CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In modern VLSI system power dissipation is very high due to rapid switching of internal signals. The complexity of VLSI circuits increases with miniaturization of integrated circuits every year due to packing more and more logic elements into smaller volumes. Hence power dissipation has become the main area of concern in VLSI design.The reduction of power dissipation has become a crucial issue in today's hardware design process. Traditional irreversible circuits generate heat due to the loss of information during computation.

According to scientist R Landauer's research in 1961, the amount of energy dissipated for every irreversible bit operation in a system is at least KTln2 joules, where$K = 1.3806*10^{-23}m^2kg^{-2}K^{-1}$(joule/kelvin^{-1}) is the Boltzmann's constant and T is the temperature at which operation is performed [1]. The heat generated due to the loss of one bit of information is very small at room temperature but when the number of bits is more as in the case of high speed computational works the heat dissipated by them will be so large that it affects the performance and results in the reduction of lifetime of the components. In 1973C.H.Bennettshowedthat KTln2energy would not dissipate from a system as long as the system allows the reproduction of the inputs from observed outputs [2]. He inferred that amount of energy dissipation would not occur if a computation is carried out in a reversible way using reversible logic[3].

Reversible logic supports the process of running the system both forward and backward. This means that reversible computations can generate inputs from outputs and can stop and go back to any point in the computation history. A circuit is said to be reversible if the input vector can be uniquely recovered from the output vector and there is a one-to-one correspondence between its input and output assignments i.e. not only the outputs can be uniquely determined from the inputs, but also the inputs can be recovered from the outputs Energy dissipation can be reduced or even eliminated if

computation becomes Information lossless[4].Thus reversible logic appears to be promising in future low power design applications.

Reversible circuits are an important class of computations that needs to be performed efficiently for the purpose of efficient quantum computation. Indeed, multiple quantum algorithms contain arithmetic units (e.g., adders, multipliers, exponentiation, comparators, quantum register shifts, and permutations) that are best viewed as reversible circuits; reversible circuits are indispensable for quantum error correction. Often, efficiency of the reversible implementation is the bottleneck of a quantum algorithm (e.g., integer factoring and discrete logarithm) or a class of quantum circuits.

1.2 PROJECT BACKGROUND

Reversible logic is gaining importance in VLSI design because of its low power dissipation. The traditional gates like AND, OR, XOR are all irreversible gates. Consider the case of traditional AND gate. It consists of two inputs and one output. As a result, one bit is lost each time a computation is carried out. According to the truth table of AND, there are three inputs (1, 0), (0, 1) and (0, 0) that corresponds to an output zero. Hence it is not possible to determine a unique input that resulted in the output zero. In order to make a gate reversible additional input and output lines are added so that a one to one mapping exists between the input and output[5]. This prevents the loss of information that is main cause of power dissipation in irreversible circuits.

Reversible circuits are similar to conventional logic circuits except that they are built from reversible logic gates. In reversible gates, there is unique i.e. one-to-one mapping between the inputs and outputs, which is not the case in conventional logic. Reversible gates are used in quantum computing system as quantum operations are reversible in nature.

To reduce the power consumption different logic design techniques like CMOS complementary logic, Pseudo NMOS, dynamic CMOS, clocked CMOS logic (C2MOS), CMOS domino logic, cascade voltage switch logic (CVSL), modified domino logic, pass transistor logic (PTL) have been proposed[6] . Although static

CMOS logic has been the most popular design approach for the past three decades, many attempts have been made to propose a better alternative to achieve lower power dissipation, smaller area and better performance reported.

Circuit designed with transmission gate (TG) solves the problem of low logic level swing by using PMOS as well as NMOS but this implementation needs true and complemented control signal and requires more area than pass transistor logic. Pseudo NMOS is simple and fast but reduces noise margins and increases power consumption. Pass transistor logic is good for certain classes of circuits (MUX/adders) [7]. On the other hand, PTL implementations of logic gates such as NANDs and NORs were found to be slower and consume more power than CMOS implementations mainly because of the reduced output swings due to the threshold drop across a single-channel pass transistor [8].

Transistor count is a primary concern which largely affects the design complexity of larger circuit. For such submicron CMOS technology area, topology selection, power dissipation and speed are very important aspect for highspeed and low power application. These issues can be overcome by incorporating gated diffusion input (GDI) technique. Gate diffusion input (GDI) is a lowest power design technique which offers improved logic swing and less static power dissipation. Using this technique several logic functions can be implemented using less number of transistor counts. This method is suitable for design of fast, low-power circuits, using a reduced number of transistors (as compared to TG and CMOS).

The Gate Diffusion Input (GDI) is a lowest power design technique which offers improved logic swing and less static power dissipation. Using this technique several logic functions can be implemented using less number of transistor counts. This method is suitable for design of fast, low-power circuits, using a reduced number of transistors (as compared to TG and CMOS). Though GDI technique offers low power, less transistor count and high speed, the major challenges occurs in the fabrication process. The GDI technique requires twin-well CMOS or Silicon on Insulator (SOI) process to realize a chip which increases the complexity as well as the cost of fabrication.

1.3 OBJECTIVE

The main objective of this project is to design and analyse the 8 bit reversible Processor using reversible logic gates with gate diffusion input (GDI) logic design technique which perform arithmetic and logical operations.

1.4 ORGANISATION OF THE THESIS

There are five chapters in the thesis. These are organized as follows:

- Chapter 1 Deals the requirement of low power circuits which is explained along with the overview of its design techniques.
- Chapter 2 Deals with a detailed discussion on Reversible Logic and basic Reversible Logic Gates and Gate Diffusion Input (GDI) techniques are included.
- Chapter 3 Deals with a design Architecture of various Reversible Logic Circuits with Logical Block Diagram.
- Chapter 4 Deals with the simulated Results with output waveform and output parameter of Reversible Logic Circuits and Processor.
- Chapter 5 Deals with the Conclusion and contribution drawn from this work with suggestions for future work are discussed.

CHAPTER 2

LITERATURE SURVEY

2.1 REVERSIBLE LOGIC GATE

A reversible logic gate is an n-input n-output logic device with one-to-one mapping. This helps to determine the outputs from the inputs and also the inputs can be uniquely recovered from the outputs. Also in the synthesis of reversible circuits direct fan-out is not allowed as one to many concepts is not reversible[9]. However fan-out in reversible circuits is achieved using additional gates. A reversible circuit should be designed using minimum number of reversible logic gates. From the point of view of reversible circuit design, there are many parameters for determining the complexity and performance of circuits [10].

The reversible circuit/gate has the following characteristics:

(i) Has equal number of inputs and outputs.
(ii) The gate output which is not used as primary output in the circuit is called garbage output.
(iii) The input which is used as control input to the gates is called constant/garbage input.
(iv) The fan-out of each gate is equal to one. A copying circuit is used if two copies of a signal are required.

An efficient design in reversible logic should have the following features:

(i) Use minimum number of reversible logic gates.
(ii) Should have less number of garbage outputs.
(iii) Less number of constant inputs.
(iv) Minimization of quantum cost.

2.2 BASIC REVERSIBLE LOGIC GATES

2.2.1 Feynman Gate

Feynman gate is a 2x2 reversible gate as shown in figure 2.1. The input vector is I (A, B) and the output vector is O (P, Q). The outputs are defined by P=A, Q=A ⊕B. Quantum cost of a Feynman gate is 1. Truth table of Feynman gate shown inTable 2.1.Feynman gate (FG) can be used as a copying gate. Since a fan-out is not allowed in reversible logic, this gate is useful for duplication of the required outputs. Feynman gate is also known as Controlled NOT (CNOT) gate.

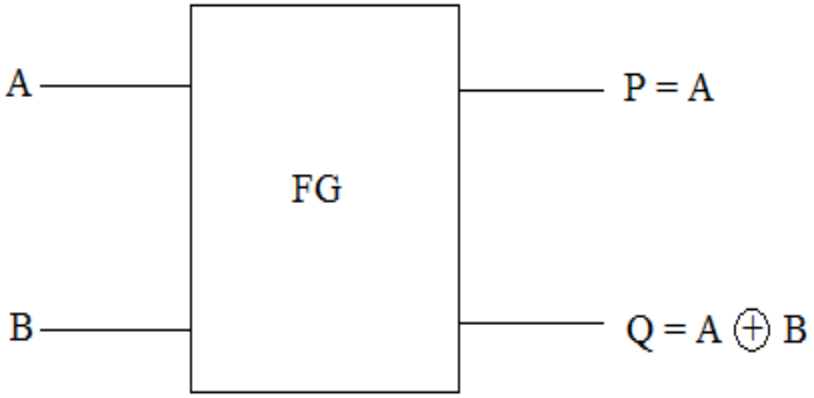


Figure 2.1 Feynman gate

Table 2.1 Truth table of Feynman gate

INPUT		**OUTPUT**	
A	**B**	**P**	**Q**
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

2.2.2 Toffoli Gate:

3x3 Toffoli gate(TG) shown in Figure 2.2. The input vector is I (A, B, C) and the output vector is O(P,Q,R). Truth table of Toffoli gate shown in Table 2.2. The outputs are defined by P=A, Q=B, R=AB⊕C.Toffoli gate is one of the most popular gate . Quantum cost of a Toffoli gate is 5.

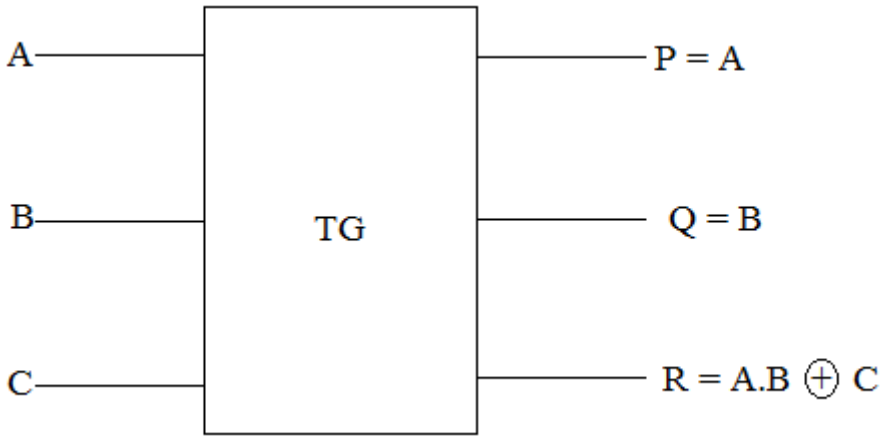


Figure 2.2 Toffoli gate

Table 2.2 Truth table of Toffoli gate

INPUT			OUTPUT		
A	**B**	**C**	**P**	**Q**	**R**
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

2.2.3 Fredkin Gate

Three input and three output (3x3) reversible Fredkin gate shown in Figure 2.3. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P=A, Q=A′B⊕AC and R=A′C ⊕ AB. Its input and output tabulated in Table 2.3. Quantum cost of a Fredkin gate is 5.

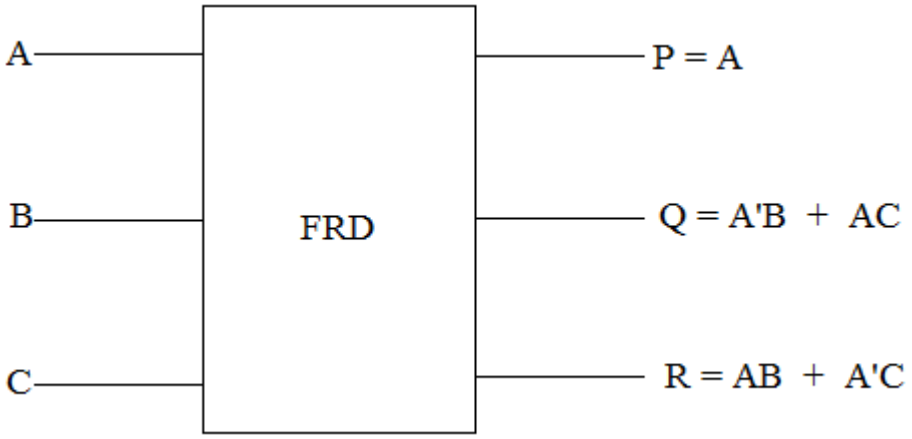


Figure 2.3 Fredkin gate

Table 2.3 Truth table of Fredkin gate

INPUT			OUTPUT		
A	**B**	**C**	**P**	**Q**	**R**
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	1	1

2.2.4 Peres Gate

Peres gate having 3 inputs and 3 outputs(3x3) reversible gate shown in Figure 2.4. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P = A, Q = A⊕B and R=AB ⊕ C. Quantum cost of a Peres gate is 4. Peres gate has the minimum number of quantum cost.

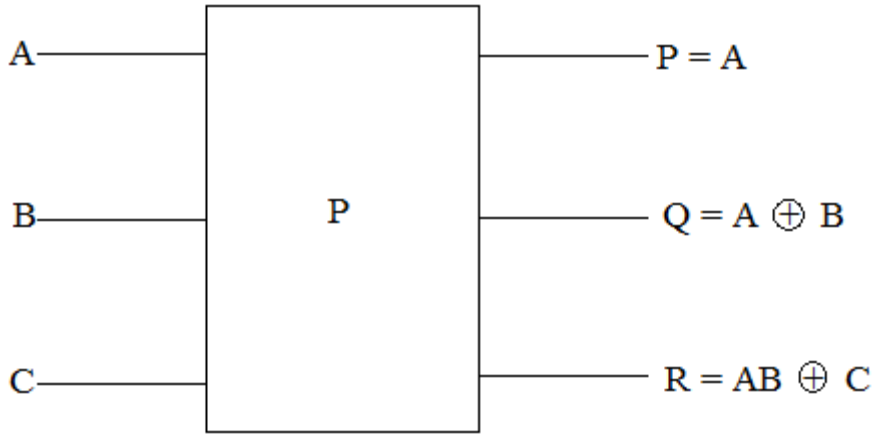


Figure 2.4 Peres gate

Table 2.4 Truth table of Peres gate

INPUT			OUPUT		
A	**B**	**C**	**P**	**Q**	**R**
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

2.2.5 TSG Gate

Reversible 4x4 TSG gate block diagram shown in Figure 2.5. The input vector is I (A, B, C, D) and the output vector is O (P, Q, R, S). The output is defined by P = A, Q = A'C'⊕B',R = (A'C'⊕B') ⊕ D and S =(A'C'⊕B').D⊕ (AB⊕ C). Quantum cost of a Peres gate is 4.

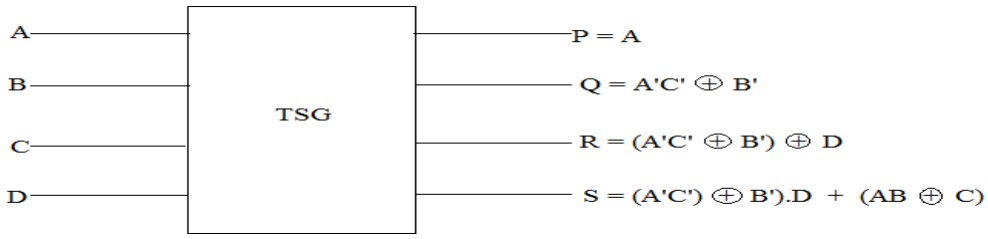


Figure 2.5 TSG gate

Table 2.5 Truth table of TSG gate

INPUT				OUPUT			
A	**B**	**C**	**D**	**P**	**Q**	**R**	**S**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	1	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	0	1	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	0	0	1
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	0
1	1	1	1	1	0	1	0

2.2.6 SayemGate

Sayen gate is a 4 inputs and 4 outputs reversible gate. The input and output vector of this gate are I = (A, B, C, D) and O = (A, A'B ⊕AC, A'B ⊕ AC⊕ D, AB ⊕ A'C ⊕D).The block diagram of this gate is shown in Figure 2.6

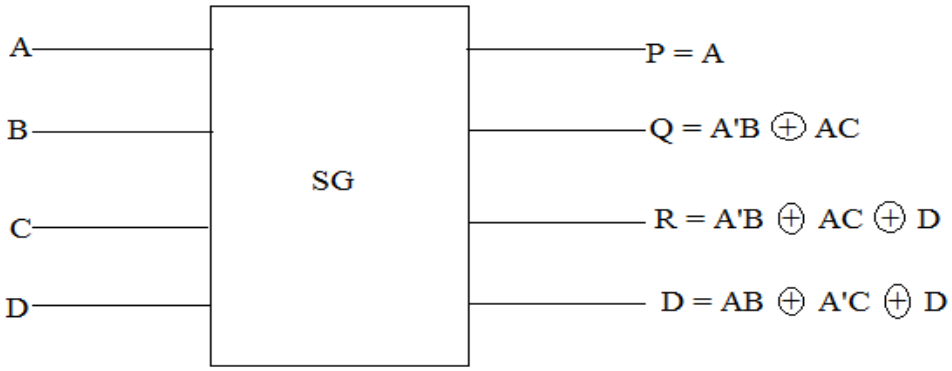


Figure 2.6 Sayem gate

Table 2.6 Truth table of Sayem gate

INPUT				OUTPUT			
A	**B**	**C**	**D**	**P**	**Q**	**R**	**S**
0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
0	1	1	1	0	0	0	0
1	0	0	0	1	0	0	0
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	0
1	0	1	1	1	0	0	1
1	1	0	0	1	0	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	1	0	0	0

2.2.7 TR Gate

The reversible TR gate is a 3 inputs 3 outputs(3x3) gate having inputs to outputs mapping as (P=A, Q=A⊕B, R = A B' ⊕ C) as shown in Figure 2.7, where A,B,C are inputs and P,Q,R are the outputs respectively.

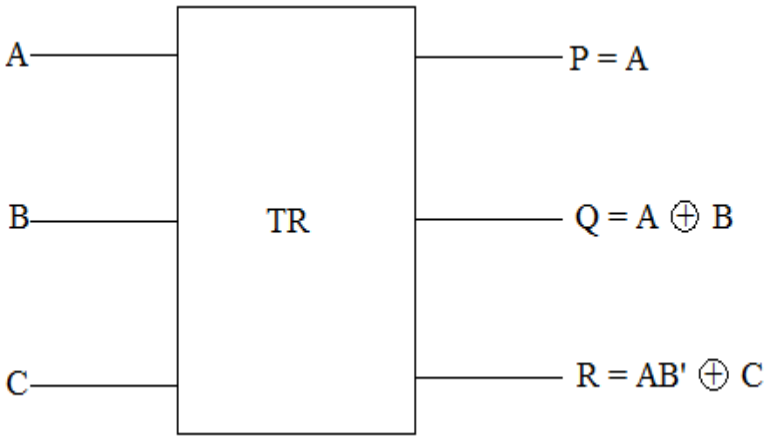


Figure 2.7 TR gate

Table 2.7 Truth table of TR gate

INPUT			OUTPUT		
A	**B**	**C**	**P**	**Q**	**R**
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	1

2.2.8 DKG Gate

DKG is a 4x4 reversibl logic gate.DKG gate with inputs A, B, C, D and outputs are P, Q, R, S. The DKG gate with 4x4 inputs and outputs shown Figure 2.8.

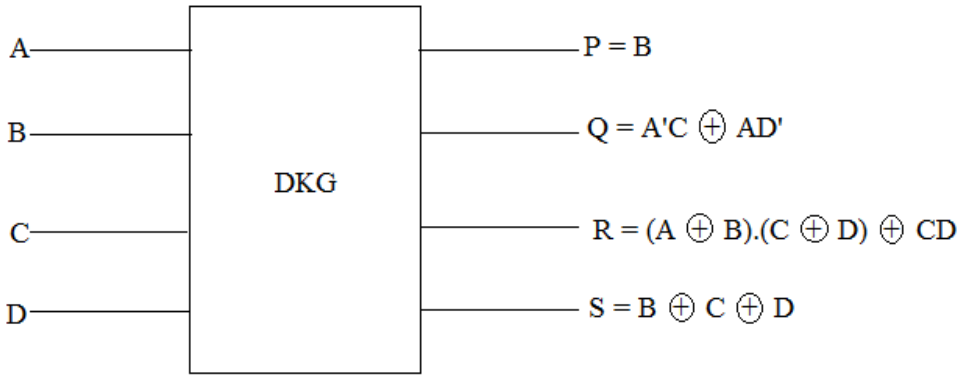


Figure 2.8 DKG gate

Table 2.8 Truth table of DKG gate

INPUT				OUTPUT			
A	**B**	**C**	**D**	**P**	**Q**	**R**	**S**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	1	0
0	1	1	1	0	1	1	1
1	0	0	0	1	1	0	0
1	0	0	1	1	0	1	1
1	0	1	0	1	1	1	1
1	0	1	1	1	0	1	0
1	1	0	0	1	1	0	1
1	1	0	1	1	0	0	0
1	1	1	0	1	1	0	0
1	1	1	1	1	0	1	1

2.3 GATE DIFFUSION INPUT

The wish to improve the performance of logic circuits, once based on traditional CMOS technology, resulted in the development of many logic design techniques during the last two decades one form of logic that is popular in low-power digital circuits is pass transistor logic (PTL). They are based on the model, where a set of control signals is applied to the gates of MOS transistors. Another set of data signals are applied to the sources of the n transistors.

Some of the main advantages of PTL over standard CMOS design are:

(i) High speed due to the small node capacitances.
(ii) Low power dissipation as a result of the reduced number of transistors.
(iii) Lower interconnection effects due to a small area.

However most of the PTL implementations have two basic problems. First, the threshold drop across the single-channel pass transistors results in reduced current drive and hence slower operation at reduced supply voltages, this is particularly important for low power design since it is desirable to operate at the lowest possible voltage level. Second, since the “high” input voltage level at the regenerative inverters is not, the pMOS device in the inverter is not fully turned off, and hence direct-path static power dissipation could be significant.

There are many sorts of PTL techniques that intend to solve the problems mentioned above:

(i) Transmission gate CMOS (TG) uses transmission gate logic to realize complex logic functions using a small number of complementary transistors. It solves the problem of low logic level swing by using pMOS as well as nMOS.

(ii) Complementary pass-transistor logic (CPL) features complementary inputs/outputs using nMOS pass-transistor logic with CMOS output inverters. CPL’s most important feature is the small stack height and the internal node low swing, which contribute to lowering the power consumption. The CPL suffers from static power consumption due to the low swing at the gates of the output inverters. To lower the power consumption of CPL circuits, LCPL and SRPL

circuit styles are used. Those styles contain pMOS restoration transistors or cross-coupled inverters (respectively).

(iii) Dual pass-transistor logic (DPL) uses complementary transistors to keep full swing operation and reduce the dc power consumption. This eliminates the need for restoration circuitry. One disadvantage of DPL is the large area used due to the presence of PMOS transistors.

An additional problem of existing PTL is top-down logic design complexity, which prevents the pass transistors from capturing a major role in real logic LSIs. One of the main reasons for this is that no simple and universal cell library is available for PTL-based design.

A new low-power design technique that allows solving most of the problems mentioned above Gate diffusion input (GDI) technique. The GDI approach allows implementation of a wide range of complex logic functions using only two transistors. This method is suitable for design of fast, low-power circuits, using a reduced number of transistors (as compared to CMOS and existing PTL techniques), while improving logic level swing and static power characteristics.

2.3.1 Basic GDI Cell

The GDI method is based on the use of a simple cell as shown in Figure 2.9. At first glance, the basic cell reminds one of the standard CMOS inverter, but there are some important differences.The GDI cell contains three inputs: G (common gate input of nMOS and pMOS), P (input to the source/drain of pMOS), and N (input to the source/drain of nMOS).Bulks of both nMOS and pMOS are connected to N or P (respectively), so it can be arbitrarily biased at contrast with a CMOS inverter. Table shows how a simple change of the input configuration of the simple GDI cell corresponds to very different Boolean functions.

Most of these functions are complex (6–12 transistors) in CMOS, as well as in standard PTL implementations, but very simple (only two transistors per function) in the GDI design method.Though GDI technique offers low power, less transistor count and high speed, the major challenges occurs in the fabrication process. The GDI

technique requires twin-well CMOS or silicon on insulator (SOI) process to realize a chip which increases the complexity as well as the cost of fabrication.

The basic structure of GDI consists of N-diffusion, P-diffusion and Gate input. In GDI technique both pMOS and nMOS are given with independent inputs so as to accommodate more logic function thereby minimizing transistor count as well as power dissipation. The main drawbacks associated with GDI are:

(i) The bulk terminals are not properly biased thereby the circuit exhibits threshold drop and variations in Vt.

(ii) Because of floating bulk the cells can be implemented in SOI process which increases the cost of fabrication.

(iii) When N diffusion input is high and P diffusion input is low the diodes between nMOS and pMOS will conduct resulting in static power dissipation. This effect reduces the output voltage to V_{out}=0.5 $V_{DD.}$

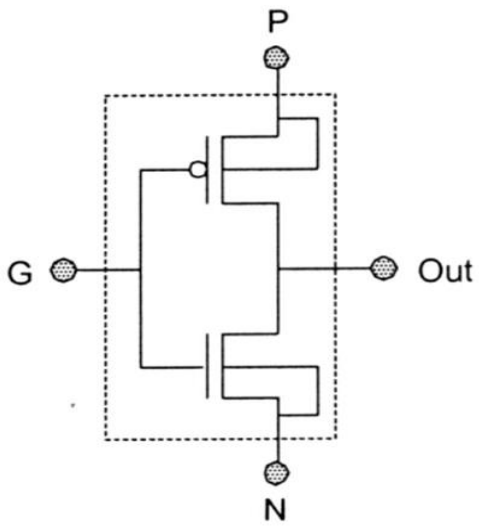
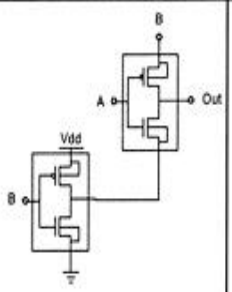
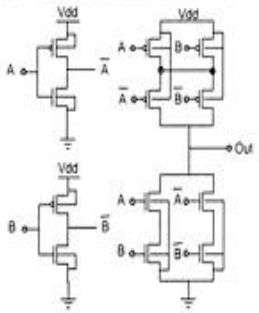
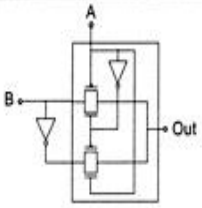
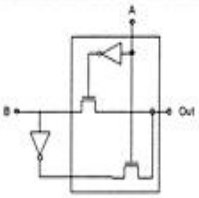
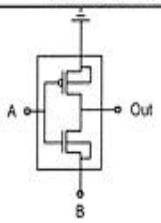
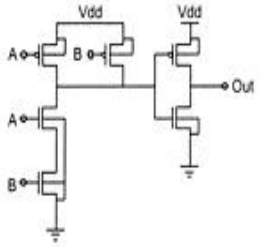
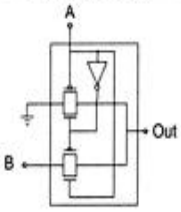
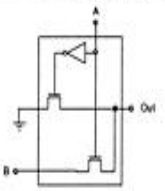
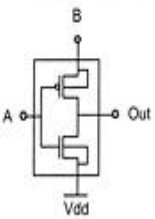
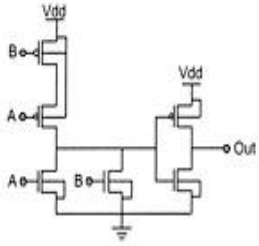
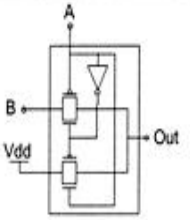
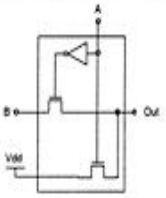


Figure 2.9 Basic GDI cell

Table 2.9 Various logic function of GDI cell

N	P	G	OUT	FUNCTION
0	B	A	A'B	F1
B	1	A	A'+B	F2
1	0	A	A+B	OR
B	B	A	AB	AND
C	1	A	A'B+AC	MUX
0	1	A	A'	NOT

Table 2.10 Basic logic cell using GDI,CMOS and PTL design techniques

	GDI	CMOS	TG	N-PG
XOR				
	4 transistors	12 transistors	8 transistors	6 transistors
AND				
	2 transistors	6 transistors	6 transistors	4 transistors
OR				
	2 transistors	6 transistors	6 transistors	4 transistors

2.4 SIMULATION TOOL

Tanner EDA version 14.11 (0.25µm) used to design all Reversible logic gates Tanner EDA is a suite of tools for the design of integrated circuits.This tool allow to enter schematics, perform SPICE simulations. There are 4 tools that are used for this process:

S-Edit - a schematic capture tool

T-Spice - the SPICE simulation engine integrated with S-Edit

L-Edit - the physical design tool

W-Edit - the waveform capture tool.

ModelSim version 6.5b used to Design the Processor using VHDL coding and simulate the output of the processor.

Xilinx version 14.2 used to implement the Processor Design. There are 3 tools that are used for this process:

ISE Design Suite – To Implement the Processor Design

PlanAhead – To obtain Layout of Devices and Packages

XPower Analyzer – To analysis Power consumed by Processor

2.5 SUMMARY

This chapter presents the literature survery of existing gates. This chapter also discusses on Reversible Logic and basic Reversible Logic Gates, Gate Diffusion Input (GDI) techniques and stimulation tool.

CHAPTER 3

DESIGN OF PROPOSED GATES

The main theme of this project is to propose a 4x4 Reversible Gate and to implement the reversible arithmetic circuit blocks using gate diffusion input (GDI) logic design technique. Initially all the existing reversible logic gates are converted into GDI based circuit and Arithmetic units such as full adder , carry look ahead adder, carry skip adder, half subtractor, full subtractor and parallel subtractor and logical units such as OR, AND, XOR, XNOR, NAND, NOR are constructed using these modified reversible logic gates.

3.1 PROPOSED REVERSIBLE GATES

3.1.1: SHA Gate

SHA is a 4x4 reversibl logic gate.SHA gate with inputs A, B, C, D and outputs are P, Q, R, S. the SHA gate with 4x4 inputs and outputs shown Figure 3.1.

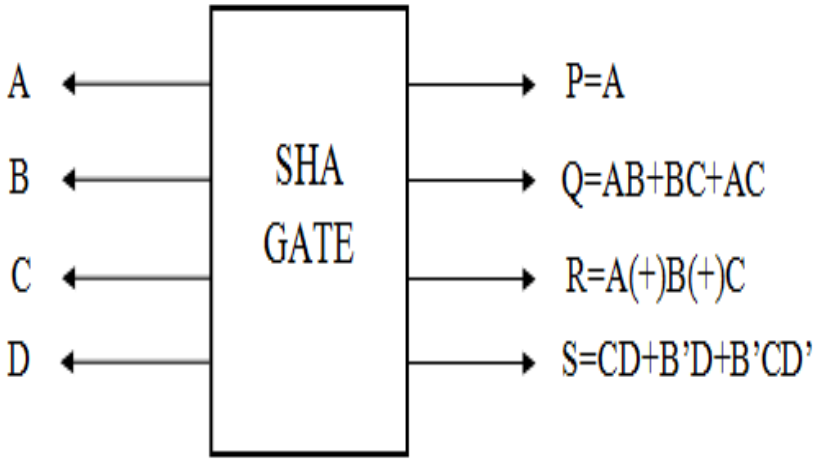


Figure 3.1 SHA gate

Table 3.1 Truth table of SHA gate

INPUT				OUTPUT			
A	**B**	**C**	**D**	**P**	**Q**	**R**	**S**
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	0
1	0	0	1	1	0	1	1
1	0	1	0	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	0	1	1	0	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	0

3.1.2: SRA Gate

SRA is a 4x4 reversibl logic gate.SRA gate with inputs A, B, C, D and outputs are P, Q, R, S. the SRA gate with 4x4 inputs and outputs shown Figure 3.2.

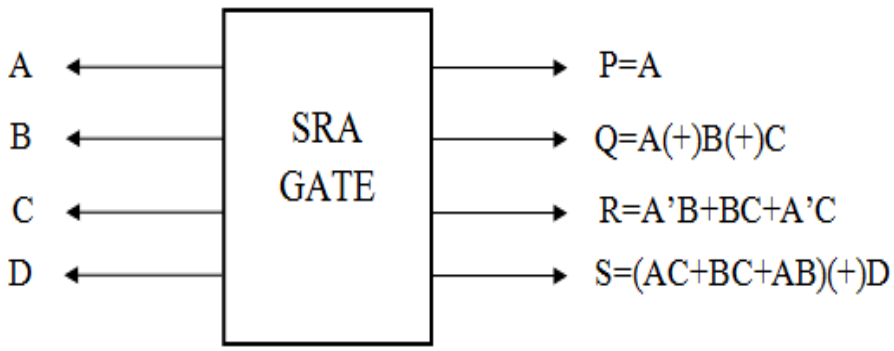


Figure 3.2 SRA gate

Table 3.2 Truth table of SRA gate

INPUT				OUTPUT			
A	**B**	**C**	**D**	**P**	**Q**	**R**	**S**
0	0	0	0	1	0	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	1	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	1
1	1	0	0	1	1	0	1
1	1	0	1	1	1	0	0
1	1	1	0	0	1	1	0
1	1	1	1	0	1	1	1

3.2 REVERSIBLE FULL ADDER DESIGN

Reversible SHA gate is shown in Figure 3.1. The corresponding truth table of the gate is shown in Table 3.1. It can be verified from the truth table that the input pattern corresponding to a particular output pattern can be uniquely determined.

SHA gate can implement all Boolean functions. Figure 3.3(a) shows the implementation of the SHA gate as XOR function. Figure 3.3(b) shows the implementation of the SHA gate as NOT function.One of the prominent functionality of this gate is that it can work singly as a reversible full adder as shown in Figure 3.4. In SHA gate, input values for A, B and C is given and D input is given low values. SHA does the functionality of Full Adder. Using single SHA gate full adder can be implemented.

Many Other reversible full adder design also available, one design requires three reversible gates (two 3*3 new gate and one 2*2 Feynman gate) and produces three garbage outputs. Another type of reversible full adder circuit requires three reversible gates (one 3*3 new gate, one 3*3 Toffoli gate and one 2*2 Feynman gate) and produces two garbage outputs.

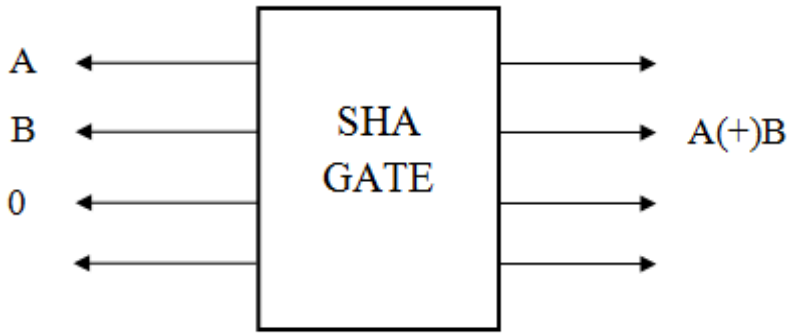


Figure 3.3 (a) SHA gate as XOR gate

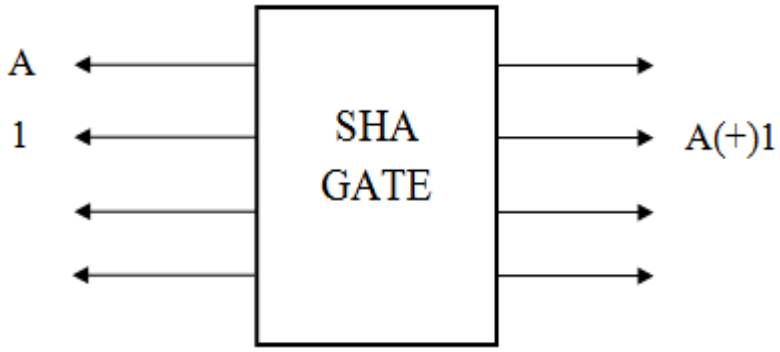


Figure 3.3 (b) SHA gate as NOT gate

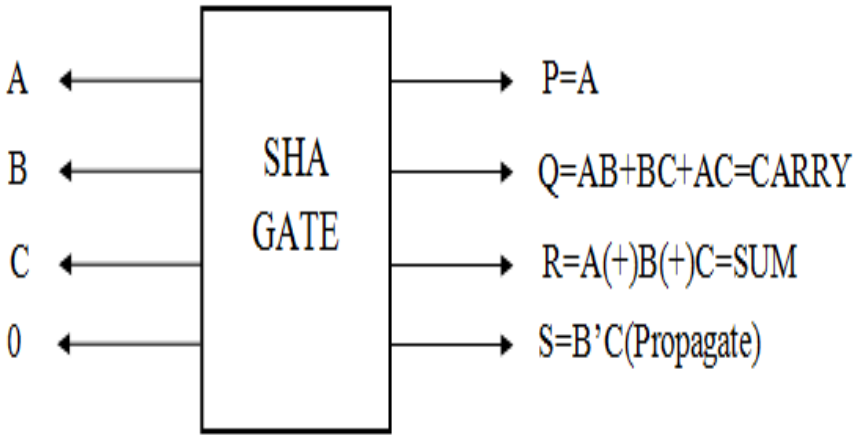


Figure 3.4 SHA gate as Full Adder

Another possible design requires five reversible Fredkin gate and produces five garbage outputs. One more Full adder design requires two reversible gates (one 3*3 New gate and one 3 *3 New Toffoli gate) and produces two garbage outputs. This implemented full adder using SHA in requires only one reversible gate (one SHA gate) and produces only two garbage outputs.

3.3 REVERSIBLE FULL SUBTRACTOR DESIGN

Reversible SRA gate is shown in Figure 3.2. The corresponding truth table of the gate is shown in Table 3.2. It can be verified from the truth table that the input pattern corresponding to a particular output pattern can be uniquely determined.

SRA gate can implement all Boolean functions. Figure 3.5 shows the implementation of SRA gate as NAND gate. Since the NAND gate is a universal gate and any Boolean function can be implemented through it. Hence SRA gate can be used to implement any Boolean function. One of the prominent functionality of this gate is that it can work singly as a reversible full subtractor as shown in Figure 3.6.

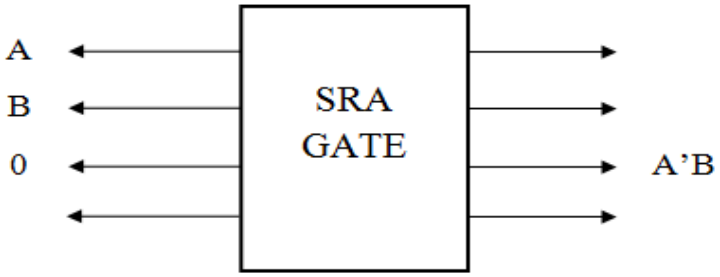


Figure 3.5 SRA gate as AND gate

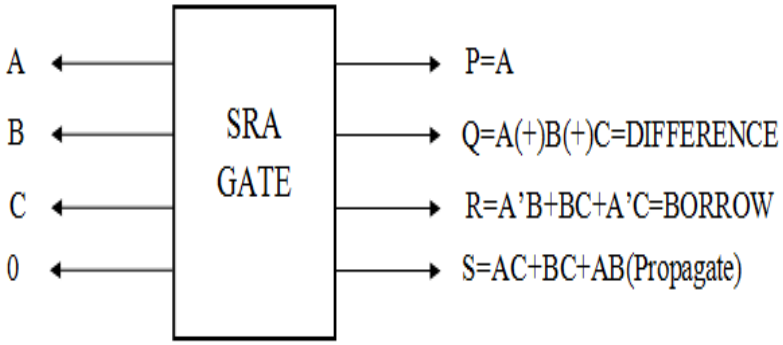


Figure 3.6 SRA gate as Full Subtractor

Another possible design requires two TR gates to design a reversible full subtractor with only two garbage outputs and quantum cost of 12. Thus the TR gate realizes the full subtractor with bare minimum of two garbage outputs. The existing design which requires 5 reversible gates, 9 garbage outputs and quantum cost of 17 of reversible. This implemented full subtractor using SHA in requires only one reversible gate (one SRA gate) and produces only two garbage outputs.

3.4 SUMMARY

This Chapter discusses about the Proposed Reversible Gates along with its truth table. The two proposed gates are used to implement some basic functions such as XOR, NOT, AND, Full Adder and Full Subtractor.

CHAPTER 4

RESULTS AND DISCUSSION

The simulation results of various reversible arithmetic blocks obtained using Tanner EDA tool are discussed. Simulations have been performed based on 0.25µm CMOS technology with supply voltage 5V. The performances of these reversible logic blocks are analysed based on circuit delay, rise time, fall time, power consumptions and number of reversible logic gates used to build the arithmetic circuits. The simulation of the Processor obtained using ModelSim tool and Implementation of Design obtained using Xilinx tool.

4.1 BASIC LOGIC CELLSIMULATION RESULT USING VARIOUS LOGIC DESIGN TECHNIQUE

The first necessary step to take towards low power design is to study the behaviour of a circuit under different logic design technique. In this project, to find the best logic design technique, various logic design technique such as CMOS, Transmission gate (TG) logic, pass transistor logic (PTL), complementary pass transistor logic (CPL), double pass Transistor logic (DPL) and gate diffusion input (GDI) were studied.

Basic logic gates such as AND, OR, NAND, NOR, XOR, XNOR are simulated and evaluated in terms of power, delay , rise time , fall time and number of transistor used. These parameters are compared and shown in Table 4.1.Amonggate diffusion input (GDI) is a lowest power logic design technique which offers improved logic swing and less static power dissipation. Using this technique several logic functions can be implemented using less number of transistor counts. This method is suitable for design of fast, low-power circuits using a reduced number of transistors (as compared to other logic design technique). The comparison bar graph of power and delay characteristics is shown in Figure 4.1(a) andFigure 4.1(b).

Table 4.1 Comparison of various logic design techniques

	Parameter	CMOS	TG	DPL	GDI
AND	Delay (ns)	**1.0034**	**0.02636**	**0.14389**	**0.53715**
	Rise time(ns)	**2.0446**	**2.4382**	**2.2802**	**0.39605**
	Fall time(ns)	**1.2964**	**3.9869**	**3.6098**	**1.2914**
	Full swing output (v)	**5**	**5**	**5**	**5**
	Power (µW)	**108.24**	**158.8W**	**519.20**	**126.52**
	Transistor count	**6**	**8**	**8**	**6**
OR	Delay (ns)	**22.508**	**22.730**	**22.108**	**23.100**
	Rise time(ns)	**2.0154**	**2.1113**	**1.5639**	**0.26430**
	Fall time(ns)	**2.0099**	**2.5559**	**3.2124**	**0.32024**
	Full swing output (v)	**5**	**5**	**5**	**5**
	Power (µW)	**55.90**	**103.16**	**117.38**	**14.39**
	Transistor count	**6**	**8**	**8**	**6**
NAND	Delay (ns)	**0.8350**	**0.12238**	**0.56167**	**0.43275**
	Rise time(ns)	**2.3229**	**2.2936**	**2.0676**	**2.2760**
	Fall time(ns)	**3.4111**	**1.3581**	**1.0269**	**1.0305**
	Full swing output (v)	**5V**	**5**	**5**	**5**
	Power (µW)	**95.16**	**74.33**	**186.01**	**130**
	Transistor count	**4**	**8**	**8**	**4**

NOR	Delay (ns)	**22.510**	**22.485**	**22.509**	**23.112**
	Rise time(ns)	**2.1504**	**2.1631**	**2.1442**	**0.31170**
	Fall time(ns)	**2.0346**	**1.4407**	**1.0803**	**0.19946**
	Full swing output (v)	**5V**	**5**	**5**	**5**
	Power (µW)	**55.29**	**58.73**	**115.29**	**14.72**
	Transistor count	**4**	**8**	**8**	**8**
XOR	Delay (ns)	**27.488**	**51.129**	**50.982**	**50.752**
	Rise time(ns)	**57.196**	**59.378**	**59.378**	**1.0308**
	Fall time(ns)	**0.3769**	**2.0103**	**98.458**	**0.28821**
	Full Swing output (v)	**5**	**5**	**5**	**5**
	Power (µW)	**140.15**	**115**	**150.96**	**87.93**
	Transistor count	**12**	**8**	**8**	**8**
XNOR	Delay (ns)	**27.422**	**50.908**	**27.660**	**51.436**
	Rise time(ns)	**0.50526**	**2.0311**		**2.0307**
	Fall time(ns)	**0.15469**	**58.528**	**58.533**	**0.57329**
	Full Swing output (v)	**5**	**5**	**5**	**5**
	Power (µW)	**140.35**	**228.03**	**133.94**	**56.23**
	Transistor count	**10**	**8**	**8**	**8**

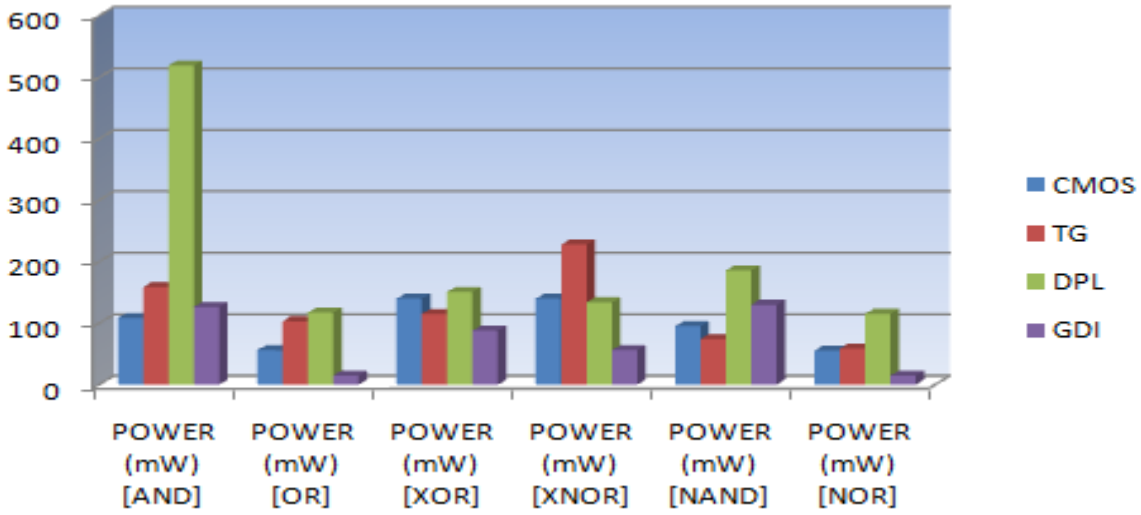


Figure 4.1(a): Comparison of Power characteristics using various Logic Techniques

4.2 REVERSIBLE FULL ADDER

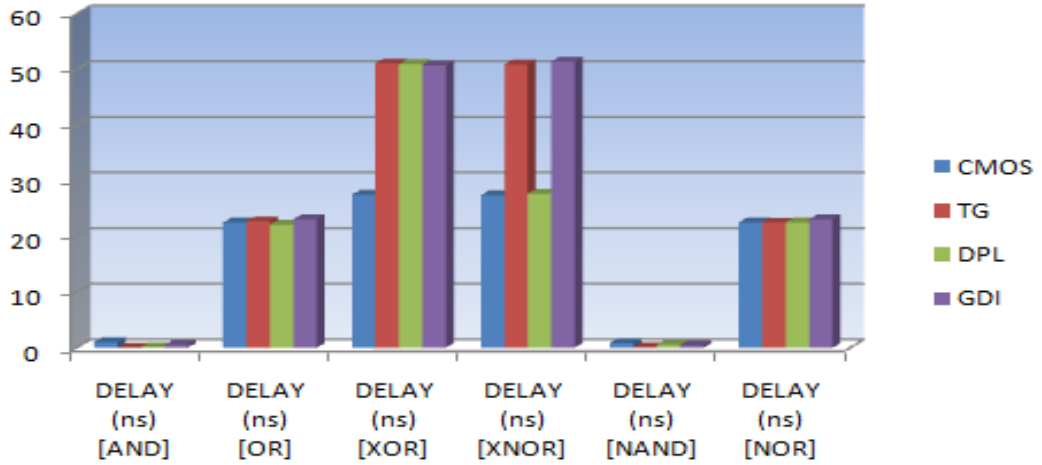


Figure 4.1(b): Comparison of Delay characteristics of various Logic Techniques

Single SHA gate is configured as reversible full adder. To verify the functionality of designed full adder 8 input bit values are given to input A,B,C from full adder truth table as shown in Table 4.2 and corresponding sum(S) and carry(CR) are verified in output waveform(Figure 4.2) and output parameter are obtained and tabulated in Table 4.3

Table 4.2 Sample input and output of Reversible Full Adder

INPTU			OUTPUT	
C	**B**	**A**	**S**	**CR**
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

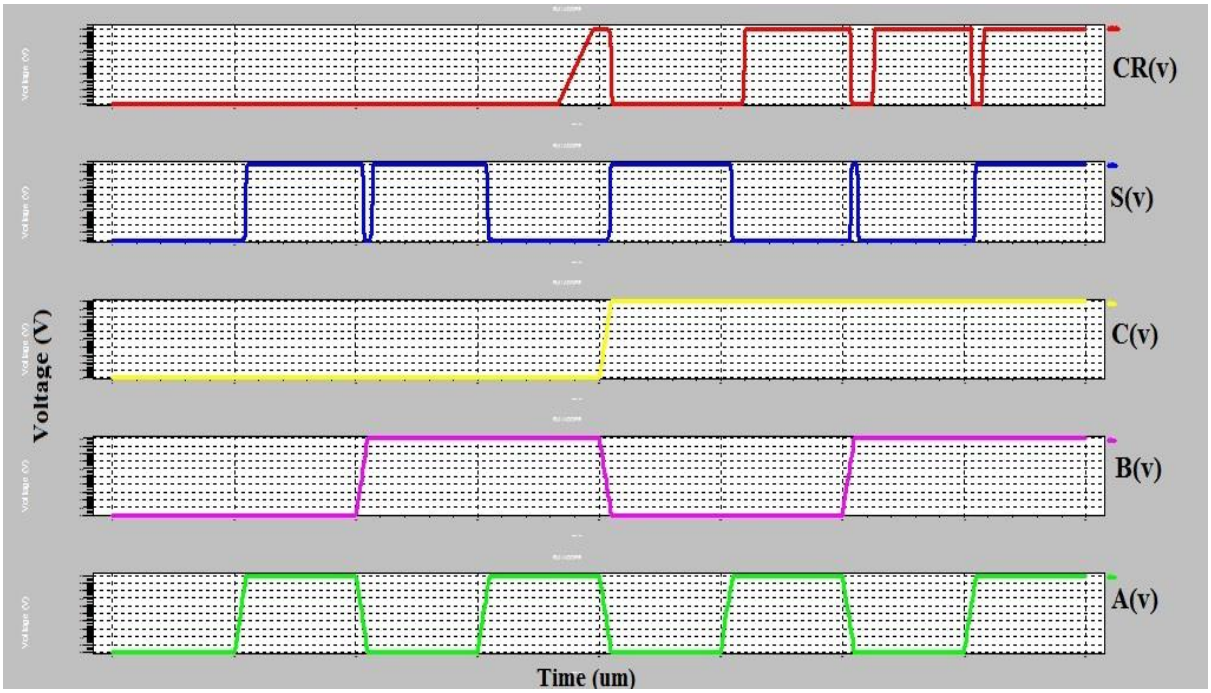


Figure 4.2 Output waveform of reversible full adder

Table 4.3 Output parameters of reversible full adder

NO. of Gates	Delay (ns)	Rise Time (ns)	Fall Time (ns)	Power (mW)
1(SHA)	0.31902	0.01542	0.09275	4.02

4.3 REVERSIBLE FULL SUBTRACTOR

Single SRA gate is configured as reversible full subtractor. To verify the functionality of designed full subtractor 8 bit input values are given to input of A,B,C from full subtractor truth table as shown in Table 4.4.

The corresponding difference (DIF) and borrow (BOR) are verified with output waveform (Figure 4.3) and output parameter are obtained and tabulated in Table 4.5

Table 4.4 Sample input and output of Reversible Full Subtractor

INPUT			OUTPUT	
C	**B**	**A**	**DIF**	**BOR**
0	0	0	0	0
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	1

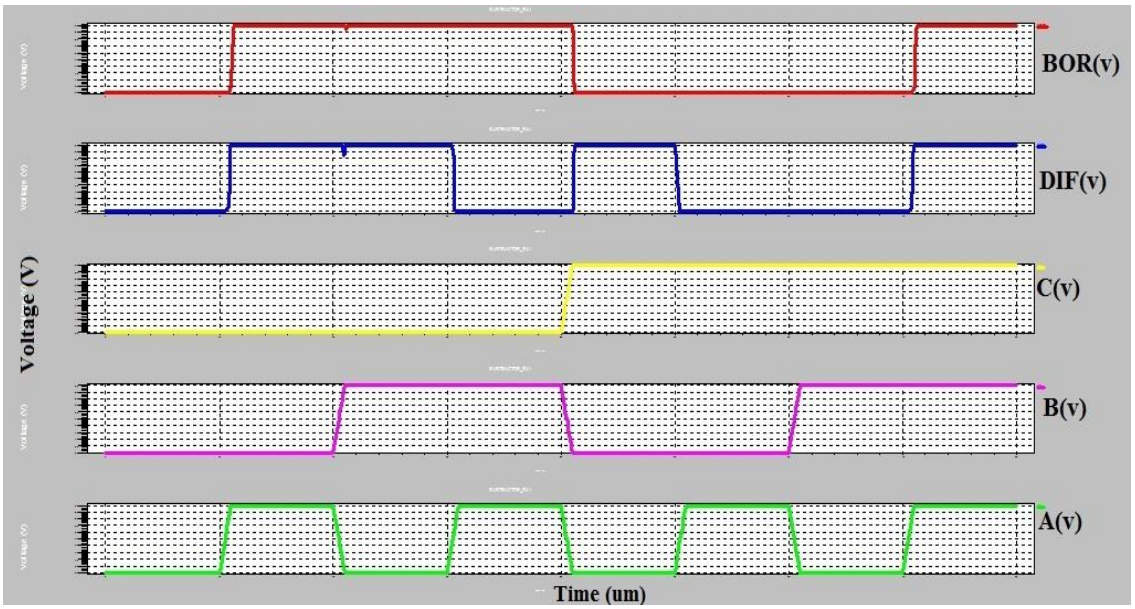


Figure 4.3 Outputwaveform of reversible full subtractor

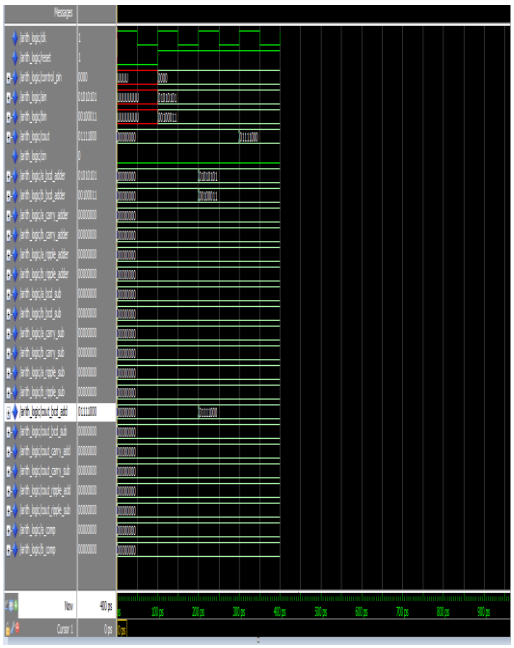
Table 4.5 Output parameters of reversible full subtractor

No. Of Gates	Delay (Ns)	Rise Time(Ns)	Fall Time (Ns)	Power(Mw)
1(SRA)	0.28383	0.08642	0.08642	1.64

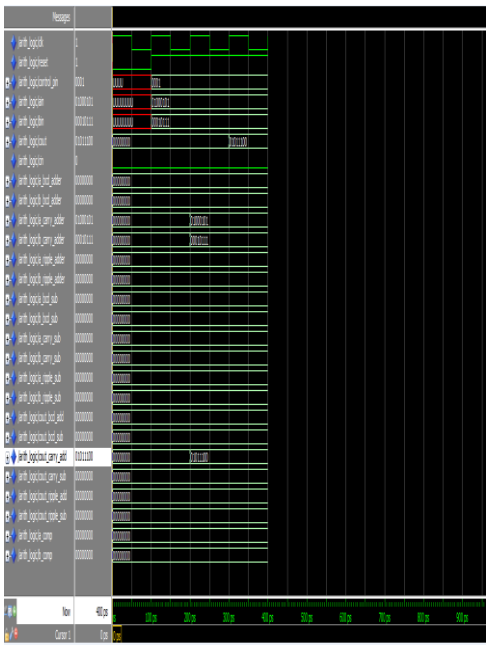
4.4 SIMULATION RESULT

4.4.1 Simulation of Adder and Subtractor

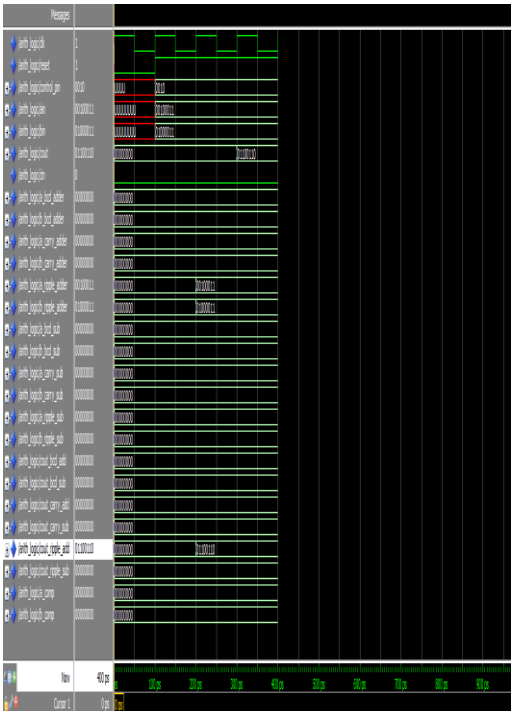
This section presents the simulated results of adder and subtractortopologies. The adder and subtractortopologies are simulated using ModelSim 6.5 b. In Model Sim, VHDL coding is used to design adder and subtractorandto simulate the output. The simulations of adder and subtractor topologies are shown in Figure 4.4.



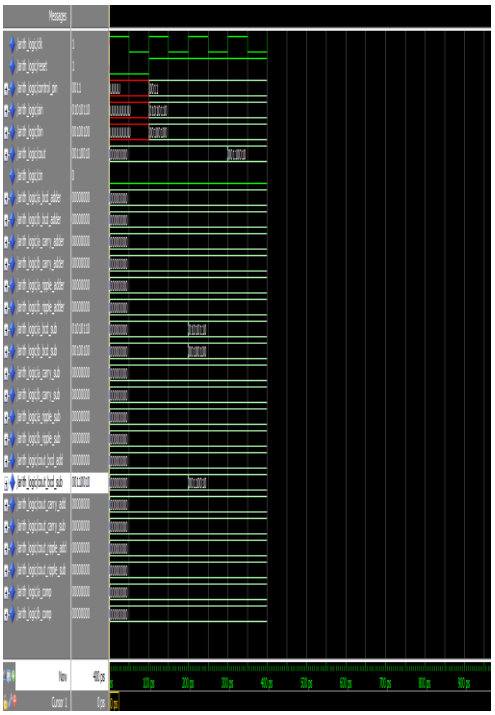
a. BCD Adder



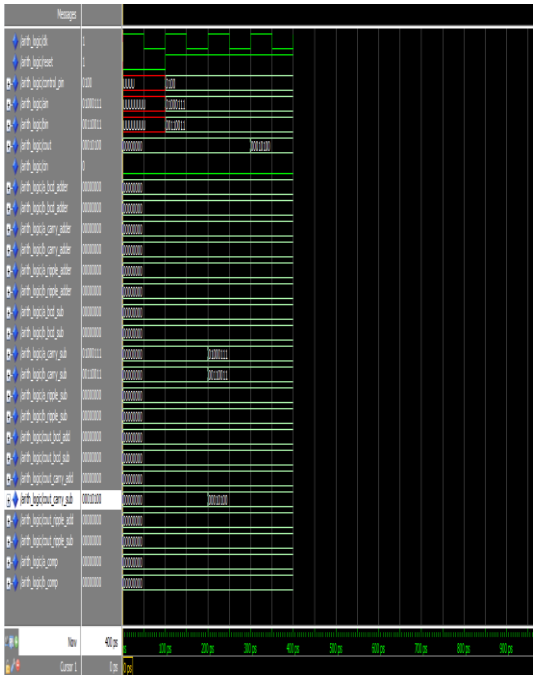
b. Carry Skip Adder

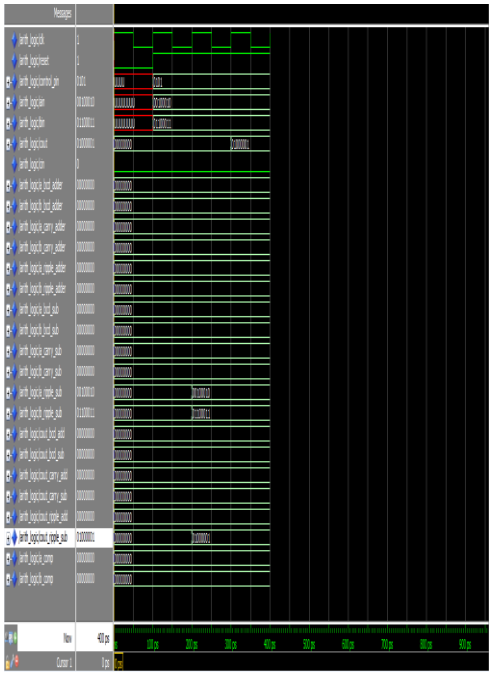


c. Ripple Adder



d. BCD Subtractor



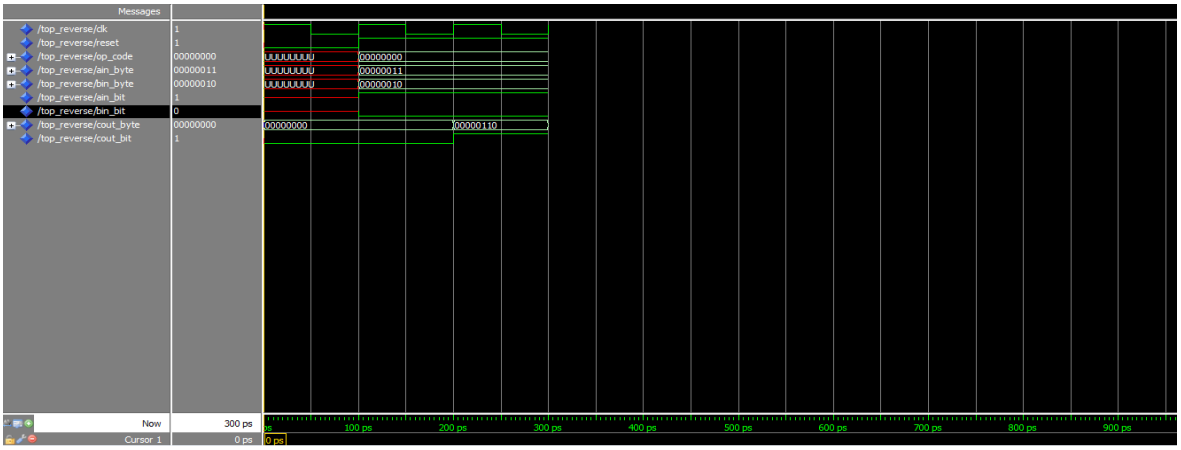


e. Carry Skip Subtractor **f. Ripple subtractor**

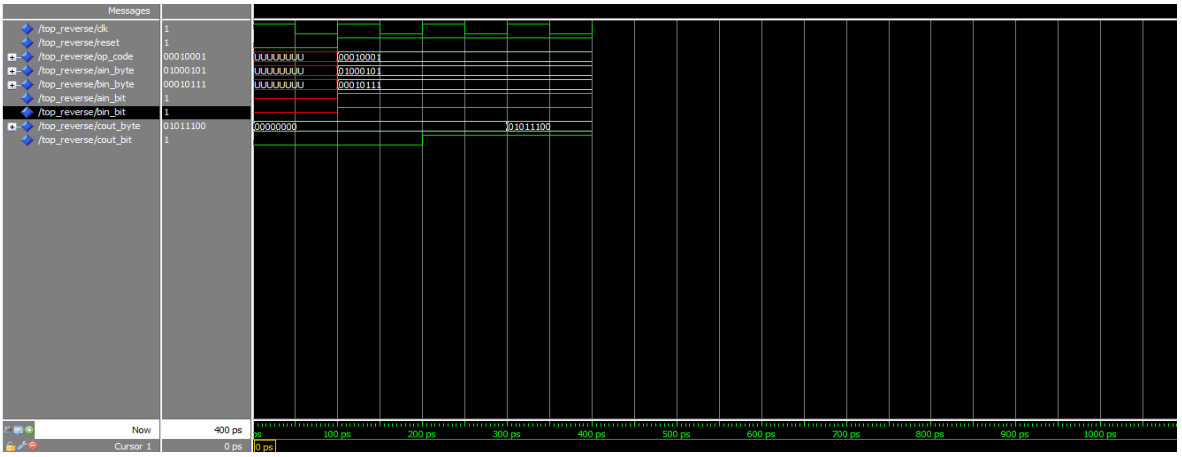
Figure 4.4: Simulation Result of Adder and Subtractor Topologies

4.4.2. Simulation of Processor output

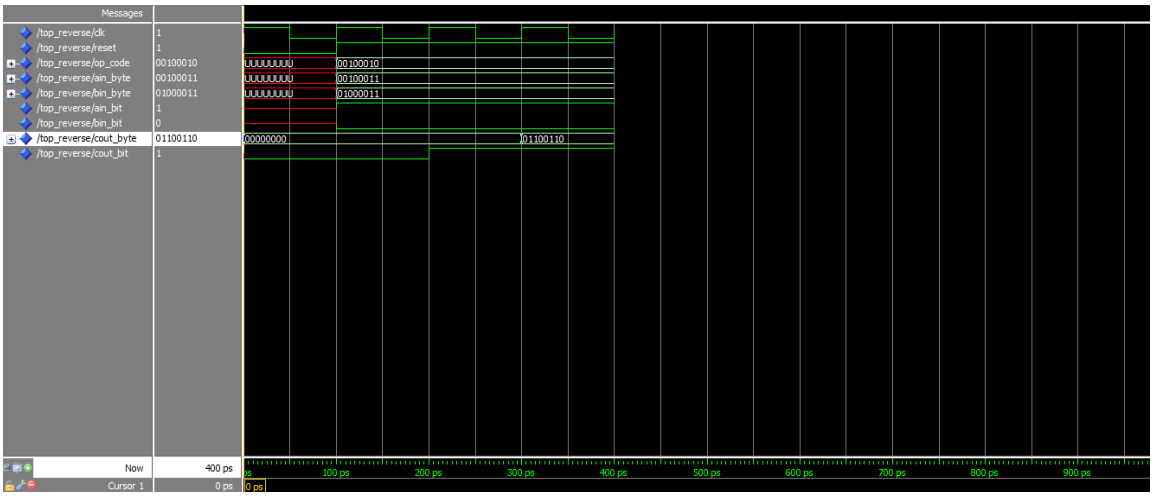
This section presents the simulated results of Processor operations. The Processor operations are simulated using ModelSim 6.5b. In ModelSim, VHDL coding is used to design Processor by using Fredkin and Toffoli Gate as basic gates. The complete arithmetic and logical operations are coded separately and called under the single top module named as Top Reverse. The simulations of Processor operations are shown in Figure 4.5.



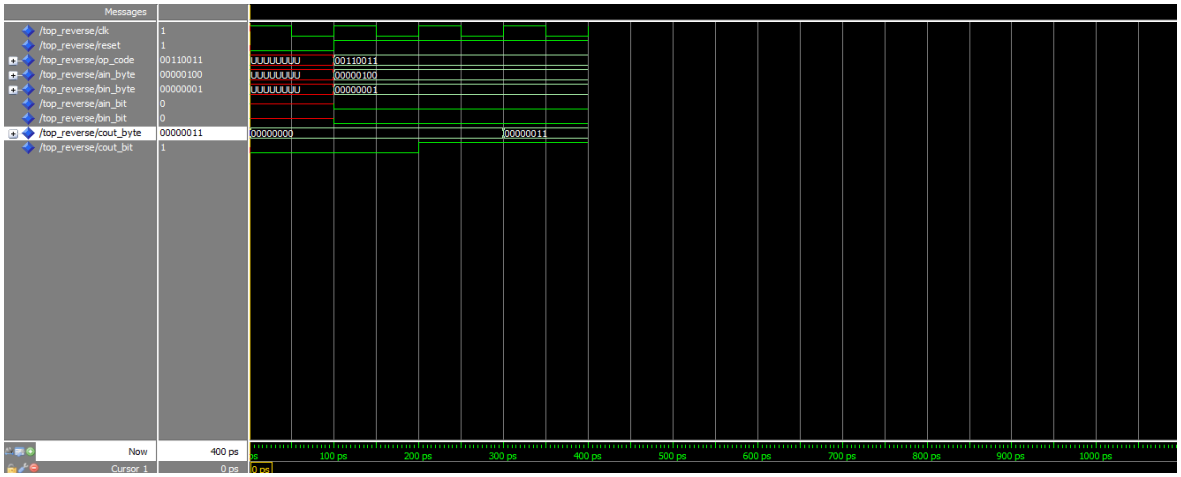
a. BCD Adder and OR GATE



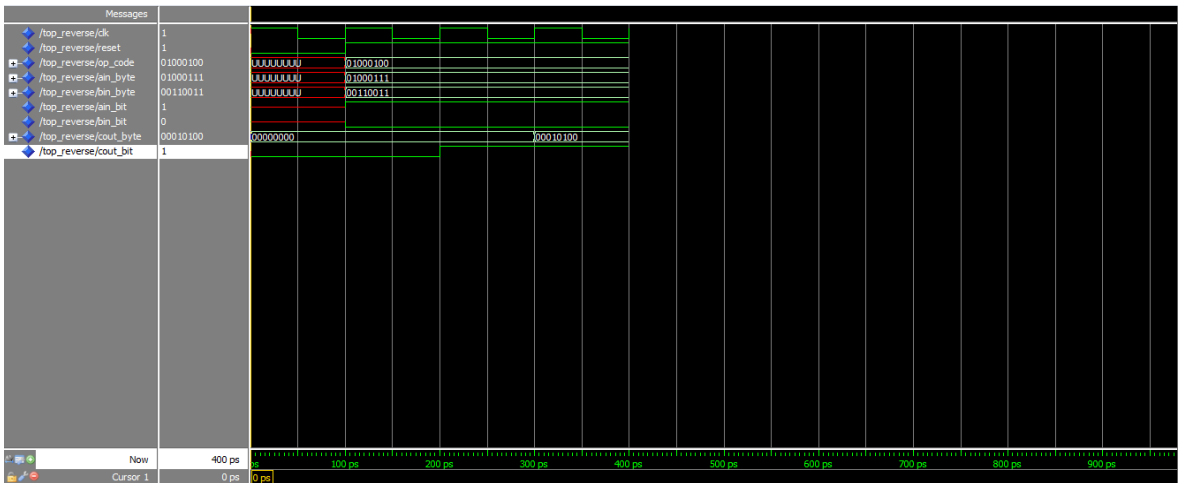
b. Carry Skip Adder and AND GATE



c. Ripple Adder and XOR GATE



d. BCD Subtractor and XNOR GATE



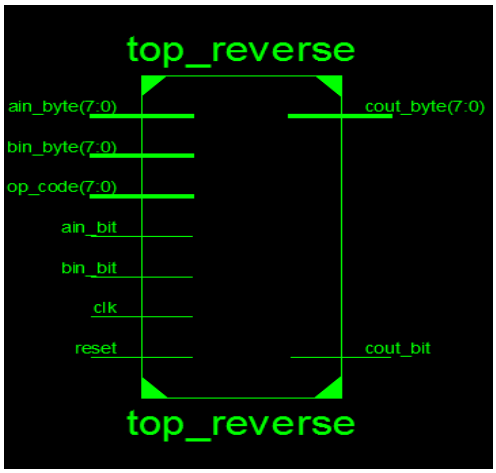
e. Carry Skip Subtractor and NAND GATE

f. Ripple Subtractor and NOR GATE

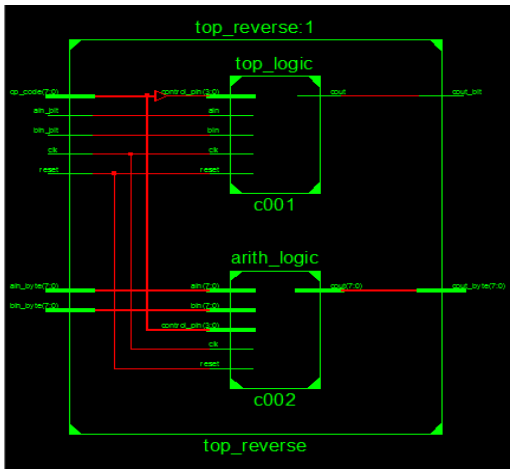
Figure 4.5: Simulation Result of Processor operation

4.5 IMPLEMENTATION OF PROCESSOR DESIGN

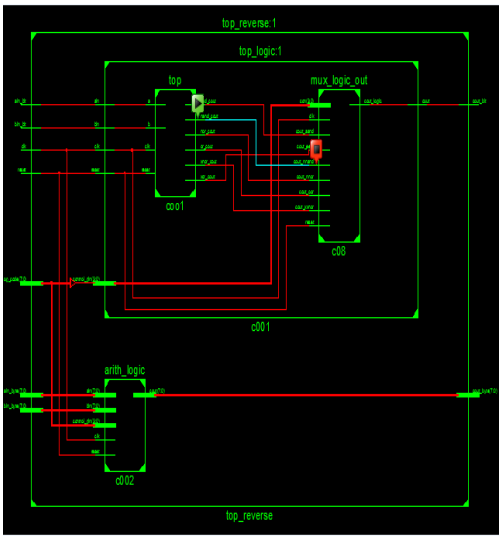
The Design implementation of Processor is implemented using the Xilinx version 14.2 software tool. In Xilinx 14.2, ISE Design Suite is used to implement the Processorby importing VHDL coding and selecting the top module as the top reverse. Then the Design of Processor is synthesised and implemented the complete Design. The Plan Ahead is used to implement the Processor package and device. Design of processor, device and packages are shown in Figure 4.6.



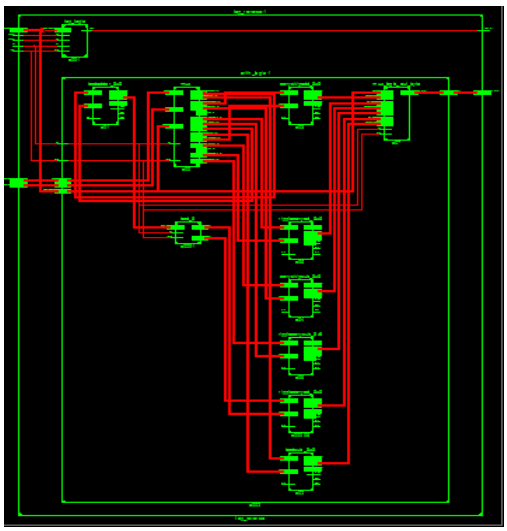
a. Top Module



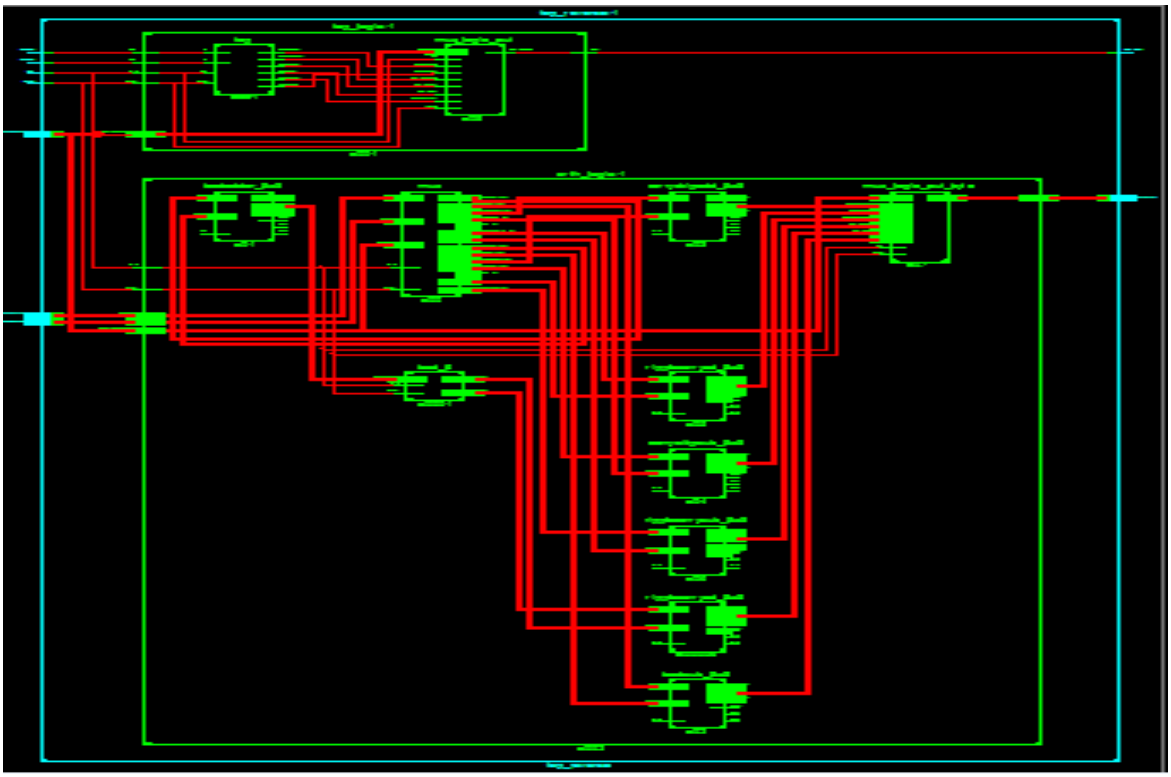
b. Arithmetic and Logic Module



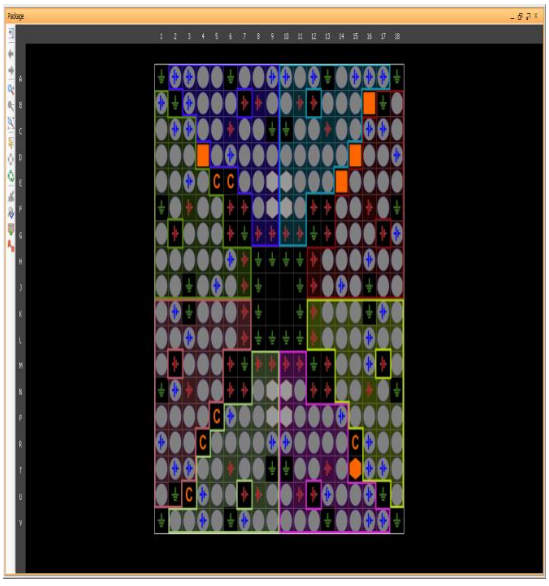
c. Logic Module



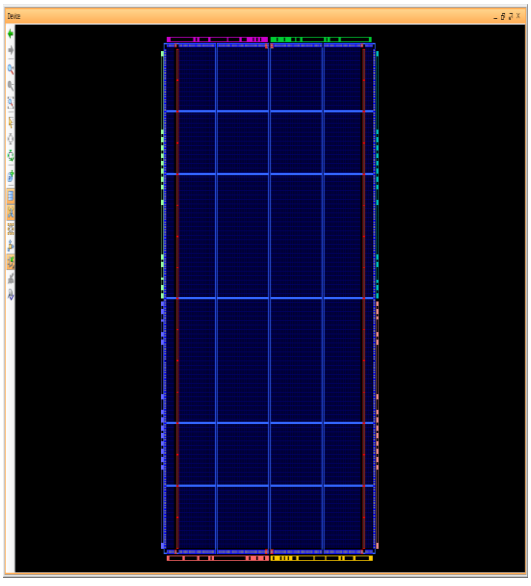
d. Arithmetic Module



e. Complete Processor Module



f. Processor Package



g. Processor Devices

Figure 4.6: Implementation of Processor Design

4.6 PROCESSOR ANALYSIS

Processor consumes power to perform arithmetic and logical operations. Powers consumed by processor for performing operations are measure by using the XPower Analyser of the Xilinx 14.2 tool and values are given in Table 4.6. The XPower Analyser is also used to measure the Thermal Properties of the processor and values are given in Table 4.7.

Table 4.6: Power consumed by Processor

Supply Source	Summary Voltage(V)	Total Current (A)
V_{ccint}	1.200	0.047
V_{ccaux}	2.500	0.035
V_{cco}	2.500	0.003

Table 4.7: Thermal Properties of Processor

Effective TJA (C/W)	Maximum Ambient (C)	Junction Temperature (C)
20.4	81.9	28.1

4.7 SUMMARY

In this work, outputs of various logic design techniques of logic gates are compared. The output waveform and output parameter of Full Adder, Full Subtractor of proposed gates are obtained. Adder topologies and Processor design are simulated, Design of Processor are implemented. Thermal properties and power consumption of processor are obtained.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

In this project, we present Design of 8-bit Reversible Processor. We have proposed two 4x4 Reversible logic gate using GDI technique and Performance of proposed gate is evaluated in terms of power, delay and number of gates. We have also compared various logic techniques in terms of rise time, fall time, delay, power and number of transistors using Tanner EDA tool. Arithmetic and Logical unit of Processor is designed and simulated using ModelSim 6.5b. Processor design is implemented using Xilinx version 14.2.

5.2 SCOPE FOR FUTURE WORK

As a future scope of the work, design of sequential and control circuits based on reversible logic can be done and power of reversible circuit can be further reduced by sizing the transistor keeping logical effort in mind. The design of the Processor can be further extended to 16 or 32-bit. Processor can be designed using 4x4 Reversible Gate, so that number of transistor can be reduced in the design and we can minimize of designing Multiplier and Divider using Reversible Logic.

REFERENCES

[1] Yibin Ye and Kaushik Roy, "QSERL: Quasi-Static Energy Recovery Logic," *IEEE J. Solid –State circuits,* vol. 36, pp. 239 – 248, Feb. 2001.

[2] Massimo Alioto and Gaetano Palumbo, "NAND/NOR Adiabatic Gates: Power Consumption Evaluation and Comparison versus the Fan – In," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 9, pp. 1253 – 1262, Sept. 2002

[3] Pallav Gupta, AbhinavAgrawal and Niraj K. Jha, "An Algorithm for Synthesis of Reversible Logic Circuits", *IEEE Transactions on Computer – Aided Design of Integrated Circuits and Systems*, vol. 25, no. 11, pp. 2317- 2330, Nov. 2006.

[4] Dmitri Maslov and Mehdi Saeedi, "Reversible Circuit Optimization Via Leaving the Boolean Domain," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems,* vol. 30, no. 6, pp. 806 – 816, June 2011.

[5] DigantaSengupta ,Mahamuda Sultana , AtalChaudhuri , "Realization of a Novel Reversible SCG Gate and its Application for Designing Parallel Adder/Subtractor and Match Logic", *International Journal of Computer Applications (0975 – 8887)* Volume 31– No.9, pp. 30-35, October 2011.

[6] Rashmi S. B1, Praveen B2, and Tilak B. G3, Design of Optimized Reversible BCD Adder/Subtractor, *IACSIT International Journal of Engineering and Technology, Vol.3, No.3, pp. 230 -234, June 2011.*

[7] Ravish Aradhya H V, Rekha G, Muralidhara K N, "Simulation and Synthesis of Combinational Shifter using Reversible Gates" , *International Journal of Computer Applications (0975 – 8887) Volume 42– No.21, pp. 01-07,March 2012.*

[8] *B. Raghu Kanth, B. Murali Krishna, G. Phani Kumar, J. Poornima, K. Siva Rama Krishna,* "A COMPARITIVE STUDY OF REVERSIBLE LOGIC GATES , *International Journal of VLSI & Signal Processing Applications", Vol.2,Issue 1, (51-55), ISSN 2231-3133 (Online),Feb 2012* .

[9] LekshmiViswanath, Ponni.M, "Design and Analysis of 16 Bit Reversible ALU", *IOSR Journal of Computer Engineering (IOSRJCE) ISSN : 2278-0661 Volume 1, pp. 46-53, Issue 1 (May-June 2012).*

[10] Rangaraju H G, Aakash Babu Suresh, Muralidhara K N "Design and Optimization of Reversible Multiplier Circuit", *International Journal of Computer Applications (0975 – 8887) Volume 52– No.10, pp. 44-50, August 2012.*

[11] R.Uma and P.Dhavachelvan, "ASIC Primitive Cells in Modified Gated Diffusion Input Technique" *International Conference on Advances in Computing (ICADC 2012) July 2012, Bangalore, Published in the book series of Springer, "Advances in Intelligent Systems and Computing" Volume 174, pp. 467-474.*

[12] R.Uma and P.Dhavachelvan, "Performance of Adders with Logical Optimization in FPGA," *First International Conference on Signal, Image Processing and Pattern Recognition (SPPR 2012) May 2012, Delhi, Published in Advances in Intelligent and Soft Computing book series of Springer," Signal, Image Processing and Pattern. pp. 245-255.*