International Journal on Science and Technology (IJSAT)
E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Design Of Combinational Circuits and Multipliers Using Reversible Logic gates

Sayan Mukherjee [1], Sutapa Trivedy [2], Anurag Laha[3], Shounak Bhattacharya[4], Dr. Asima Adak[5], Dr. Anindya Sen[6]

[1,2,3,4]Students, [5]Assistant Professor, [6]Professor
[1,2,3,4]Department of Electronics and Communication Engineering
[1,2,3,4]Heritage Institute of Technology, Kolkata-700107, India
Corresponding Author: asima.maity@heritageit.edu

1. Abstract:

Reversible logic design has come out as a promising area of research in the field of modern computing due to its capability to reduce energy dissipation. In traditional logic circuits, energy loss occurs mainly due to information loss. However, reversible logic guarantees that no information is lost during computing, aligning with Landauer's principle of low-power computing. This property makes reversible logic ideal for applications where power efficiency is to be maintained. This paper deals with the design and synthesis of circuits using reversible logic, primarily focusing on key reversible gates such as the Feynman Gate, Fredkin Gate, Toffoli Gate, Peres Gate. These gates serve as the key element for more complex systems and circuits. The paper traverses how these gates can be used to design efficient combinational circuitsshowcasing their versatility and practicality.Through simulation and theory, the paper evaluates the performance of these circuits in terms of gate count, garbage outputs, and quantum cost.The results of this project demonstrate the possibility of adopting reversible logic for sustainable and low-power electronic systems, enhancing its importance in future technological innovations.

Keywords: Reversible logic gates, Half Adder, Full Adder, Half Subtractor, Full Subtractor, Multiplexer, 1 bit comparator, Encoder, decoder, Parity Generator and Multiplier.

2. INTRODUCTION

In the age of miniaturization of advanced computers and electronic devices, the demand for low-power circuits has become something of prime importance. Reversible logic has proven to be an innovative concept, offering great advantages in minimizing power dissipation—an important requirement in Very Large-Scale Integration (VLSI) systems[1]. The principle of reversible logic addresses the energy dissipation inherent in conventional irreversible computing, thereby making way for sustainable systems. Conventional computing systems suffer from energy loss due to the irreversibility of logic operations. According to Landauer's law, the removal of one bit of information results in an energy loss of $KT \ln(2)$ where k is Boltzmann's constant and T is the absolute temperature of the system[1]. This energy loss proves to be an inefficiency in high-density VLSI circuits, making it a bottleneck for further miniaturization. Reversible logic overcomes this bottleneck by ensuring that no information is lost during computation, making it a trustworthy approach for designing low-power systems[1][2].This paper

presents the design, synthesis, and optimization of reversible logic circuits, making use of fundamental gates such as Feynman, Toffoli, Fredkin, Peres[4],[5],[6],[7]. These gates serve as the foundation for complex circuits, using which complex combinational can be easily constructed. The study highlights the optimization of important metrics such as quantum cost, garbage outputs, gate levels, and hardware complexity and power dissipation calculations.

3. Literature Review

In conferrance with Moore's Law, the number of transistors on integrated circuits doubles in about every eighteen months, enabling the development of higher-performance general-purpose processors within the exact timeframe. This trend has significantly enhanced computational capabilities but has also aggravated the problem of power dissipation in traditional systems[1].Tommaso Toffoli proposed that using reversible logic gates, it is possible to create sequential systems with zero internal power loss as the inputs can be reproduced from the outputs. This pioneering idea laid the building blocks for reversible logic design, suggesting that reversible gates could lead to energy-efficient computation by removing information loss[2].In 1961, Rudolf Landauer established a basic relationship between logical and physical irreversibility. He proposed that for irreversible logic computations, each bit of information lost generateskTln(2) joules of heat energy, where k is Boltzmann's constant and T is the absolute temperature[1]. Landauer's principle accentuated that heat dissipation could be avoided if computational systems were reversible, underscoring the importance of reversible logic in reducing power dissipation[1].In 2008, Majid Mohammdi et al. brought in quantum gates for implementing binary reversible logic gates. The authors used quantum gates V and V^+ , representing them in truth table, and proposed a new behavioral model. This model was used to simulate the quantum realization of reversible logic designs[4].In 2010, D. Michael Miller and Zahra Sasanian worked on reducing the quantum gate costs of reversible designs[6]. Their work underscored the following: The circuit generation composed of binary reversible gates, Mapping these circuits to similar quantum gate realizations for improved and increased efficiency, Their contributions emphasized the importance of reducing quantum costs in increasing the efficiency of reversible logic circuits. In2013, Raghava Garipelly worked on the application of basic reversible logic gates in designing complex systems. He proposed how reversible circuits could work as primary components in quantum computers, executing complex operations with increased efficiency. The introduction of gates, such as BSCL, SBV, NCG, and PTR, expanded the toolkit for designing more complicated reversible circuits [8,9].In 2014, Ashima Malhotra et al. demonstrated the use of a reversible modified Fredkin gate for designing multiplexers with lower quantum costs[9]. Comparisons were made with the existing multiplexer designs, highlighting the advantages of the modified gate.

4. Gates and Circuits Used in Design

4.1 Basic Reversible Logic Gates

- **Feynman Gate:** The Feynman Gate, also called the Controlled NOT (CNOT) gate, is a 2-input, 2-output reversible gate[3].

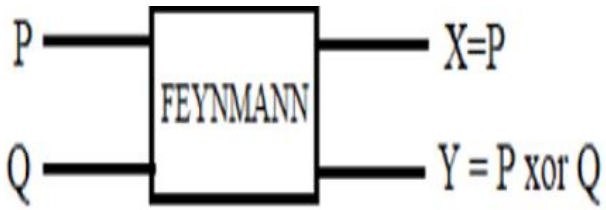


Fig 1. Feynman Gate

- **Double Feynman Gate:** The Double Feynman Gate is an extension on the Feynman Gate with three inputs and three outputs[3].

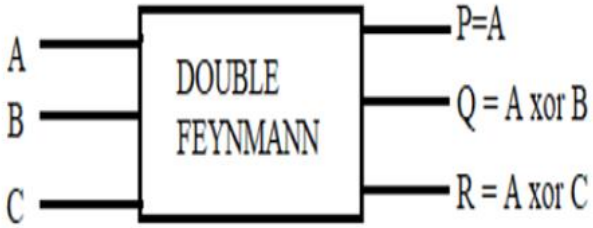


Fig 2. Double Feynman Gate

- **Toffoli Gate:** The Toffoli Gate, also called the Controlled-Controlled NOT (CCNOT) gate, is a 3-input, 3-output reversible gate[3].

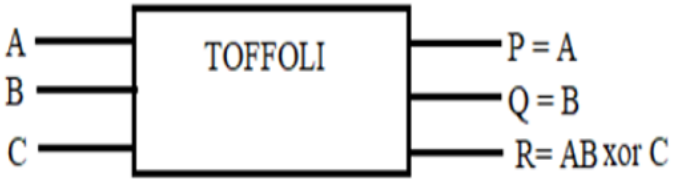


Fig 3. Toffoli Gate

- **Fredkin Gate:** The Fredkin Gate, also called the Controlled-Swap gate, is a 3-input, 3-output reversible gate[3].

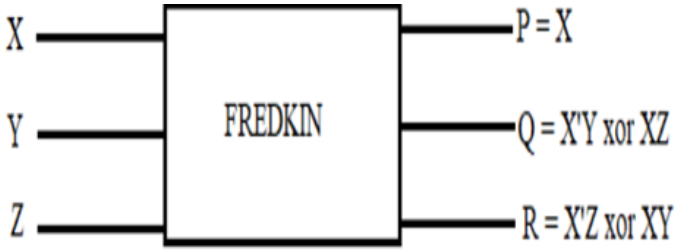


Fig 4. Fredkin Gate

- **Peres Gate:** The Peres Gate is a 3-input, 3-output reversible gate that includes features of both Toffoli and Feynman gates[3].

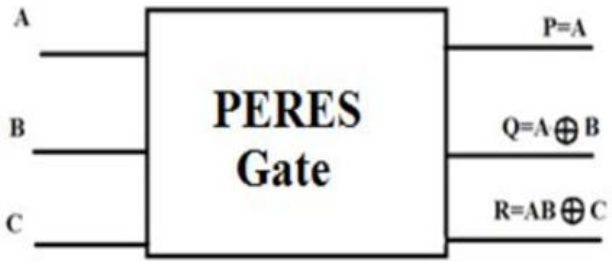


Fig 5. Peres Gate

- **URG Gate:** The URG gate is a 3×3 reversible logic gate with outputs P=(A+B)⊕C, Q=B, and R=AB⊕C, used in low-power circuit design[3].

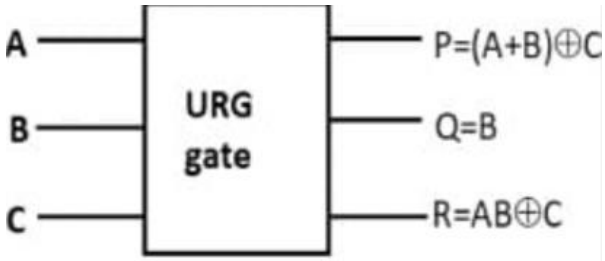


Fig 6. URG Gate

- **Modified Fredkin Gate (MFRG):** The Modified Fredkin Gate (MFRG) is a 3×3 reversible logic gate that conditionally swaps its last two outputs based on the control input, optimized for reduced quantum cost and garbage outputs[9].

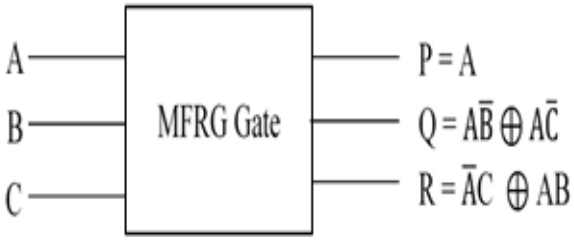


Fig 7. Modified Fredkin Gate

4.2 Basic Combinational Gates:

- **Half Adder:** Combinatuional circuit that adds two single-bit binary numbers (A and B) and produces two outputs: Sum (S): XOR operation of A and B, S=A⊕B. Carry (C): AND operation of A and B, C= A.B.

The Peres Gate[3] is used in designing half adder, as it can efficiently produce both sum and carry.

A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth Table:

Fig 8. Truth Table of Half Adder

- **Full Adder:** Combinational circuit that adds three single-bit binary numbers (A, B, and Cin) and produces two outputs: **Sum (S)**: XOR operation of A, B, and Cin, S=A⊕B⊕Cin.
Carry(Cout):Carry-outbit generated during the addition.

Cout=AB+(A⊕B)·Cin

Two Peres Gates[3] are used in designing a full adder efficiently.

Truth Table:

A	B	Cin	SUM	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig.9 Truth Table for Full Adder

- **Half Subtractor:**Combinational circuit that subtracts one single-bit binary number (B) from another (A) and produces two outputs:
Difference (D): XOR operation of A and B. D=A⊕B
Borrow (Borrow): Represents the logical AND of B with the complement of A. Borrow=A`·B
A half subtractor is implemented efficiently using a **Feynman Gate**[3]and a **Fredkin Gate**[3], while adhering to reversible logic principles.

Truth Table:

A	B	DIFFERENCE	BORROW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Fig.10 Truth Table for Half Subtractor

- **Full Subtractor:** Combinational logic circuit that subtracts three single-bit binary numbers: the minuend (A), the subtrahend (B), and the borrow-in (Bin), producing two outputs:

Difference (D): XOR of A, B, and Bin. $D=A \oplus B \oplus Bin$
Borrow-out (Bout): Represents whether a borrow is generated during the subtraction. This is the result of B being greater than A which can be expressed as: $Bout=A` \cdot B+(A` \cdot Bin)+(B \cdot Bin)$

The **Fredkin Gate**[3] and the **Peres Gate**[3] are used to efficiently calculate the difference and borrow-out of a full subtractor.

Truth Table:

A	B	Bin	DIFFERENCE	BORROW
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Fig 11. Truth Table for Full Subtractor

- **2:4 Decoder:** Combinational circuit that converts 2 binary inputs into 4 unique outputs. For each input combination, only one output line is activated at a time(set to 1) while all others remain 0. It decodes the 2-bit binary input into one of four outputs.

Inputs: A0, B1

Outputs: D0, D1, D2, D3

The Peres[3] and Fredkin[3] gates are used to efficiently implement the circuit.

Truth Table:

A0	A1	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Fig 12. Truth Table for 2:4 Decoder

- **4:2 Encoder:** Combinational logic circuit that converts 4 inputs into 2 outputs. Performs the inverse logic operation of a decoder.

Inputs: Y0, Y1, Y2, Y3

Outputs: A0, A1

The encoder assumes that only one input is active at any given time.

The Universal Reversible Gate(URG)[13] is used to efficiently design the circuit.

Truth Table:

INPUTS				OUTPUTS	
Y_3	Y_2	Y_1	Y_0	A_1	A_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Fig 13. Truth Table for 4:2 Encoder

- **2:1 Multiplexer:** Combinational circuit that selects one of two input signals and forwards it to a single output, depending on the value of a select line.

Two data inputs: A, B
One select input: S
One output: Z

Logic Expression:

$Z=S'\cdot A + S\cdot B$

Where:

- S' is the complement of the select line S

If S = 0, then Y = A
If S = 1, then Y = B

The Modified Fredkin Gate(MFRG)[19]is used to efficiently design the circuit.

Truth Table:

S	A	B	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Fig 14. Truth Table for 2:1 Multiplexer

- **Even Parity Generator:** Combinational circuit design which is used to detect errors in digital data transmission. It makes sure that the total number of 1s in a given data word (including the parity bit) is even.

Function:
Takes n input bits and generates 1 parity bit.
Parity bit is set such that the overall number of 1s becomes even.

Logic Expression: Parity Bit $(P)=A\oplus B\oplus C$
Two Feynman Gates[3] are used for the implementation.

Truth Table:

A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Fig 15. Truth Table for Even Parity Generator

- **Odd Parity Generator:** Combinational logic design ehich is used to detect errors in digital communication. It makes sure that the total number of 1s in a given data word (including the parity bit) is odd.

Function:
Takes n input bits and produces 1 parity bit.
Parity bit is set such that the overall number of 1s becomes odd.

Logic Expression: Parity Bit (P)= (A⊕B⊕C)`
Two Feynman Gates[3] are used for the implementation.

Truth Table:

A	B	C	P
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Fig 16. Truth Table for Odd Parity Generator

- **1 bit Comparator:** Combinational circuit that compares two single-bit binary inputs, A and B, and determines their relationship[7].

Function:

It produces three outputs indicating: **A > B, A = B, A < B**

Logic Expressions:

A > B → Output = $A \cdot B'$

A = B → Output = $A' \cdot B' + A \cdot B$

A < B → Output = $A' \cdot B$

The Feynman and Fredkin gates are used in the design.

Truth Table:

A	B	A<B	A=B	A>B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Fig 17. Truth Table for 1 bit Comparator

- **2 Bit Binary Multiplier:** Digital circuit that multiplies two 2-bit binary numbers to produce a 4-bit binary result.

Let the two 2-bit numbers be: A = A1 A0, B = B1 B0

The multiplication involves partial products and their addition:

A0 × B0 → Least Significant Bit (LSB)

A1 × B0 and A0 × B1 → Added to get middle bits

A1 × B**1** → Most Significant Bit (MSB)

Output:

The result is a 4-bit number: P = P3 P2 P1 P0

Truth Table:

A0	A1	B0	B1	P0 (MSB)	P1	P2	P3 (LSB)
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Fig 18. Truth Table for 2 Bit Binary Multiplier

5. Results and Discussions

5.1 Half Adder: To implement a half adder we used a **Peres Gate**, the third input (C) is set to 0, so that the second output (Q) equivalent to the AND operation Sum(A⊕B) and the third output R corresponds to the Carry(AB)[7],[8].

Inputs: A, B, C.
Outputs:P=A, Q=A⊕B (XOR operation), R=AB⊕C.

The estimated power dissipation of AND, XOR and NOT gates are: 125mW, 250mW, 83.33mW respectively. The power dissipation of our circuit is in the range of micro-watts.

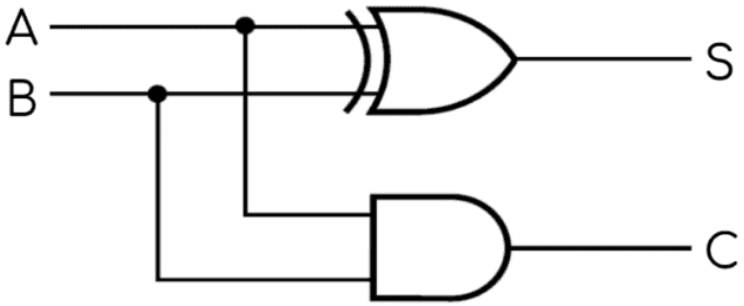


Fig 19. Half Adder Schematic

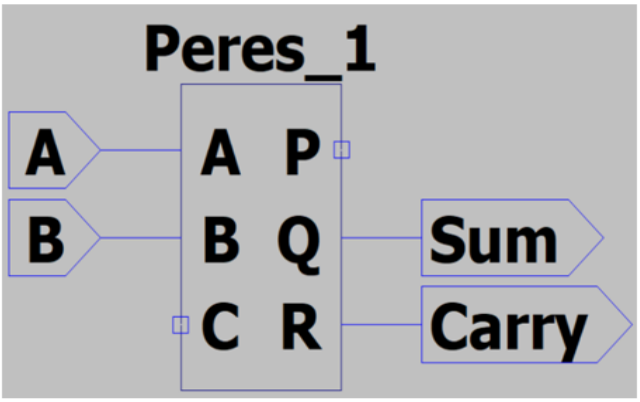


Fig 20. Implementation of Half Adder Circuit

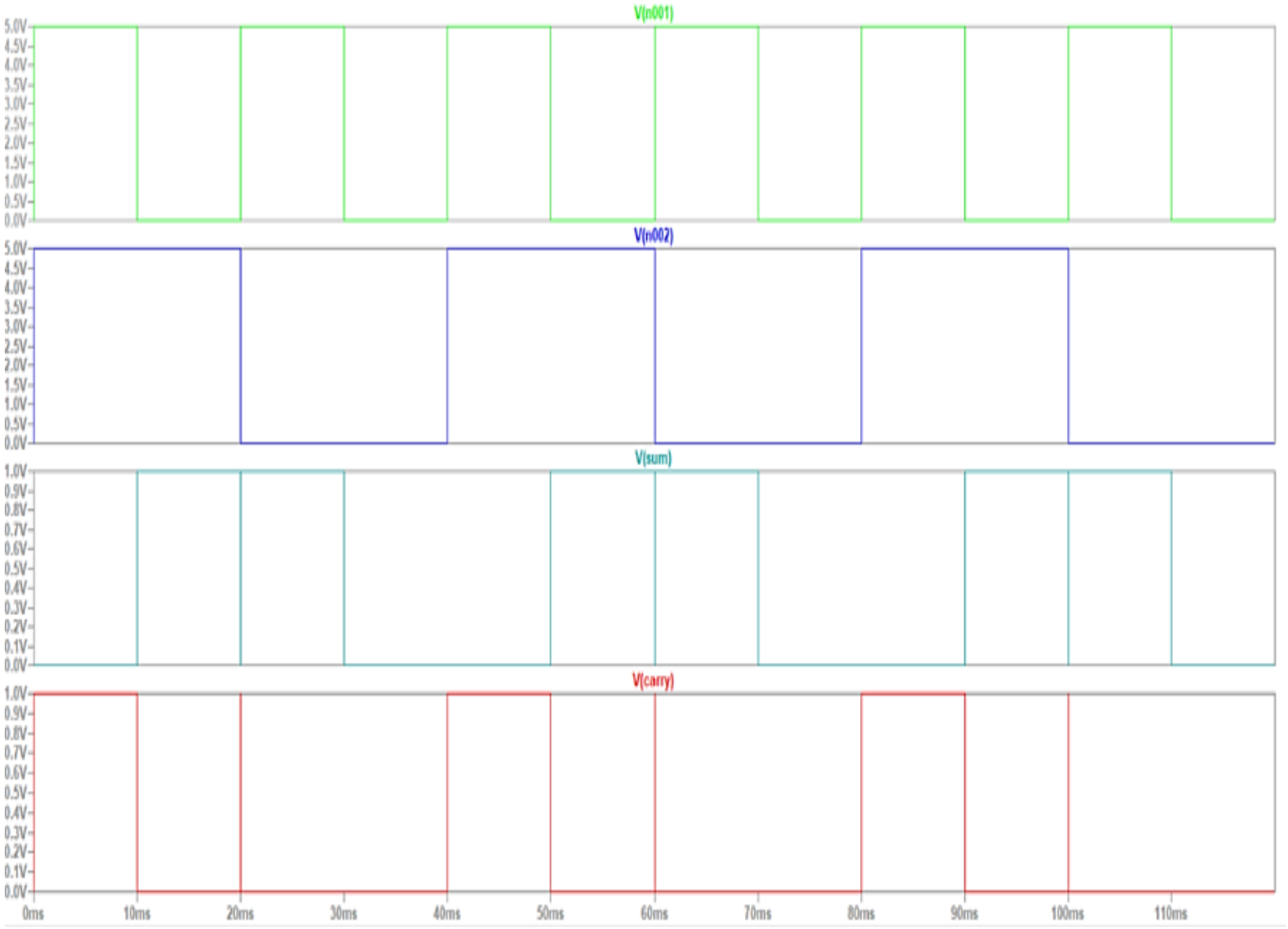


Fig 21. Output Simulation Plot for Half Adder

5.2 Full Adder: To construct a full adder two Peres Gates are used[9],[10],[11]. The first **Peres Gate** is used to find the partial sum of A and B and their carry:

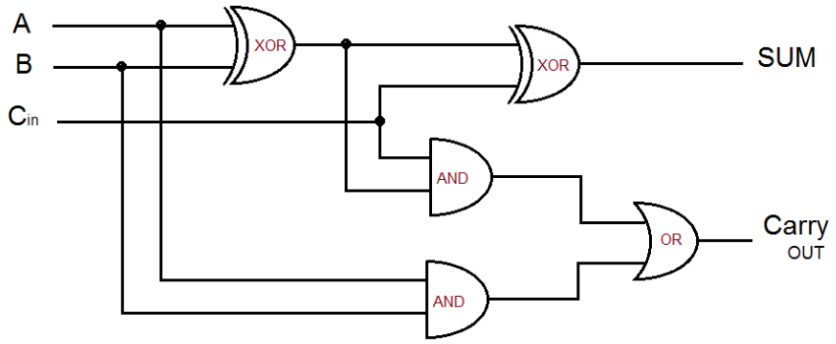
- Inputs: A, B, C=0.
- Outputs:P=A (unchanged), Q=A⊕B (partial sum), R=AB (partial carry).

The second **Peres Gate** combines the partial sum (A⊕B) with the carry-in (Cin):

- Inputs: Q=A⊕B, Cin, C=0.
- Outputs:P=A⊕B (unchanged), Q=(A⊕B)⊕Cin (final sum), R=(A⊕B)·Cin (carry generated in the second step).

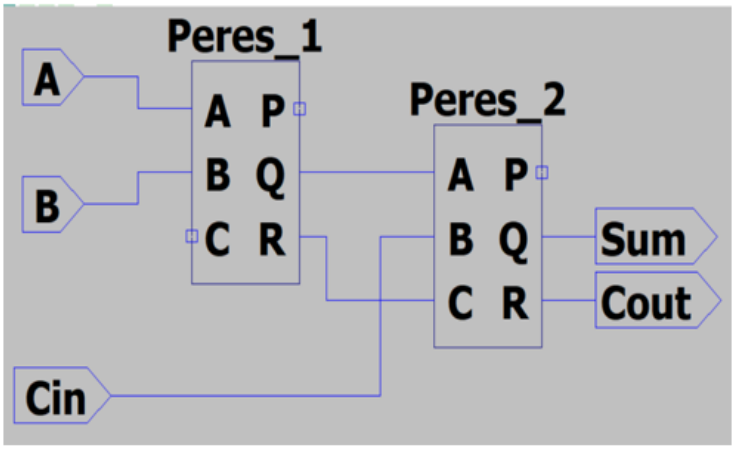
The final carry-out (Cout) is the OR-operation of the two partial carries:

- Cout=R1+R2, where R1 is the output from the first Peres Gate and R2 is the carry generated from the second Peres Gate.



The estimated power dissipation in our designed circuit is approximately 3 micro-watts.

Fig 22. Full Adder Schematic



- Inputs to the Feynman Gate: A,B.
- Outputs:P=A (unchanged), Q=A⊕B (Difference output).

Fig 23. Implementation of Full Adder Circuit

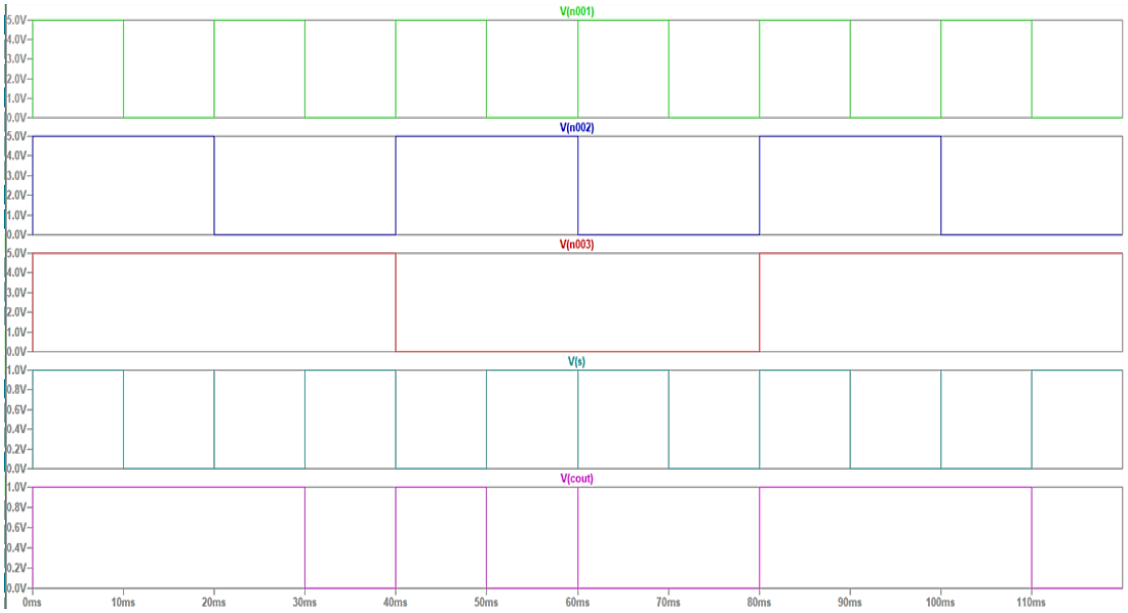


Fig 24. Output Simulation Plot for Full Adder

5.2 Half Subtractor: Efficiently designed using one **Feynman** and one **Fredkin** gate[5],[10],[11].

Step 1: Compute the Difference (D) using a Feynman Gate

Step 2: Compute the Borrow (Borrow) using a Fredkin Gate

- Inputs to the Fredkin Gate:C=A (control input), D=A`(complement of A), E=B (second input to be subtracted).
- Outputs:P=A (unchanged control input), Q=A`(unchanged, not used further), R=B if A=0, otherwise R=0(Borrow output).
- The Fredkin Gate ensures that Borrow=A`·B

The power dissipation of the designed circuit is estimated at about 4 micro-watts.

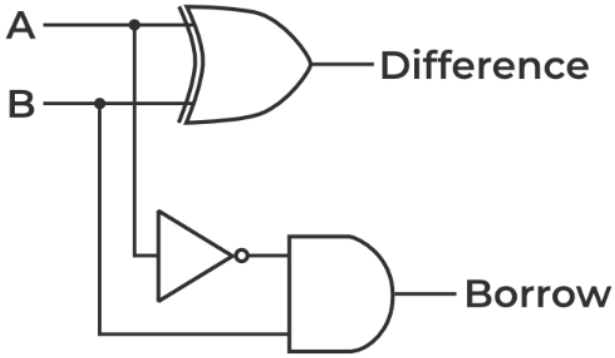


Fig 25. Half Subtractor Schematic

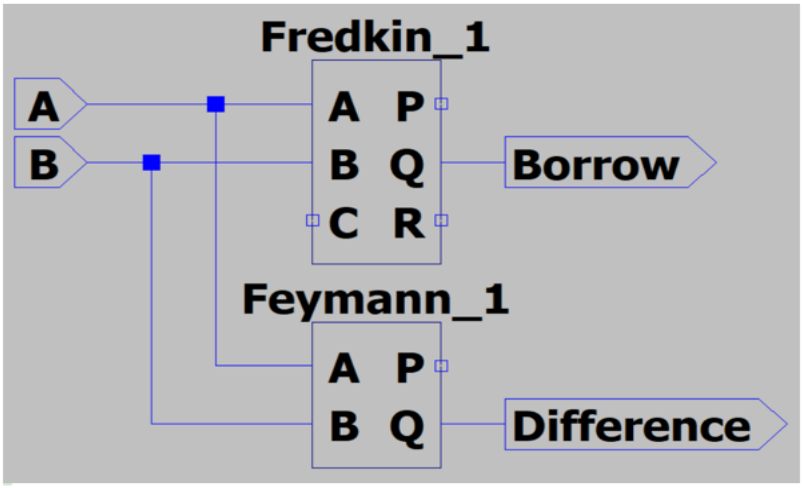


Fig 26. Implementation of Half Subtractor Circuit

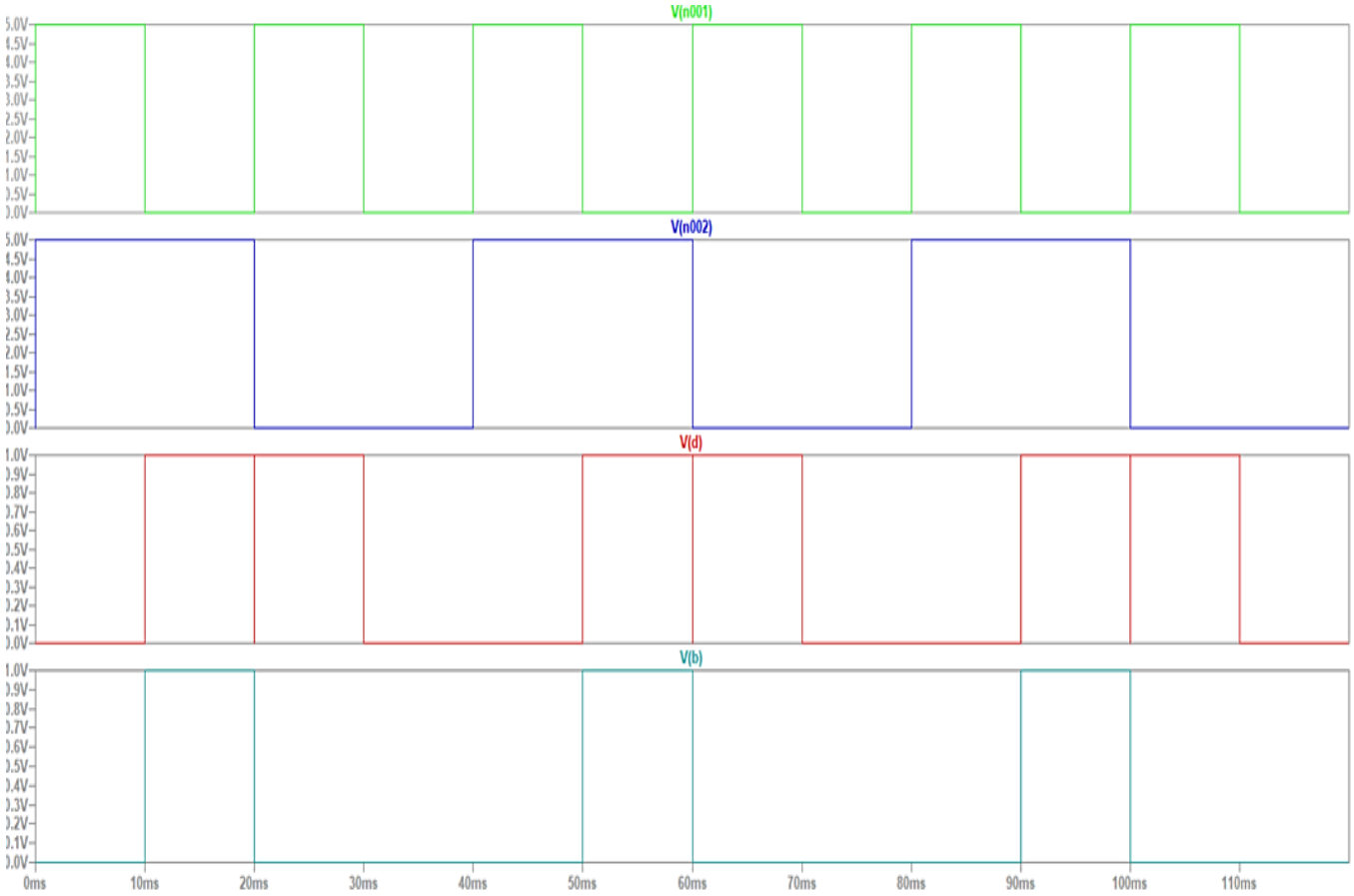


Fig 27. Output Simulation Plot for Half Subtractor

5.3 Full Subtractor: Designed using two Peres Gates and two Fredkin Gates[12].

Step 1: Calculate the Difference Using Peres Gates: To compute the difference (D), we need to perform the XOR of A, B, and Bin. This can be done in two steps using two Peres gates.

- **First Peres Gate**:

Inputs: A, B, and C=BinC = Bin

Outputs:P1=A, Q1=A⊕B, R1=A·B⊕Bin

- **Second Peres Gate** (to finalize the XOR operation):

Inputs: Q1=A⊕B, and C=Bin

Outputs:P2=A⊕B (unchanged), Q2=(A⊕B)⊕Bin (final difference, D), R2=(A⊕B)·Bin (carry-out for borrow calculation)

Step 2: Calculate the Borrow-out Using Fredkin Gates: The borrow-out depends on the conditions when a borrow occurs. Here we use two Fredkin gates to calculate the borrow-out.

- **First Fredkin Gate**:

Inputs: A, Band C=Bin

Outputs: P1=A (unchanged control input), Q1=B if A=0, otherwise Q1=Bin, R1=Bin if A=0, otherwise R1=B

- **Second Fredkin Gate** (final borrow-out calculation):

Inputs: P1=A, Q1=B, and R1=Bin

Outputs: P2=A (unchanged control input),Q2=B if A=0, otherwise Q2=Bin, R2=(A·B)⊕(A·Bin)⊕(B·Bin) (final borrow-out)

Estimated power dissipation is about 9.4 micro-watts for the designed circuit.

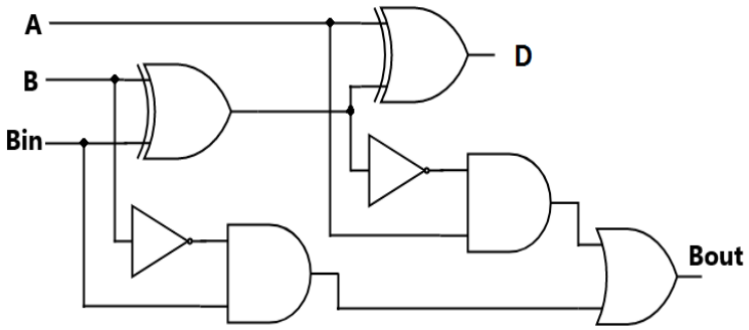


Fig 28. Full Subtractor Schematic

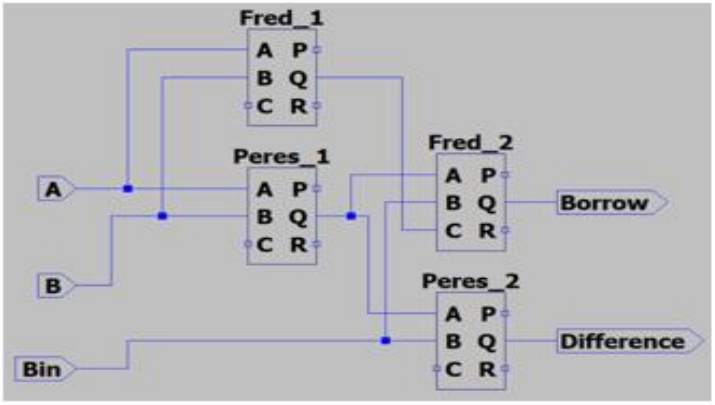


Fig 29. Implementation of Full Subtractor Circuit

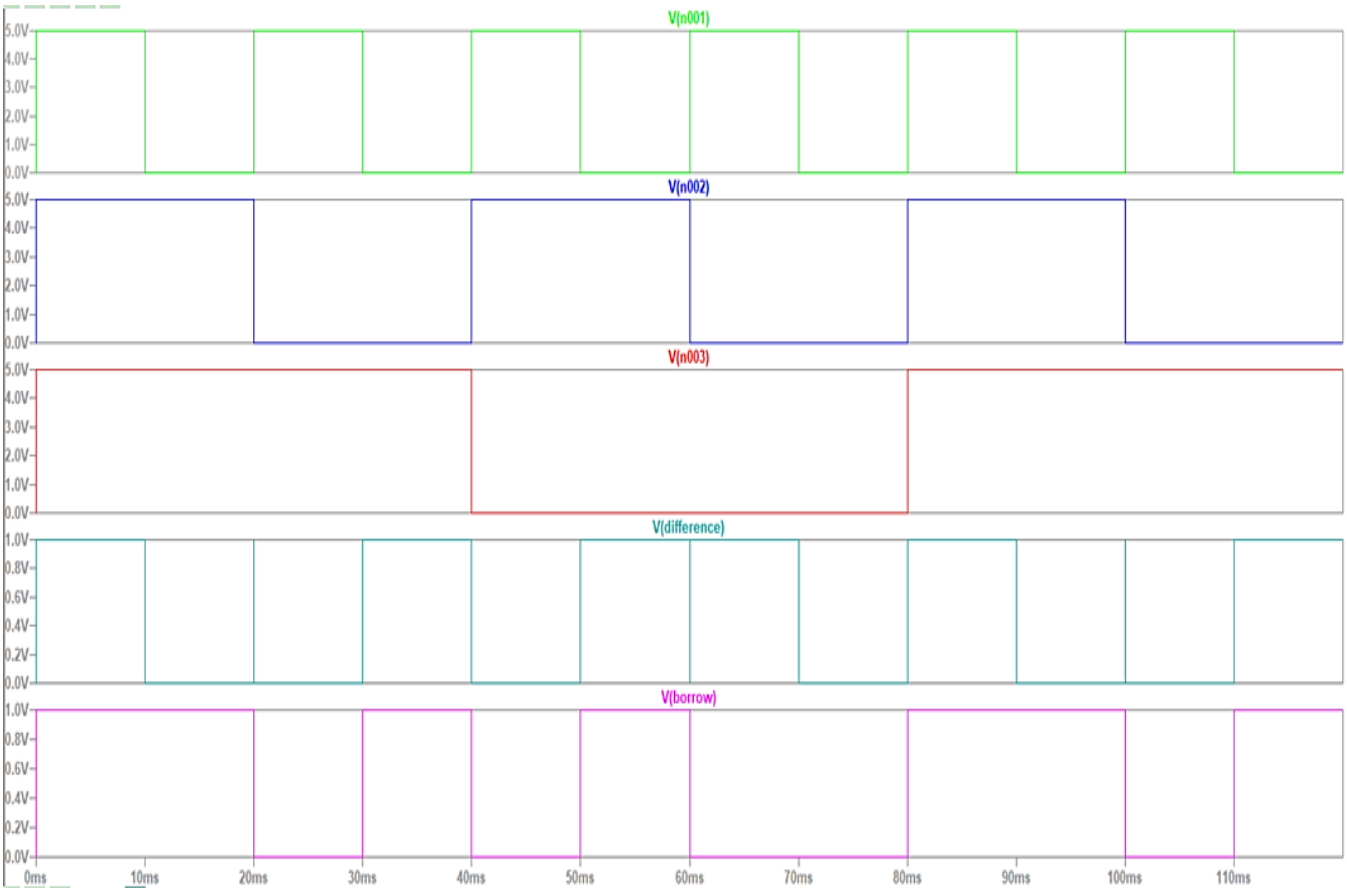


Fig 30. Output Simulation Plot for Full Subtractor

5.4 4:2 Decoder: Designed using one Peres and three Fredkin Gates[13],[14].

- Required logic:

$Y_0 = \bar{A}\bar{B}$

$Y_1 = \bar{A}B$

$Y_2 = A\bar{B}$

$Y_3 = AB$

Step 1:Using the Peres Gate: Inputs: A, B, 0

Outputs: P = A, Q = A $\oplus$ B, R = AB

Step 2: Fredkin Gate 1: Generating $\bar{A}$

Inputs: Control: Constant 1, B: A, C: 0

Output: One of the outputs give $\bar{A}$.

Step 3: Fredkin Gate 2: Generating $\overline{B}$

Inputs: Control: Constant 1, B: B, C: 0

Output: One of the outputs give $\overline{B}$

Step 4: Fredkin Gate 3: Generating Outputs Y_0, Y_1, Y_2

Inputs: Control: A or B, Using combinations such as ($\overline{A}$, $\overline{B}$), ($\overline{A}$, B), (A, $\overline{B}$)

Output: Q or R (depending on logic) gives:

- $Y_0 = \overline{A} \cdot \overline{B}$
- $Y_1 = \overline{A} \cdot B$
- $Y_2 = A \cdot \overline{B}$

$Y_3 = A \cdot B$ obtained from the Peres gate

Estimated power dissipation for the designed circuit is 12.3 micro-watts.

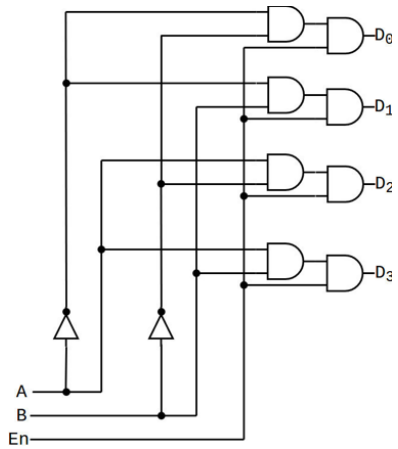


Fig 31. 2:4 Decoder Schematic

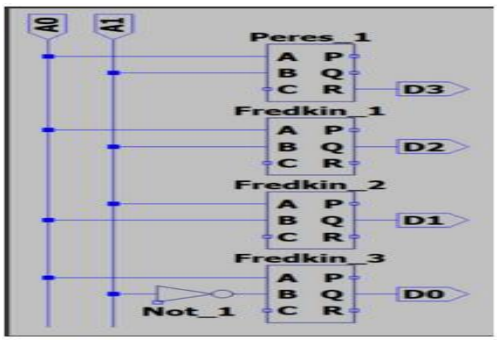


Fig 32. Implementation of 2:4 Decoder Circuit

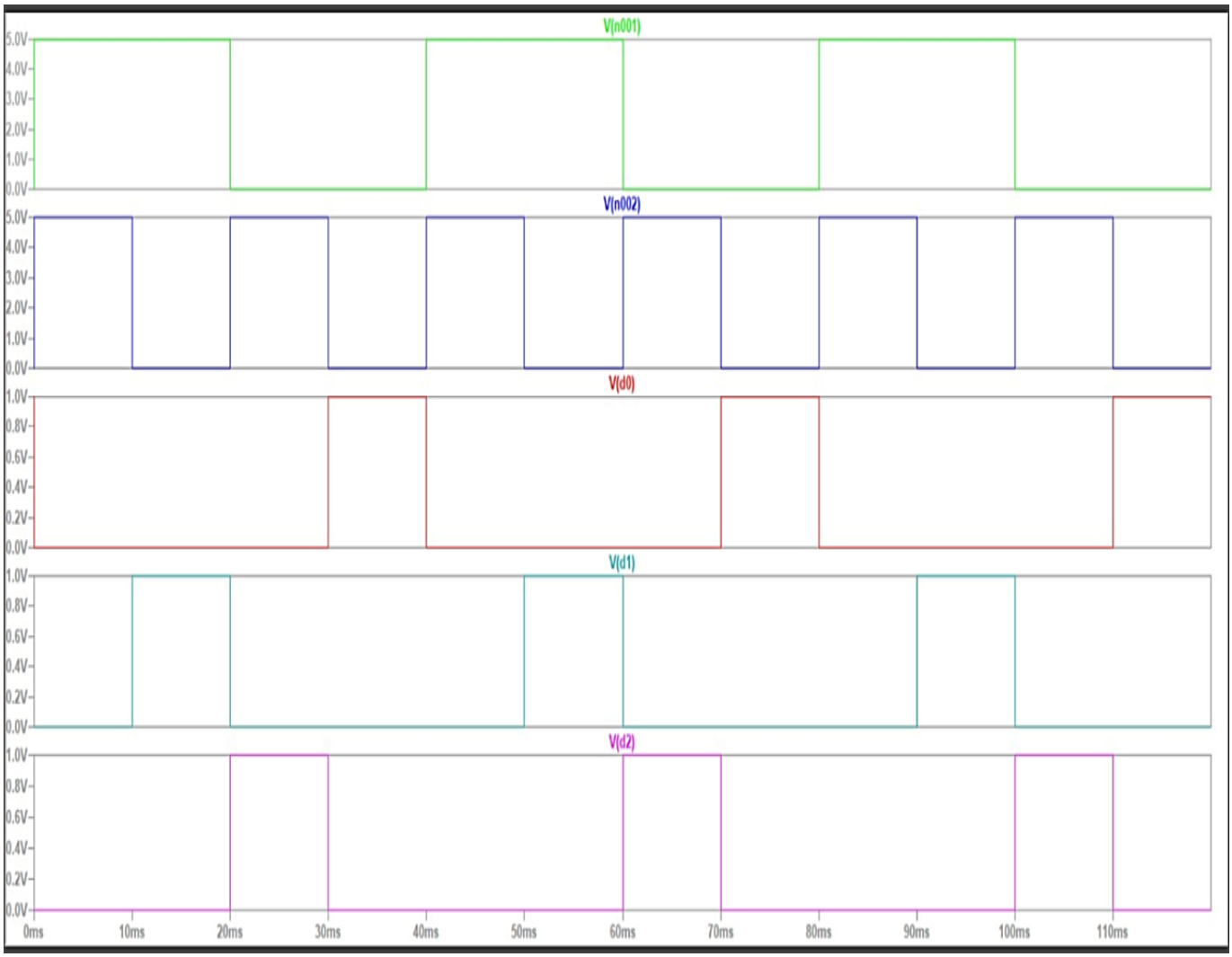


Fig 33. Output Simulation Plot for 2:4 Decoder

5.5 4:2 Encoder: Implemented using Universal Reversible Gates (URG)[15],[16].

URG Gate Logic:P = (A + B) ⊕ C

Q = B

R = AB ⊕ C

Step 1:Configuring URG1 for Y0 = D1 + D3

We choose: A = Y2, B = Y3, C = 0 (logical 0)

Then: P = (Y2 + Y3) ⊕ 0 = Y2 + Y3

A1 = P of URG1

Step 2: **Configuring URG2 for Y1 = D2 + D3**

We choose: A = Y3, B = Y1, C = 0

Then: P = (Y3 + Y1) ⊕ 0 = Y3 + Y1

A0= P of URG2

Step 3: **Final Output Connections**

Output of **URG1 → A1**

Output of **URG2 → A0**

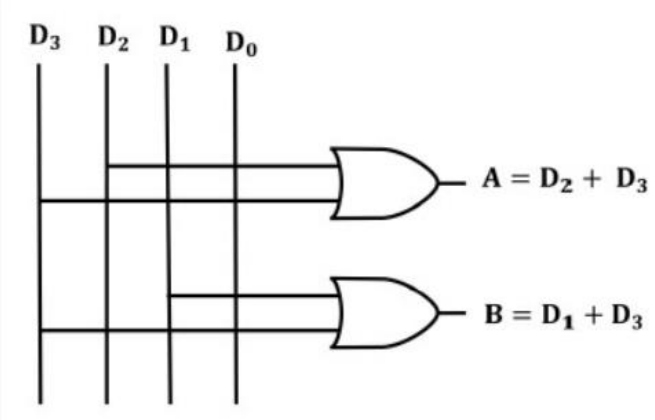
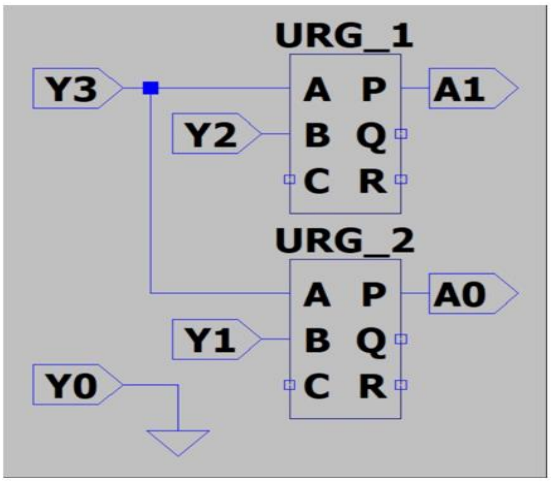


Fig 34. 4:2 Encoder Schematic

Estimated power dissiption of the designed circuit is 12 micro-watts.

Fig 35. Implementation of 4:2 Encoder Circuit



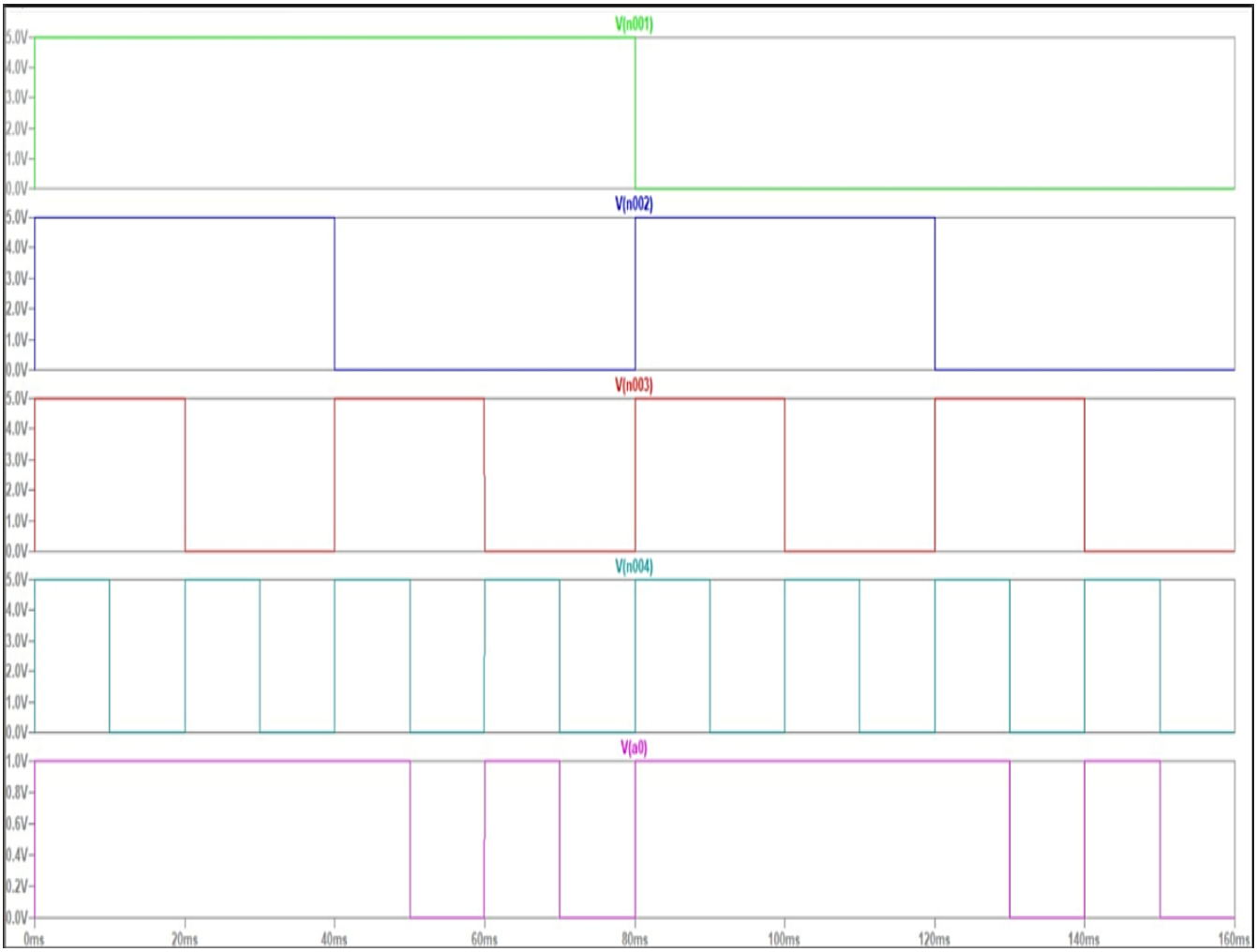


Fig 36. Output Simulation Plot for 4:2 Encoder

5.6 2:1 Multiplexer: Implementation is done using one **Modified Fredkin Gate (MFRG)**[17],[18],[19]

- Inputs: Using three inputs — A (input 0), B (input 1), and S (select line).
- Function of Gate: Use one Modified Fredkin Gate, which behaves as:

  If S = 0, output passes A.
  If S = 1, output passes B.

- MFRG Inputs: Input 1 → S (control)

  Input 2 → A

  Input 3 → B

- Outputs: The second output gives the MUX output:$Y = S' \cdot A + S \cdot B$

- Conclusion: A single MFRG is sufficient to implement a 2:1 multiplexer efficiently with reduced quantum cost.

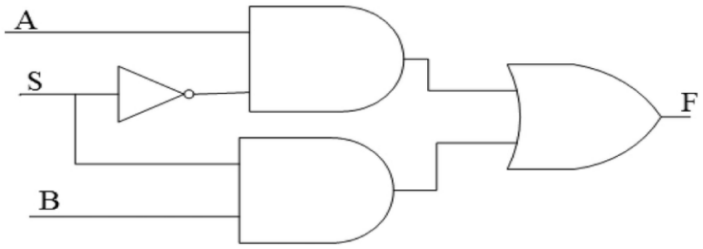


Fig 37. 2:1 Multiplexer Schematic

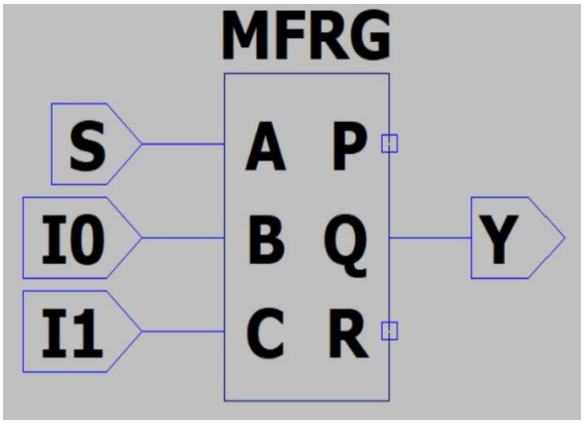


Fig 38. Implementation of 2:1 Multiplexer Circuit

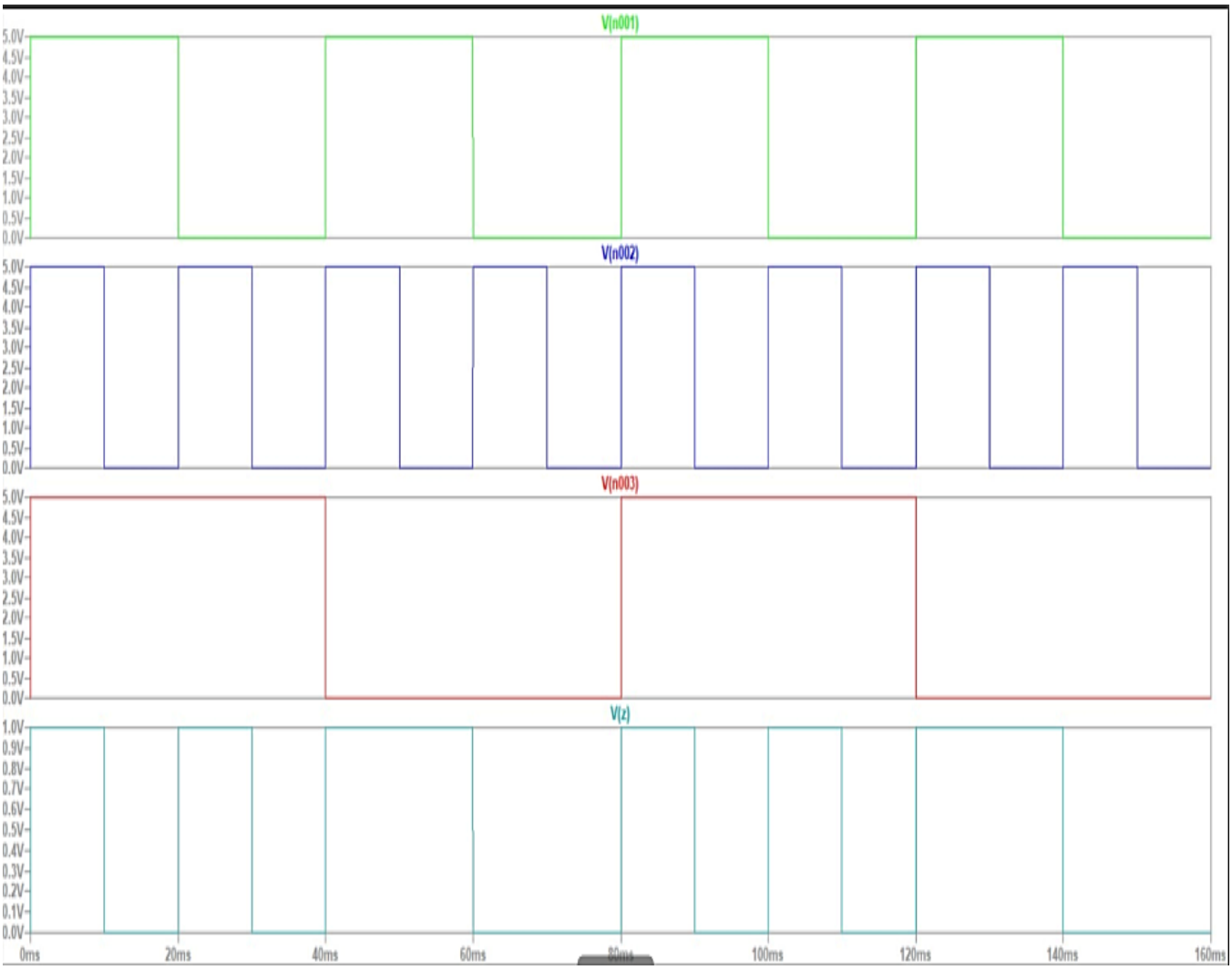


Fig 39. Output Simulation Plot for 2:1 Multiplexer

5.7 Even Parity Generator: Implemented using two Feynman Gates[20],[21].

Step 1:First Feynman Gate
Inputs: A, B
Outputs: P1 = A, Q1 = A ⊕ B

Step 2: Second Feynman Gate
Inputs: Q1 (A ⊕ B), C
Outputs: P2 = A ⊕ B, Q2 = (A ⊕ B) ⊕ C = A ⊕ B ⊕ C

Estimated power dissipation of the designed circuit is 3 micro-watts.

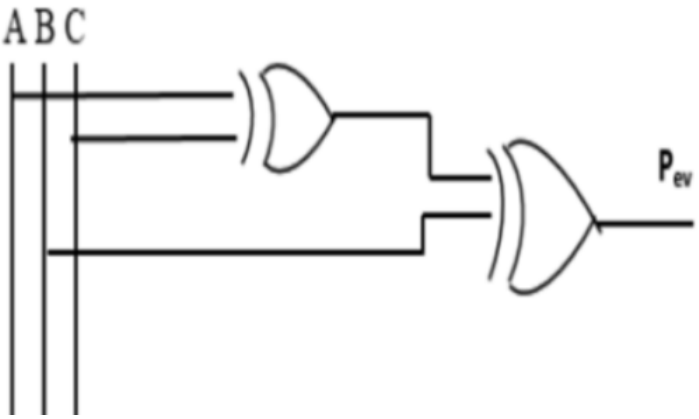


Fig 40. Even Parity Generator Schematic

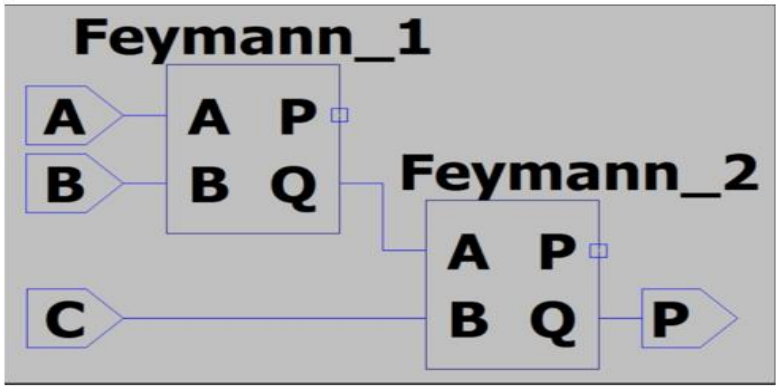


Fig 41. Implementation of Even Parity Generator Circuit

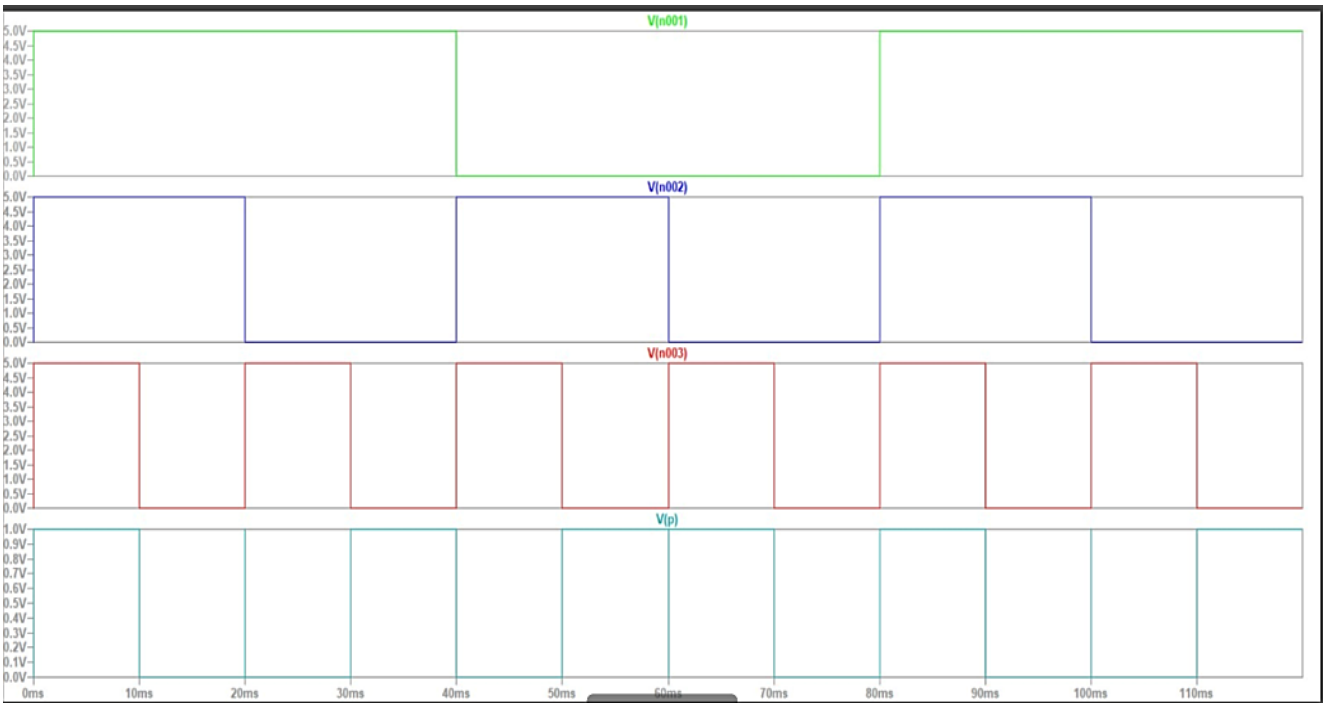


Fig 42. Output Simulation Plot for Even Parity Generator

5.8 Odd Parity Generator: We used two Feynman Gates[20],[21].

Step 1: First Feynman Gate

Inputs: A, B

Outputs: Q1 = A ⊕ B

Step 2: Second Feynman Gate

Inputs: Q1, C

Outputs: Q2 = A ⊕ B ⊕ C

Step 3: Finally Q2 is passed through a NOT gate to obtain the desired result (P)**.**

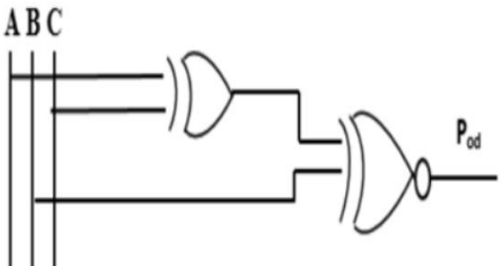


Fig 43. Odd Parity Generator Schematic

Estimated power dissipation for the designed circuit is 3.6 micro-watts.

1-bit Comparator: Circuit is designed using two Fredkin gates and one Feynman gate[22],[23],[24].

Step 1: First Fredkin Gate (Detect A > B)

Inputs: A, B′ , 0

Output represents **A > B**

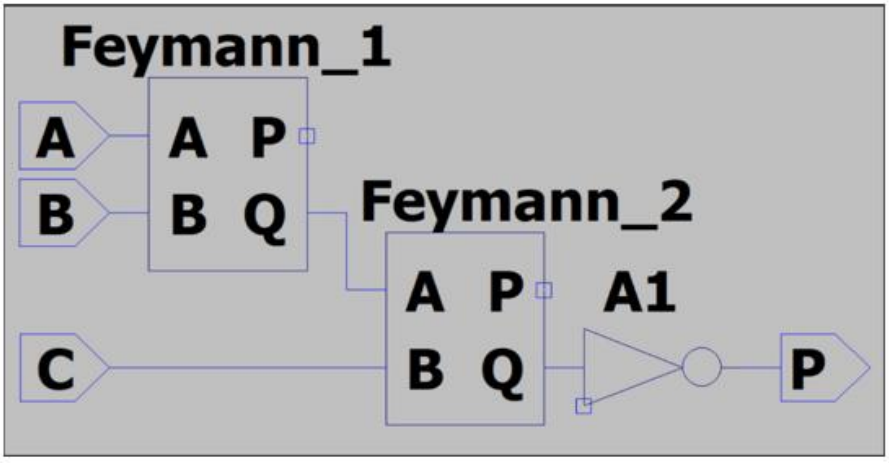


Fig 44. Implementation of Odd Parity Generator Circuit

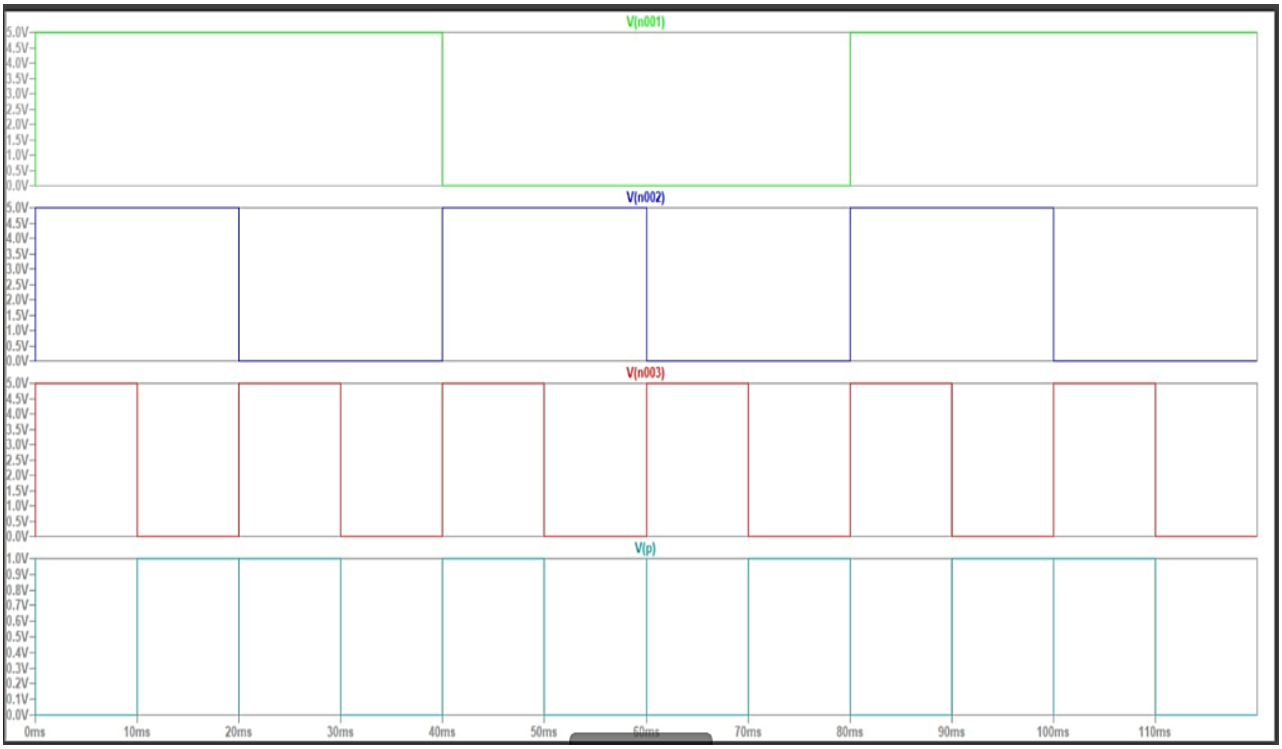


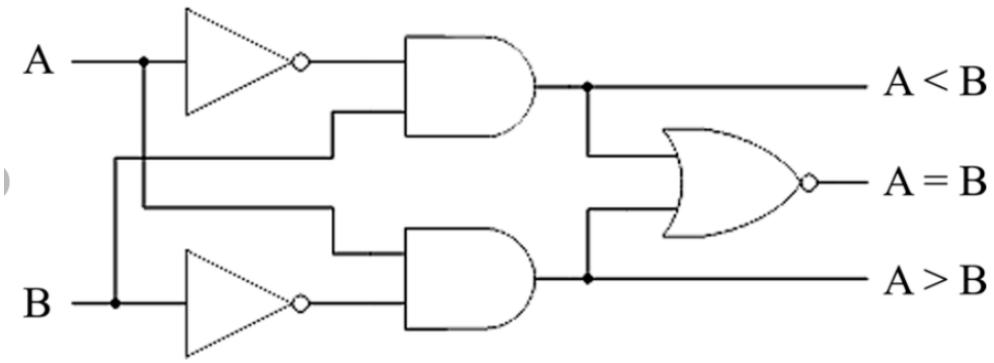
Fig 45. Output Simulation Plot for Odd Parity Generator

Step 2: Second Fredkin Gate (Detect A < B)
Inputs**:** B , A′ , 0
Output represents **A < B**

Step 3: Feynman Gate (A=B)

Inputs: A, B′
Output: A ⊕ B

This is passed through a not gate to obtain the desired result.



Estimated power dissipation for the designed circuit is 8.2 micro-watts.

Fig 46. 1-bit Comparator Schematic

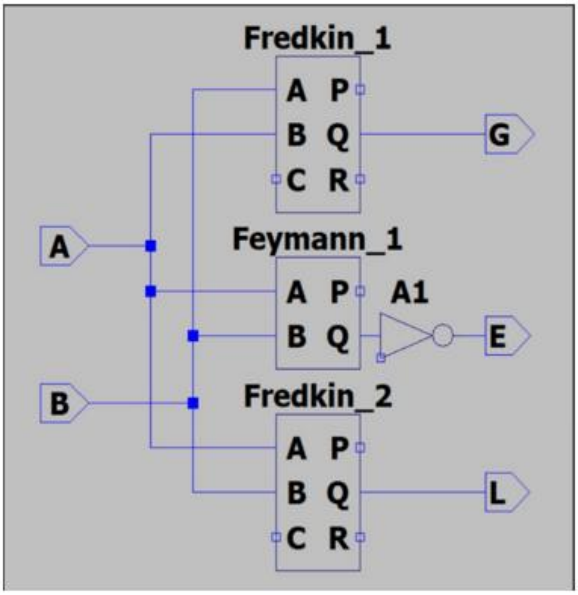


Fig 47. Implementation of 1-bit Comparator Circuit

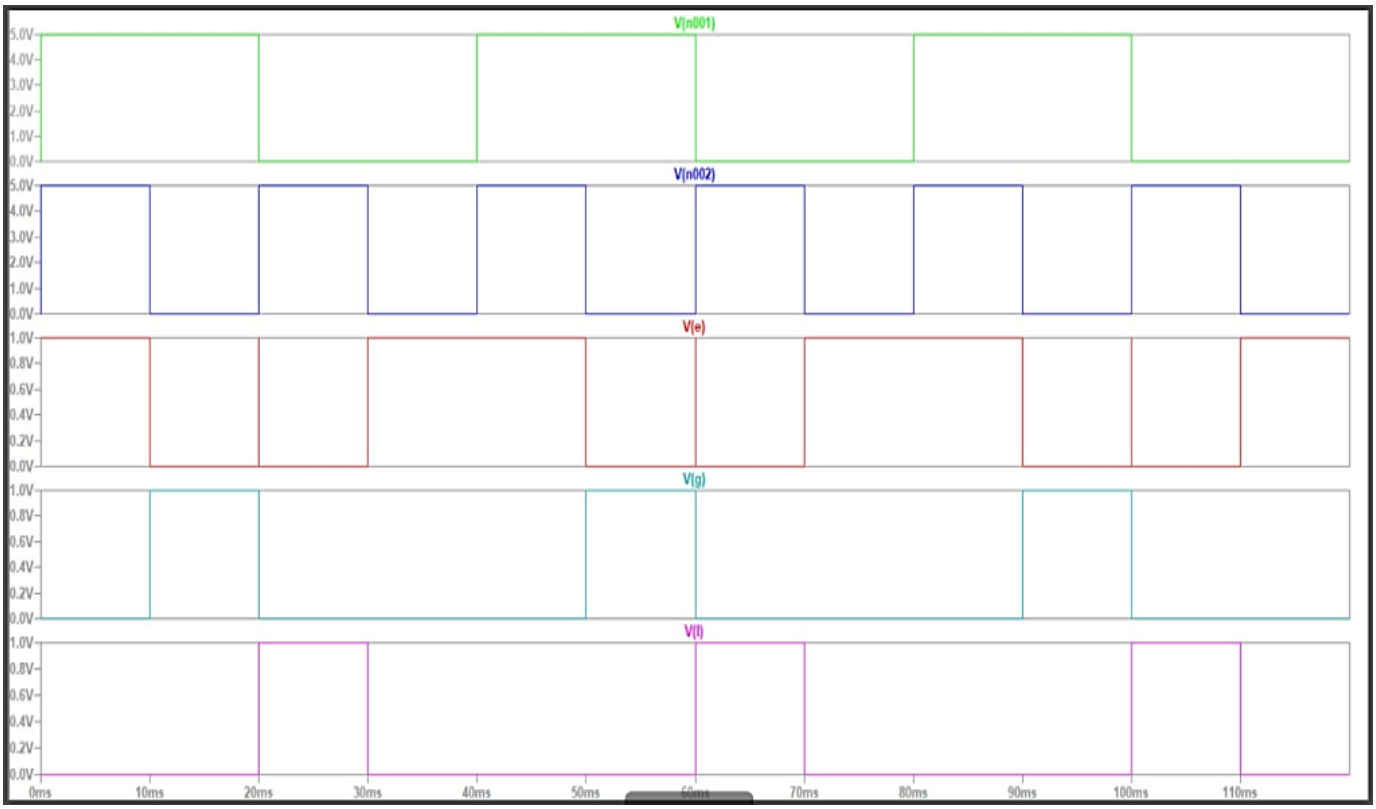


Fig 48. Output Simulation Plot for 1-bit Comparator

5.9 2-bit Binary Multiplier: Implemented using 4 Peres Gates[25],[26]**.**

Required Logic: P0= A0A1B0B1

P1= A0B0B1` + A0A1`B0 = A0B0 (A1`+B1`)= A0B0 . (A1B1)`

P2= A1B0B1` + A0A1`B1 + A0B0`B1 + A0`A1B0 = A1B0 (A0`+B1`) + A0B1(A1`+B0`) = A1B0 . (A0B1)` + A0B1 . (A1B0)`= A1B0 ⊕ A0B1

P3= A1B1

Step 1: We use Peres gates to generate the required bitwise AND operations:

Peres 1: Inputs = A0, B0, 0 → Output R = A0B0

Peres 2: Inputs = A0, B1, 0 → Output R = A0B1

Peres 3: Inputs = A1, B0, 0 → Output R = A1B0

Peres 4: Inputs = A1, B1, 0 → Output R = A1B1

Step 2: P0 = A0B0: R output of Peres 1

Step 3: P1 = A0B0 . (A1B1)': Using the R outputs of Peres 1 (A0B0) and Peres 4 (A1B1).

Now, feeding them into a Fredkin gate to implement the logic:

If A1B1 = 0, output P1 = A0B0

If A1B1 = 1, output P1 = 0

Step 4: P2 = A1B0 ⊕ A0B1: Using Feynman gate:Inputs = A1B0 (from Peres 3), A0B1 (from Peres 2); Output Q = A1B0 ⊕ A0B1

Step 5: P3 = A1B1: R output of Peres 4

Estimated power dissipation of the designed circuit is 11.9 micro-watts.

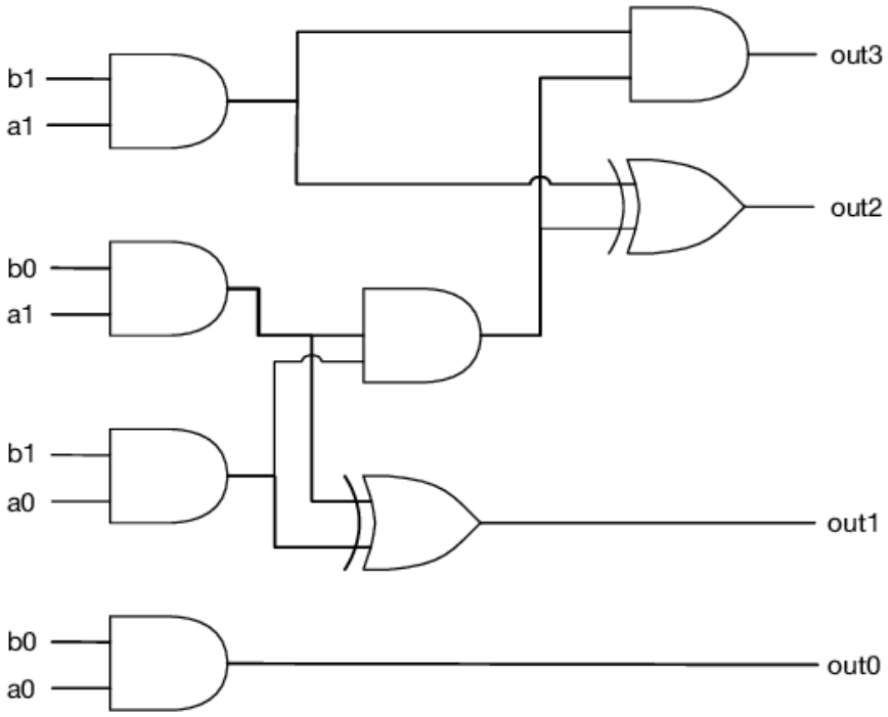


Fig 49. 2-bit Binary Multiplier Schematic

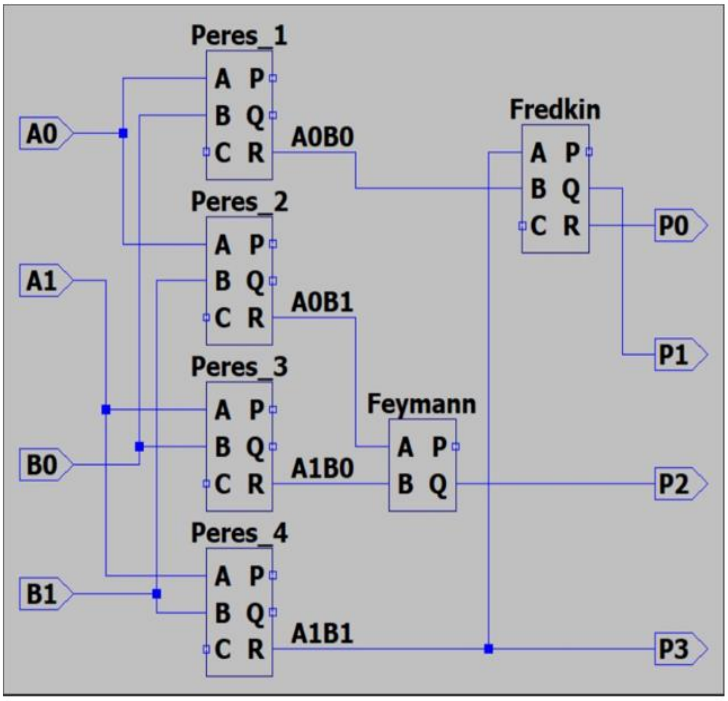


Fig 50. Implementation of 2-bit Binary Multiplier Circuit

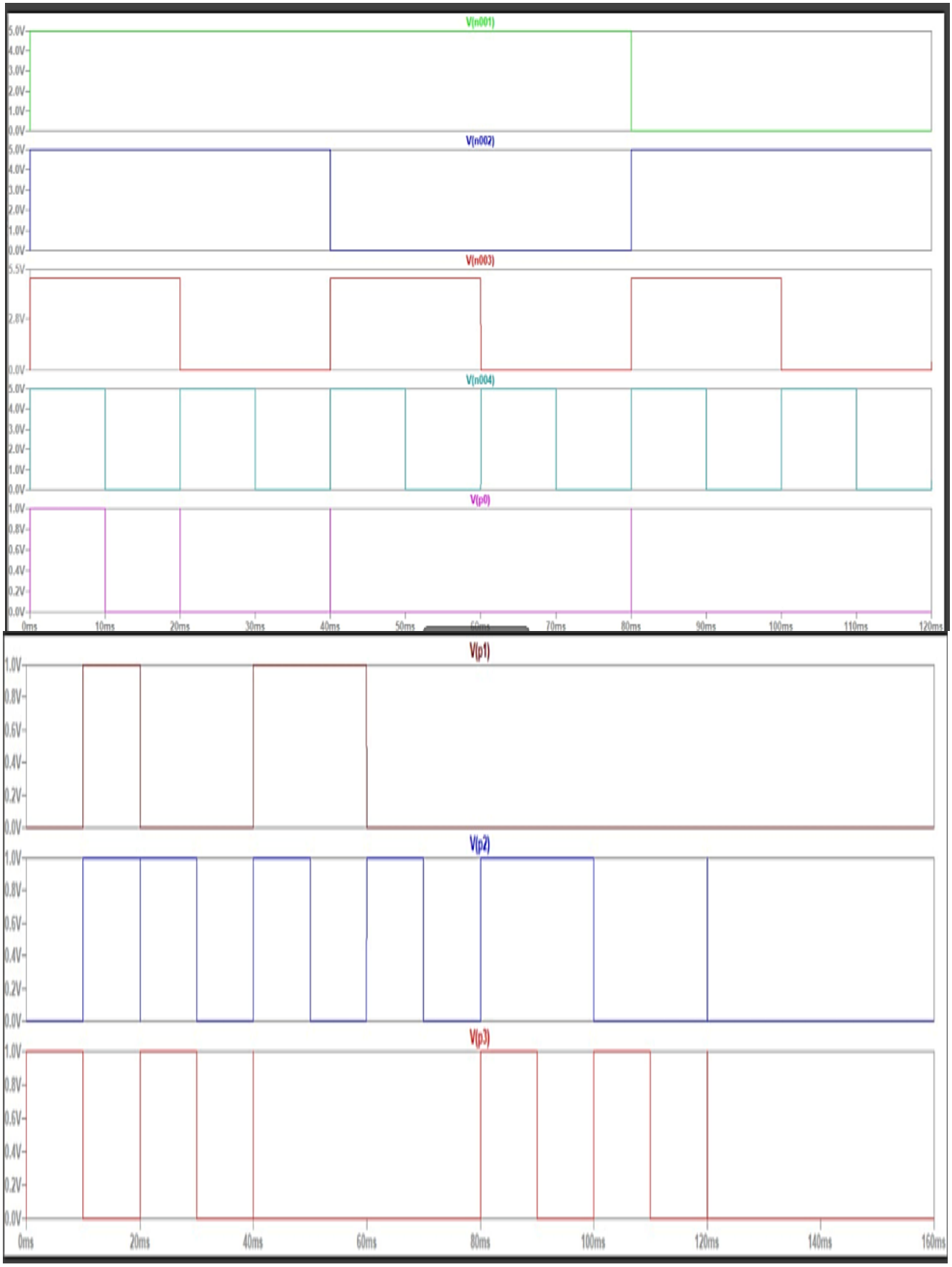


Fig 51. Output Simulation Plot for 2-bit Binary Multiplier

6. Comparison

- **Half Adder:** Comparing our design with the existing circuits [5] and [6].

PARAMETERS/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUT
Our Circuit (1 Peres Gate)	1	4	1
Existing Circuit [5]	2	10	6
Existing Circuit [6]	1	10	2

Table 1. Comparison Table for Half Adder

- **Full Adder:** Comparing our design with the existing circuits [7], [8] and [9]

PARAMETERS/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUT
Our Design (2 Peres Gates)	2	8	2
Existing Circuit [7]	1	10	2
Existing Circuit [8]	1	6	2
Existing Circuit [9]	1	11	2

Table 2. Comparison Table for Full Adder

- **Half Subtractor:** Comparing our design with the existing circuits [5], [10] and [11]

PARAMETERS/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUT
Our Design(1 Fredkin, 1Feynman)	2	6	3
Existing Circuit[10]	2	10	3
Existing Circuit[11]	1	7	1
Existing Circuit[5]	2	10	6

Table 3. Comparison Table for Half Subtractor

- **Full Subtractor:** Comparing our design with the existing circuit [12]

PARAMETERS/ CIRCUITS	NO OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUTS
Our Design (2 Fredkin, 2 Peres)	4	18	8
Existing Circuit [12]	3	14	4

Table 4. Comparison Table for Full Subtractor

- **2:4 Decoder:** Comparing our design with the existing circuits [13] and [14]

PARAMETERS/ CIRCUITS	NO OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUT
Our Design	4	19	8
Existing Design [13]	6	22	10
Existing Design [14]	17	67	17

Table 5. Comparison Table for 2:4 Decoder

PARAMETERS/ CIRCUITS	NO OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUTS
Our Design (2 URG)	2	8	4
Existing Circuit [15]	3	12	7
Existing Circuit [16]	3	15	7

- **4:2 Encoder:** Comparing our design with the existing circuits [15] and [16]

Table 6. Comparison Table for 4:2 Encoder

- **2:1 Multiplexer:** Comparing our design with the existing circuits [17], [18], and [19]

PARAMETERS/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUT
Our Circuit (1 MFRG)	1	4	5
Existing Circuit [17]	1	5	3
Existing Circuit [18]	2	10	6
Existing Circuit [19]	2	14	5

Table 7. Comparison Table for 2:1 Multiplexer

- **Even Parity Generator:** Comparing our design with the existing circuits [20] and [21]

PARAMETERS/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUTS
Our Design (2 Feynman)	2	2	2
Existing Circuit [20]	1	7	3
Existing Circuit [21]	16	16	16

Table 8. Comparison Table for Even Parity Generator

- Odd Parity Generator:Comparing our design with the existing circuits [20] and [21]

PARAMETERS/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUT
Our Design (2 Feynman, 1 NOT)	3	3	2
Existing Circuit [20]	2	8	3
Existing Circuit [21]	24	24	16

Table 9. Comparison Table for Odd Parity Generator

- **1 Bit Comparator:** Comparing our design with the existing circuits [22], [23] and [24]

PARAMETER/ CIRCUITS	NO. OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUTS
Our Design (1 Feynman, 2 Fredkin)	3	11	5
Existing Circuit [22]	2	14	4
Existing Circuit [23]	3	15	5
Existing Circuit [24]	3	16	7

Table 10. Comparison Table for 1 Bit Comparator

- **2 Bit Binary Multiplier:** Comparing our design with the existing circuits [25] and [26]

PARAMETERS/ CIRCUITS	NO OF REVERSIBLE GATES USED	TOTAL QUANTUM COST	TOTAL GARBAGE OUTPUTS
Our Design (4 Peres, 1 Fredkin, 1 Feynman)	6	22	10
Existing Circuit [25]	9	36	5
Existing Circuit [26]	6	28	12

Table 11. Comparison Table for 2 Bit Binary Multiplier

7. Conclusion

Basically,reversible circuits do not lose information and conserves information. Reversible computation is performed only when system comprises of reversible gates. The reversible logic is designed, mainly for the purposes of reduced quantum cost, depth of the circuits and the number of garbage outputs[1],[2],[3],[4]. We have successfully implemented some common combinational circuits in a reversible approach and also tried to optimize the total quantum cost and the garbage outputs. In reality these circuits help to cut the power dissipation as there is no information loss in the process. Advancements will pave the way for more innovative developments in quantum computing, nanotechnology, and low-power design in the digital world.

8. Acknowledgement

We would like to thank the department of Electronics and Communication Engineering of Heritage Institute of Technology for providing all the resources to support the work and their unwavering support and dedication to our growth.

References

1. R. Landauer, “Irreversibility and Heat Generation in the Computational Process”, IBM Journal of Research and Development.
2. M. Khan, "Reversible Realization of Quaternary Decoder, Multiplexer, and Demultiplexer Circuits," in Multiple Valued Logic, 2008. ISMVL 2008. 38th International Symposium on, 2008.
3. C. H. Bennett, "Logical reversibility of Computation", IBM J. Research and Development, pp. 525-532, November 1973.
4. E. Fredkin and T. Toffoli, “Conservative Logic", Int'l J. Theoretical Physics Vol 21, pp. 219-253, 1982.
5. Miller, D. & Wille, Robert & Dueck, G.W. (2009). Synthesizing Reversible Circuits for Irreversible Functions. 12th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, DSD 2009. 749 - 756.
6. Lukas Burgholzer, Robert Wille, Richard Kueng,Characteristics of reversible circuits for error detection,Array,Volume 14,2022,100165,ISSN 2590-0056
7. Aakash Gupta, Pradeep Singla, Jitendra Gupta, Nitin Maheshwari, An Improved Structure Of Reversible Adder And Subtractor, International Journal of Electronics and Computer Science Engineering, Vol 2, No. 2, pp712-718, June 2013
8. Neeraj Kumar Misra, Mukesh Kumar Kushwaha, Subodh Wairya, Amit Kumar, Cost Efficient Design of Reversible Adder Circuits for Low Power Applications
9. Gandhi, S.M., Devishree, J., Sathish Mohan, S. (2014). A New Reversible SMG Gate and Its Application for Designing Two’s Complement Adder/Subtractor with Overflow Detection Logic for Quantum Computer-Based Systems. In: Krishnan, G., Anitha, R., Lekshmi, R., Kumar, M., Bonato, A., Graña, M. (eds) Computational Intelligence, Cyber Security and Computational Models. Advances in Intelligent Systems and Computing, vol 246. Springer, New Delhi.
10. Shekoofeh Moghimi, Mohammad R. Reshadinezhad, "A Novel 4×4 Universal Reversible Gate as a Cost-Efficient Full Adder/Subtractor in Terms of Reversible and Quantum Metrics", International Journal of Modern Education and Computer Science (IJMECS), vol.7, no.11, pp.28-34, 2015.
11. Arwa S. Bazmalah1, Noorfazila Kamal, Design of Reversible Binary Half and Full Adder/Subtractor Using Reversible Gates, Progress in Engineering Application and Technology Vol. 3 No. 2 (2022) 988-997
12. Nayana D. K., Sujatha B. K., Efficient Design of Half Adder and Half Subtractor Using New SN Reversible Gate, International Journal of Electronics Communication and Computer Engineering Volume 4, Issue 2, ISSN (Online): 2249–071X, ISSN (Print): 2278–4209
13. G. Nikhil, B. Sharanya, P. B. Reddy and R. P. Vidyadhar, "Reversible 2:4 Decoder Using Universal Fredkin Gate," 2021 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 173-176.
14. C. Kalamani, R. Murugasami, S. Usha, S. Saravanakumar,Design of encoder and decoder using reversible logic gates,Volume 31, 2024,100989,ISSN 2665-9174
15. Morrison M, Ranganathan N. Design of a reversible ALU based on novel programmable reversible logic gate structures. In VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on 2011 (pp.126-31).

16. Hatkar AP, Hatkar AA, Narkhede NP. ASIC design of reversible multiplier circuit. In Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on 2014 (pp. 47-52).
17. A P Sooriamala1, V Sherlin Solomi, Reeba Korah, Aby K Thomas, Design of Multiplexers using Reversible Logic Technique, International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 11 Issue: 9s
18. Gopal, N. Raj, A. A. Gopalai and A. K. Singh, "Design of reversible multiplexer/de-multiplexer," 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014), Penang, Malaysia, 2014, pp. 416-420.
19. Praveen.B1, Vinay Kumar S.B2, 2n:1 Reversible Multiplexer and its Transistor Implementation, International Journal of Computational Engineering Research / ISSN: 2250–3005
20. Vinay Kumar, Divya Dhawan, Design of Reversible Even and Odd Parity Generator and Checker Using Multifunctional Reversible Logic Gate (MRLG), International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 4 Issue: 6 85 – 91
21. A. Anjana, Even and Odd Parity Generator and Checker using the Reversible logic gates, International Journal of Computer Science and Engineering Communications-IJCSEC. Vol.1 Issue.1, December 2013
22. Akshata S, Arpitha E, Deepashree V A, Pushpa Lekha Nag D, 2018, Low Power Comparator Design Using Reversible Logic Gates – Adiabatic Circuits, International Journal of Engineering RESEARCH & Technology (IJERT) NCESC – 2018 (Volume 6 – Issue 13)
23. Saxena, Arti. (2013). Optimized Study of One-Bit Comparator Using Reversible Logic Gates. 2. 2321-7308.
24. Pratik Kumar Bhatt', Arti Saxena², Optimized Study of One-bit Comparator Using Reversible Logic Gates, IJRET: International Journal of Research in Engineering and Technology, ISSN: 2319-1163
25. Kadbe, P.K.; Markande, S.D. Efficient Design of Reversible Adder and Multiplier Using Peres Gates. Appl. Sci. 2024, 14, 9385.
26. Himanshu Thapliyal, M.B Srinivas, Novel Reversible Multiplier Architecture Using Reversible TSG Gate, Published in Proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06), Dubai, March 2006. pp. 100-103