
       Src Latency:+      0            0
       Net Latency:+      0 (I)        0 (I)
           Arrival:=  10000            0

            Setup:-     248
     Required Time:=    9752
      Launch Clock:-       0
         Data Path:-     373
             Slack:=    9379

#-------------------------------------------------------------------------------------------------
#      Timing Point       Flags   Arc   Edge   Cell     Fanout Load Trans Delay Arrival Instance
#                                                           (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------------
  delay_line_reg[3][2]/CK -       -     R     (arrival)     20    -     0     0       0    (-,-)
  delay_line_reg[3][2]/Q  -       CK->Q R     SDFFRHQX1      3 11.8   128   373     373    (-,-)
  delay_line_reg[4][2]/SI -       -     R     SDFFRHQX1      3    -     -     0     373    (-,-)
#-------------------------------------------------------------------------------------------------
```

Fig 6.2.2 Static Timing Analysis Report: Critical Path and Delay Breakdown of an 5 tap FIR Filter using Reversible Logic

6.2.3 Power Report

This report presents the detailed power usage breakdown for a 5-tap FIR digital filter, evaluated after synthesis in 90nm technology. The data is sorted by hardware categories—registers, logic, memory, clock, and pads. Each is shown for three power components: leakage (static losses), internal (from circuit activity within cells), and switching (signal toggling on nets). The dominant power consumer is internal cell activity, making up 62.24% of total power. Registers and logic blocks consume almost all the total design power (around 7.34e-05 W), while memory and pad usage are negligible, indicating no explicit memory or pad structures present. The analysis reveals that most power is due to dynamic switching inside the logic, useful for designers who wish to optimize FIR filter hardware for lower power. The clock tree only contributes internal power. This comprehensive summary lets engineers easily identify and target high-impact optimization areas in advanced digital signal processing designs.

Instance: /fir_filter_5tap
Power Unit: W
PDB Frames: /stim#0/frame#0

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	3.05244e-06	3.75644e-05	1.04733e-06	4.16642e-05	56.75%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	2.17943e-06	8.12669e-06	1.86083e-05	2.89144e-05	39.39%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	2.83500e-06	2.83500e-06	3.86%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	5.23187e-06	4.56911e-05	2.24906e-05	7.34135e-05	100.00%
Percentage	7.13%	62.24%	30.64%	100.00%	100.00%

Fig 6.2.3 Power Consumption Analysis of 5 tap FIR Filter using Reversible Logic

6.2.4 Gate Count

The report use of 17 different standard cells in the design, totaling 83 instances and occupying a total cell area of 969.589 (units).

The instance count and area for each cell type are as follows:

ADDFX1: 11 instances, 216.473 area, ADDMXL: 3 instances, 36.331 area,

AND2X1: 1 instance, 4.541 area, AOI22X1: 2 instances, 9.083 area,

AOI22XL: 1 instance, 6.055 area, CLKXOR2X1: 3 instances, 24.978 area,

INVXL: 4 instances, 9.083 area, MUX2X1: 5 instances, 30.276 area,

NAND2X1: 4 instances, 18.166 area, NAND2XL: 14 instances, 42.386 area,

NOR2X1: 1 instance, 4.541 area,NOR2XL: 1 instance, 4.541 area,

NOR3X1: 3 instances, 9.083 area, OR3BXL: 1 instance, 4.541 area,

OAI21XL: 7 instances, 31.790 area, SDFFRHMX1: 20 instances, 499.554 area,

XNOR2X1: 1 instance, 16.652 area

```
================================================================
  Generated by:           Genus(TM) Synthesis Solution 21.14-s082_1
  Generated on:           Sep 29 2025  10:19:54 am
  Module:                 fir_filter_5tap
  Operating conditions:   slow (balanced_tree)
  Wireload mode:          enclosed
  Area mode:              timing library
================================================================

   Gate    Instances   Area   Library
-----------------------------------------
ADDFX1             11  216.473    slow
ADDHXL              3   36.331    slow
AND2X1              1    4.541    slow
AOI21XL             2    9.083    slow
AOI22XL             1    6.055    slow
CLKXOR2X1           3   24.978    slow
INVXL               4    9.083    slow
MXI2XL              5   30.276    slow
NAND2BXL            4   18.166    slow
NAND2XL            14   42.386    slow
NOR2BX1             1    4.541    slow
NOR2BXL             1    4.541    slow
NOR2XL              3    9.083    slow
NOR3BXL             1    6.055    slow
OAI21XL             7   31.790    slow
SDFFRHQX1          20  499.554    slow
XNOR2X1             2   16.652    slow
-----------------------------------------
total              83  969.589
```

Fig 6.2.4 Gate Count

6.2.5 Performance Comparison of Standard Adders and Multipliers

Component Type	Sub-Type	Delay(ps)	Area(μm²)	Power(%)
Adder	CSA(Carry Save Adder)	Input Delay=1000 OutputDelay=1000 Slack=7827	157.435	Leakage=15.92 Internal=49.84 Switching=34.24
Adder	RCA(Ripple Carry Adder)	InputDelay=1000 OutputDelay=1000 Slack=7644	157.435	Leakage=12.11 Internal=62.73 Switching=25.16
Adder	LCSA(Look ahead Carry Save Adder)	InputDelay=1000 OutputDelay=1000 Slack=7827	153.651	Leakage=12.51 Internal=38.09 Switching=49.40
Multiplier	Wallace Multiplier	InputDelay=1000 OutputDelay=1000 Slack=7888	277.025	Leakage=14.38 Internal=56.85 Switching=28.85
Multiplier	Vedic Multiplier	InputDelay=1000 OutputDelay=1000 Slack=7917	224.042	Leakage=14.38 Internal=57.60 Switching=26.23

Table 6.2.5 Performance Comparison of Standard Adders and Multipliers

6.3 Schematic

The schematic shown below represents the gate-level implementation of the designed area-efficient reversible 5-tap FIR filter synthesized using Cadence Genus in 90 nm CMOS technology. It shows how various standard cells such as logic gates, flip-flops, and interconnections form the complete circuit. The schematic helps visualize signal flow and verify the logical connectivity between different functional blocks like multipliers, adders, and control units. It confirms that the design has been correctly synthesized, optimized for area and power, and is ready for layout and physical verification stages.

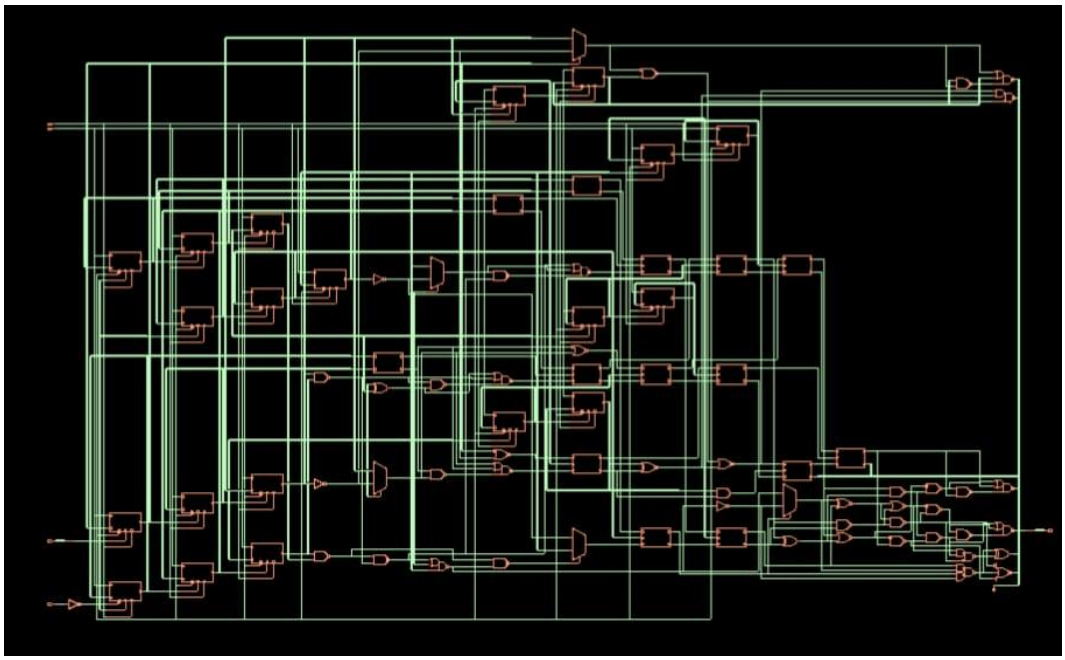

Fig 6.3.1 Schematic of 5 tap FIR Filter using Reversible Logic

6.4 Layout

The below diagram represents the final layout view of the 5-Tap FIR Filter designed using Cadence Innovus Implementation System in 90 nm CMOS technology. This layout is the completed version of the physical design process, which includes all major steps such as floorplanning, placement, routing, and power planning.The layout shows the metal layers, vias, and interconnections that form the physical connections between standard cells. The horizontal blue lines represent metal routing layers used for signal and power distribution, while the vertical connections (seen in cyan) represent vias and inter-metal contacts.

The brown boundaries indicate the core and boundary regions, where input/output pads and power rails are placed. This layout ensures that the entire circuit meets timing, power, and area constraints, and passes all physical verification checks such as Design Rule Check (DRC) and Layout Versus Schematic (LVS). As all design and verification steps have been completed, this represents the final and complete layout ready for fabrication.

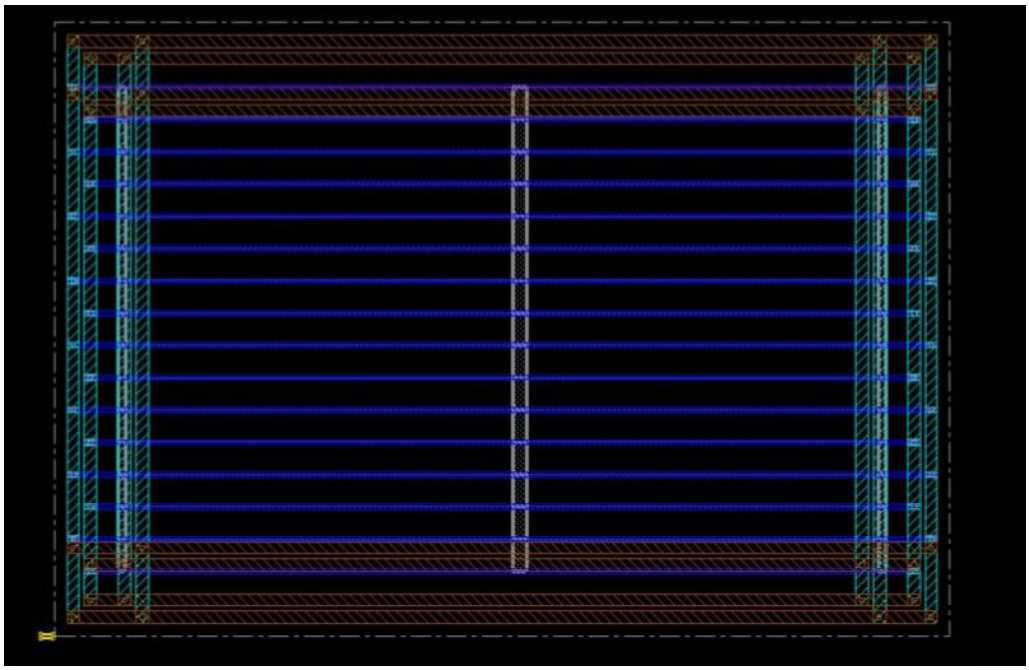

Fig 6.4.1 Layout of 5 tap Fir Filter using Reversible Logic

6.5 Device Utilization summary

The Utilization Summary is a critical report generated by Electronic Design Automation (EDA) tools during the implementation of digital circuits onto programmable hardware like FPGAs (Field-Programmable Gate Arrays). Its purpose is to quantify the design's hardware footprint by comparing the amount of each resource used by the circuit's logic against the total available resources on the specific target chip. This report breaks down usage for key components such as Look-Up Tables (LUTs) for combinational logic, Flip-Flops for sequential logic, and I/O Blocks (IOBs) for external communication. Ultimately, the summary yields a utilization percentage, which informs the designer whether the design fits onto the chip and serves as a vital metric for assessing the area-efficiency and complexity of the resulting hardware implementation.

Project File:	fff.xise	**Parser Errors:**	No Errors
Module Name:	simple_fir_reversible	**Implementation State:**	Placed and Routed
Target Device:	xc3s50-5pq208	**• Errors:**	No Errors
Product Version:	ISE 14.7	**• Warnings:**	17 Warnings (17 new, 0 filtered)
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	6	1,536	1%	
Number of 4 input LUTs	3	1,536	1%	
Number of occupied Slices	4	768	1%	
Number of Slices containing only related logic	4	4	100%	
Number of Slices containing unrelated logic	0	4	0%	
Total Number of 4 input LUTs	3	1,536	1%	
Number of bonded IOBs	8	124	6%	
Number of BUFGMUXs	1	8	12%	
Average Fanout of Non-Clock Nets	2.33			

Fig 6.5.1 Device Utilization Summary of 5 tap FIR Filter using Reversible Logic

ADVANTAGES AND APPLICATIONS

CHAPTER 7

ADVANTAGES AND APPLICATIONS

7.1 Advantages

I. Low Power Consumption

Reversible logic circuits ideally do not lose information, which means they avoid the energy loss that occurs in conventional (irreversible) logic due to bit erasure. This makes FIR filters based on reversible gates highly energy-efficient, especially important in battery-operated and portable devices.

II. Quantum Computing Compatibility

Reversible logic is a **requirement for quantum computing**, where operations must be invertible. Designing FIR filters using reversible gates like Peres gates makes them directly translatable into quantum logic circuits, useful for future quantum signal processing systems.

III. Information Preservation

In reversible FIR filters, all input data can be recovered from the output, which ensures no loss of computation data. This can be beneficial in critical applications such as cryptography, error correction, and fault-tolerant systems.

IV. Reusable Intermediate Results

Because reversible logic allows for traceable data paths, intermediate results (such as partial products or sums) can be reused or reversed efficiently, improving hardware reusability and debugging visibility.

V. Scalability for Low-Power Architectures

Reversible FIR filter designs are easily scalable for multi-bit or higher-order filters while maintaining low power characteristics, making them suitable for embedded systems and IoT applications where power and space are limited.

7.2 Applications

I. Quantum Signal Processing

Reversible FIR filters are ideal for quantum computing systems, where every operation must be reversible. They are used in quantum versions of signal processing tasks such as filtering, modulation, and error correction.

II. Low-Power VLSI Systems

In battery-powered and portable devices, such as wearables, mobile phones, and hearing aids, power efficiency is critical. Reversible FIR filters significantly reduce power dissipation, making them suitable for low-power embedded systems.

III. Biomedical Signal Processing

Applications like ECG, EEG, and EMG signal filtering benefit from FIR filters for noise reduction. Using reversible logic further enables ultra-low power medical devices that can be used in body-worn or implantable systems.

IV. IoT and Edge Devices

FIR filters are widely used in sensor signal conditioning in IoT devices. Reversible FIR filters help reduce power consumption in edge devices that operate on limited energy and must process data locally.

V. Aerospace and Satellite Systems

Reversible FIR filters are useful in space electronics, where heat dissipation and power efficiency are critical due to the harsh and resource-limited environment.

CONCLUSION AND FUTURE SCOPE

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The design and implementation of a 5-Tap Reversible FIR Filter using 90 nm CMOS technology successfully demonstrate the effectiveness of reversible logic in achieving low-power and area-efficient digital signal processing. By integrating Vedic and Wallace multipliers along with optimized adder architectures such as LCSA, CSA, and RCA, the design achieves significant improvements in speed, power efficiency, and silicon area utilization compared to conventional FIR filter designs.

The reversible logic approach ensures minimal energy dissipation by preserving information during computation, which is highly beneficial for next-generation VLSI and low-power applications. The use of Cadence tools for schematic design, simulation, and layout generation enables accurate physical realization and verification of the circuit, ensuring that it meets all functional and timing requirements.

Overall, the proposed reversible FIR filter design proves to be a promising solution for energy-efficient digital computation. It not only enhances system performance but also contributes to the advancement of quantum-compatible and low-power VLSI architectures, opening new possibilities for future digital and quantum signal processing systems.

8.2 Future Scope

I. Quantum Computing Integration

As quantum computing becomes more practical, reversible FIR filters can serve as foundational components in quantum signal processing. Since quantum circuits must be reversible by nature, these filters are directly compatible with quantum algorithms and quantum communication systems.

II. Development of Low-Power DSP Hardware

There is growing demand for ultra-low-power digital signal processing (DSP) units in IoT, wearable devices, and implantable medical electronics. Reversible FIR filters can be optimized and scaled for use in DSP chips that operate on minimal power without compromising performance.

III. Reversible Arithmetic and Logic Units (ALUs)

FIR filter architectures using reversible gates contribute to the design of larger reversible computing blocks, such as reversible ALUs, which are essential for building complete reversible processors or green computing systems.

IV. Nano and Optical Computing

Reversible logic is also compatible with emerging technologies such as quantum-dot cellular automata (QCA) and optical computing, where conventional logic fails due to high energy dissipation. FIR filters designed with Peres gates could be directly applied in these next-gen nano-scale platforms.

V. Secure and Traceable Systems

With data security being a major concern, reversible FIR filters may find applications in secure signal processing systems, where traceability and reversibility of computations enhance data integrity and auditability.

REFERENCES

CHAPTER 9

REFERENCES

1. Jour of Adv Research in Dynamical & Control Systems, Vol. 10, 13-Special Issue, 2018
2. International Journal of VLSI System Design and Communication Systems Volume.04, IssueNo.11, October-2016, Pages: 1192-1198
3. Anjulata Choudhary, Nashrah Fatima and Paresh Rawat PhD (2016), "Survey paper on FIR Filter using Programming Reversible Logic Gate", International Journal of Computer Applications, Volume 151 – No.11.
4. Bennett, C.H (1973), "Logical reversibility of Computation", IBM J. Research and Development, 17, pp 525-532.
5. Bhagyalakshmi, V, Venkatesha. M.K (2010), "An Improved Design of A multiplier using Reversible logic gates", International Journal of Engineering Science and Technology, Vol. 2(8), 3838-3845.
6. Bruce.J.W, Thornton.M. A,Shivakumaraiah.L, Kokate.P.S, and X. Li, "Efficient Adder Circuits Based on a Conservative Reversible Logic Gate