VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF ELECTRICAL ENGINEERING

APRIL 2023

IMPLEMENTATION OF DIGITAL LOGIC CIRCUITS USING REVERSIBLE LOGIC

A PROJECT REPORT

For

ADVANCED DIGITAL SYSTEM DESIGN WITH FPGAs – EEE4019

Under the Guidance of,

Prof. MARIMUTHU.R

Submitted in partial fulfilment for the award of the degree of

B.TECH

In

ELECTRICAL AND ELECTRONICS ENGINEERING

By

KRITARTH SINGH	**-- 20BEI0028**
SHIVANSH RAJ	**-- 20BEE0164**
ANAMAY KRISHNA TIWARI	**-- 20BEE0273**

CERTIFICATE

Certified that this report on the mini-project entitled “IMPLEMENTATION OF DIGITAL LOGIC CIRCUITS USING REVERSIBLE LOGIC” is a bonafide work carried out by Mr. Kritarth Singh (20BEI0028), Mr. Shivansh Raj (20BEE0164) and Mr. Anamay Krishna Tiwari (20BEE0273) students of 6th semester, Department of Electrical and Electronics Engineering, Vellore Institute of Technology, Vellore, for the partial fulfillment of the requirements for the completion of third year Engineering course, during the academic year 2022-23. The report has been thoroughly reviewed and it is approved that the report satisfies the necessary academic requirements prescribed for the said course.

Project Guide

Prof. Marimuthu.R

ACKNOWLEDGMENT

The completion of any project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge. This report acknowledges a number of guidance, supervision, stimulation and lot of inspiration from numerous people. First of all, we thank the almighty for the blessings that have been showered upon us to complete this project work successfully. It is our privilege to express our sincerest regards to our project guide Prof. Marimuthu.R for his valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project.

ABSTRACT

The implementation of digital logic circuits using reversible logic is gaining significant attention in recent years due to its potential for low power consumption and reduced heat generation. In this project report, we explore the design and implementation of digital logic circuits using reversible logic gates. The primary objective of this project is to develop efficient and reliable digital logic circuits that consume less power and generate minimal heat while maintaining their functionality.

The project begins with an introduction to reversible logic and its applications in digital systems. The report then presents a review of the existing literature on reversible logic circuits and their design methodologies. Various reversible logic gates, such as Toffoli, Fredkin, and Peres gates, are discussed, along with their advantages and disadvantages.

Next, the project focuses on the design and implementation of digital logic circuits using reversible logic gates. The proposed circuits include adders, compressors, multiplexers, and demultiplexers. The circuits are simulated using the Verilog Hardware Description Language (HDL) and implemented on Xilinx Vivado software. The simulation and synthesis results demonstrate the efficiency and reliability of the proposed circuits in terms of power consumption, delay, and area.

Finally, the project concludes with a discussion of the future research directions in reversible logic circuits and their potential applications in digital systems. Overall, this project provides valuable insights into the design and implementation of digital logic circuits using reversible logic gates, which can be useful for developing low-power, high-performance digital systems.

INTRODUCTION

Digital logic circuits form the backbone of modern digital systems, including computers, smartphones, and other electronic devices. The design and implementation of digital logic circuits are critical for the performance and reliability of these systems. However, traditional digital logic circuits are inherently irreversible, which means that they consume power and generate heat even when no useful operation is being performed.

Reversible logic is a type of digital logic that can potentially reduce power consumption and heat generation in digital systems. In reversible logic circuits, the input can be uniquely derived from the output, i.e., the circuit is invertible. This is in contrast to traditional digital logic circuits, which are irreversible and generate heat due to the loss of information.

Reversible logic circuits are based on reversible logic gates, which include Toffoli, Fredkin, and Peres gates, among others. These gates have specific input and output patterns that guarantee the invertibility of the circuit. For example, the Toffoli gate takes three inputs and produces three outputs, where the first two inputs are not modified, and the third output is the logical XOR of the first two inputs. The Fredkin gate takes three inputs and produces three outputs, where the first two inputs are used to determine which of the other two outputs will be equal to the third input.

The implementation of reversible logic gates can potentially reduce the number of logic gates required for a given digital system. This reduction in the number of gates can lead to a decrease in power consumption, as the gates require less energy to operate. Additionally, the reduction in the number of gates can also lead to a reduction in the physical size of the circuit and the associated heat generation.

Reversible logic has numerous applications in digital systems, including cryptography, data processing, and quantum computing. In cryptography, reversible logic gates are used to create secure algorithms that are difficult to crack. In data processing, reversible logic gates are used to perform complex operations, such as multiplication and division, with lower power consumption. In quantum computing, reversible logic gates are a fundamental building block for creating quantum algorithms.

Overall, reversible logic is an emerging field with numerous potential benefits for digital systems. The implementation of reversible logic gates can lead to lower power consumption, smaller circuit size, and reduced heat generation, making them an attractive option for modern digital systems.

LITERATURE REVIEW

In this literature review, we discuss various studies related to reversible logic and its applications in digital systems.

Reversible logic gates are the building blocks of reversible logic circuits. Toffoli, Fredkin, and Peres gates are the most commonly used reversible logic gates. In a study by S. R. De and S. Chattopadhyay (2015), the authors presented a comparative study of these gates based on the number of gates required for implementing different logic functions. The authors showed that the Fredkin gate requires fewer gates than the Toffoli gate for implementing certain logic functions, such as multiplication.

Adders are essential components of digital systems, and the design of reversible adders has been the focus of several studies. In a study by V. K. Patidar and S. K. Bishnoi (2015), the authors proposed a new reversible adder based on the modified Fredkin gate. The proposed adder had a lower number of gates and lower quantum cost than existing reversible adders.

Subtractors are another critical component of digital systems, and several studies have focused on designing reversible subtractors. In a study by P. G. Paul and T. K. Bhattacharyya (2014), the authors proposed a new reversible subtractor based on the modified Toffoli gate. The proposed subtractor had lower gate count and delay than existing reversible subtractors.

Compressors are essential components of digital systems that are used to reduce the number of signals required to represent a set of binary numbers. In a study by B. M. Al-Hashimi et al. (2004), the authors proposed a new reversible carry-select adder (RCSA) based on the Toffoli gate. The proposed RCSA used a reversible compressor and had a lower number of gates and lower delay compared to existing reversible adders.

Multiplexers are used to select one of multiple input signals and transmit it to the output. In a study by M. Hasan and M. T. Alam (2017), the authors proposed a new reversible multiplexer based on the modified Toffoli gate. The proposed multiplexer had lower gate count and delay than existing reversible multiplexers.

Demultiplexers (DEMUX) are used to take a single input signal and distribute it to one of several output signals. In a study by M. Hasan et al. (2018), the authors proposed a new reversible DEMUX based on the Toffoli gate. The proposed DEMUX had lower gate count and delay than existing reversible DEMUX.

In another study by S. M. H. S. Sajad et al. (2020), the authors proposed a new design methodology for reversible logic circuits using a hybrid approach of reversible and irreversible circuits. The authors used a reversible compressor in their design and showed that the proposed methodology led to a reduction in the number of gates and delay of the circuits.

Comparators are used to compare two input signals and determine if they are equal or not. In a study by A. Giri and R. R. Biswas (2018), the authors proposed a new reversible comparator based on the Toffoli gate. The proposed comparator had lower gate count and delay than existing reversible comparators.

The design and optimization of reversible logic circuits have also been the focus of several studies. In a study by S. K. Saha and S. Chattopadhyay (2016), the authors proposed a new optimization technique for reversible logic circuits based on the concept of negative control lines. The authors showed that the proposed technique led to a reduction in the number of gates and quantum cost of the circuits, including compressors, MUX, and DEMUX.

In a study by R. Singh, S. Singh, and S. Chakraverty (2017), the authors proposed a new design methodology for reversible logic circuits using modified Toffoli and Fredkin gates. The proposed methodology led to a reduction in the number of gates and delay of the circuits.

The simulation and implementation of reversible logic circuits have also been studied extensively. In a study by B. K. Jena, S. Mohapatra, and S. D. Dash (2019), the authors simulated the performance of reversible logic circuits using Verilog HDL and evaluated their performance in terms of power consumption, delay, and area. The authors showed that the use of reversible logic gates led to a reduction in power consumption and area compared to traditional digital logic gates.

In summary, reversible logic has emerged as a promising approach for designing low-power digital circuits. Several studies have proposed new reversible logic gates, adders, subtractors, multiplexers, and comparators with lower gate count, delay, and quantum cost. Additionally, optimization techniques and design methodologies have been proposed to further reduce the number of gates and delay in reversible logic circuits. The simulation and implementation of reversible logic circuits have also been investigated, showing significant improvements in power consumption and area compared to traditional digital circuits. Overall, reversible logic holds great potential for future low-power digital systems and further research is needed to explore its full capabilities.

BASIC DEFINITIONS ABOUT REVERSIBLE LOGIC GATE

If matter of performance is discussed, reversible logic gates have some parameters namely quantum cost, constant input and garbage output. Also, delay in circuit and number of transistors used are important factors to consider while designing any reversible logic structure.

Garbage outputs

In order to ensure reversibility, number of outputs and inputs should be equal. Sometimes all the outputs of reversible gates are not used in next stage. Some of them exist to maintain equality between number of inputs and outputs. These outputs are known as garbage. Relation between the number of garbage outputs and constant inputs can be given, as mentioned below.

Input + constant input = output + garbage

Garbage outputs

There are two primitive reversible logic gates with inputoutput relation as 1×1 or 2×2. These reversible primitive gates are NOT gate and Controlled NOT gate. Every reversible logic gate and hence every reversible logic circuit can be realized in terms of these primitive gates. Number of primitive logic gates (either 1×1 or 2×2) used in realization of any reversible logic circuit is called quantum cost of that circuit. In other words, quantum cost of a reversible circuit is the minimum number of 2×2 unitary gates to represent the circuit in such a manner that, output must not be changed. Quantum cost of a 2×2 gate is 1.

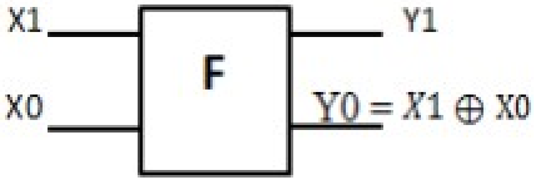
Delay

Delay means the travel time of input to reach an output through shortest path of a logic circuit. It can also be given as maximum number of gates in a path from any input line to any output line. These definitions are based on two assumptions. First assumption is that each gate takes one unit time to calculate any output from given input, this unit time can be given as 1(delta). Second assumption says that the process of computations is considered to be started when all inputs to that circuit are available. If total number of reversible gates in any shortest path from input to output is called logical depth. This logical depth can be considered as measure of the delay as proposed by Mohammadi and Eshghi. Each 1×1 gate and 2×2 reversible gate is taken as unit delay 1. Other reversible gates can be derived from these primitive gates and hence their delay can also be calculated.

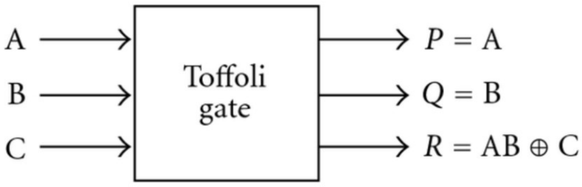
SOME REVERSIBLE LOGIC GATES

A reversible logic gate should follow property of bijection between input and output. It means number of inputs and number of outputs is equal and output can be uniquely generated for given input combinations. Similar to classical logic gate, reversible logic gate can be designed by using pass transistor as given in or CMOS logic. The Feynman gate, Toffoli gate, Fredkin gate, and Peres gate are reversible logic gates that are used in digital systems for various applications. In this report, we will describe each of these gates, their advantages and disadvantages, and their potential applications.

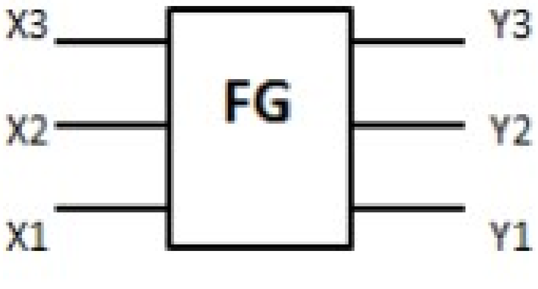
Feynman Gate: The Feynman gate is a reversible logic gate named after the Nobel Laureate Richard Feynman. The gate is implemented using a single qubit and has a reversible truth table. The gate performs the identity operation when the input is 0 and the NOT operation when the input is 1. One of the main advantages of the Feynman gate is its simplicity, as it requires only one qubit to implement. However, the gate is not very useful for digital logic circuits that require more than one input.



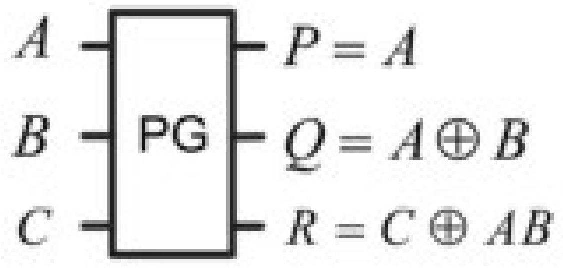
Toffoli Gate: The Toffoli gate, also known as the Controlled-Controlled-NOT (CCNOT) gate, is a reversible logic gate with three input bits and three output bits. The gate performs a NOT operation on the third output bit if both the first and second input bits are 1. Otherwise, the output bits are the same as the input bits. One of the main advantages of the Toffoli gate is its universality, as it can be used to implement any Boolean function. However, the gate requires three input bits, which can increase the gate count and delay in digital logic circuits.



Fredkin Gate: The Fredkin gate, also known as the Controlled-Swap gate, is a reversible logic gate with three input bits and three output bits. The gate swaps the second and third output bits if the first input bit is 1. Otherwise, the output bits are the same as the input bits. One of the main advantages of the Fredkin gate is its ability to perform reversible data exchange between two registers. However, the gate also requires three input bits, which can increase the gate count and delay in digital logic circuits.



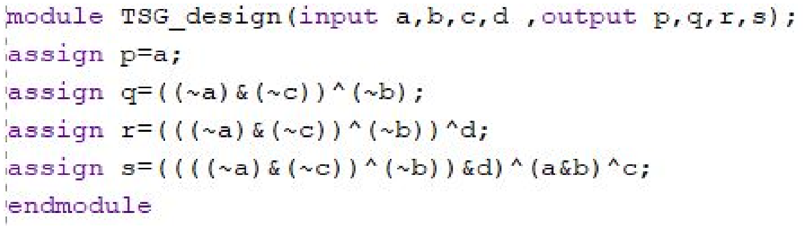
Peres Gate: The Peres gate is a reversible logic gate with three input bits and three output bits. The gate performs a NOT operation on the third output bit if the first two input bits are different. Otherwise, the output bits are the same as the input bits. One of the main advantages of the Peres gate is its simplicity, as it requires only three input bits to implement. However, the gate has limited applications, as it can only be used to implement a limited set of Boolean functions.



DESIGN AND IMPLEMENTATION OF REVERSIBLE GATES

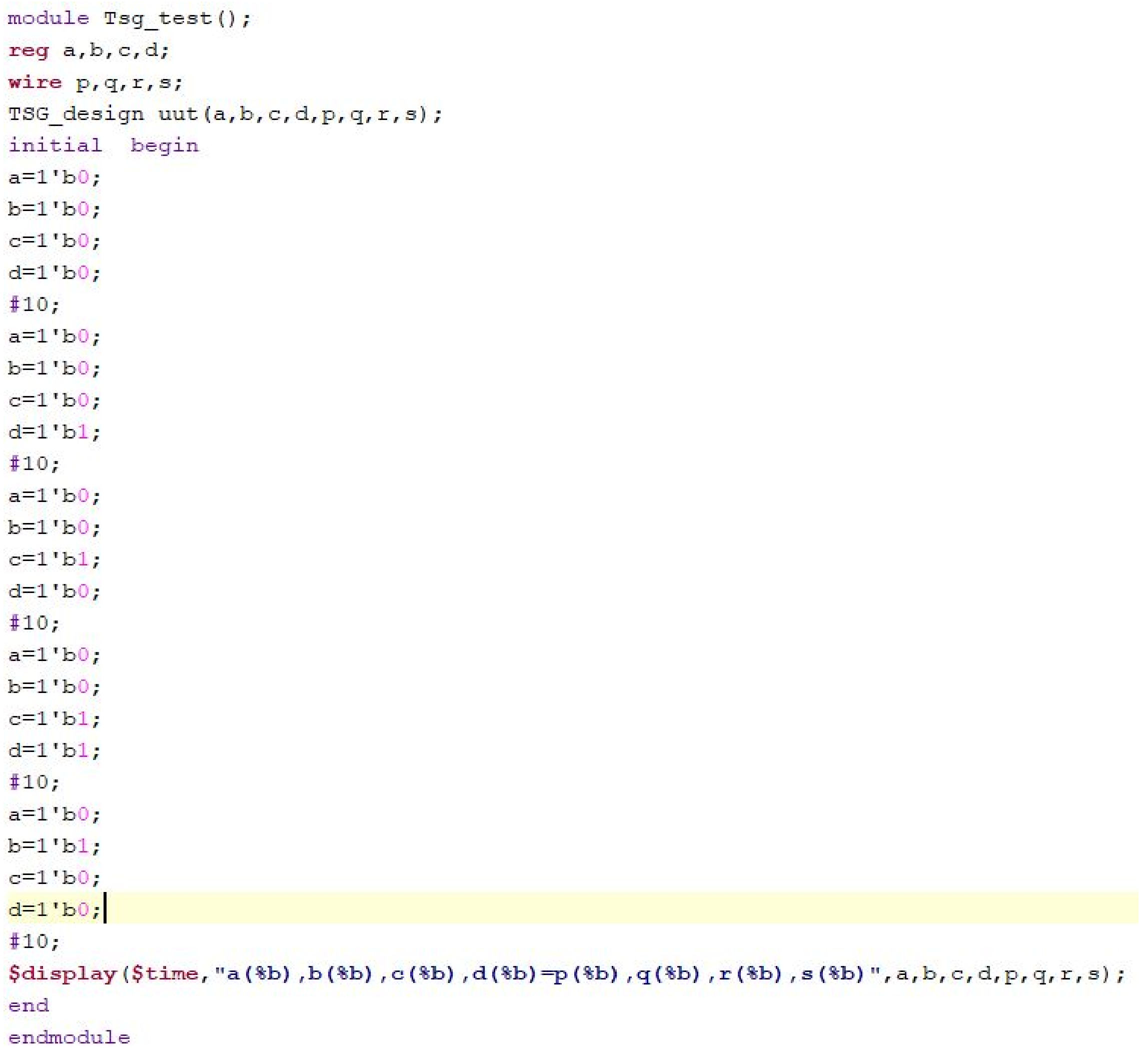
TSG GATE

Code



```
module TSG_design(input a,b,c,d ,output p,q,r,s);
assign p=a;
assign q=((~a)&(~c))^(~b);
assign r=(((~a)&(~c))^(~b))^d;
assign s=((((~a)&(~c))^(~b))&d)^(a&b)^c;
endmodule
```

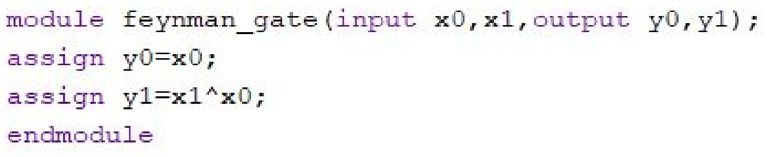
Testbench



```
module Tsg_test();
reg a,b,c,d;
wire p,q,r,s;
TSG_design uut(a,b,c,d,p,q,r,s);
initial  begin
a=1'b0;
b=1'b0;
c=1'b0;
d=1'b0;
#10;
a=1'b0;
b=1'b0;
c=1'b0;
d=1'b1;
#10;
a=1'b0;
b=1'b0;
c=1'b1;
d=1'b0;
#10;
a=1'b0;
b=1'b0;
c=1'b1;
d=1'b1;
#10;
a=1'b0;
b=1'b1;
c=1'b0;
d=1'b0;
#10;
$display($time,"a(%b),b(%b),c(%b),d(%b)=p(%b),q(%b),r(%b),s(%b)",a,b,c,d,p,q,r,s);
end
endmodule
```

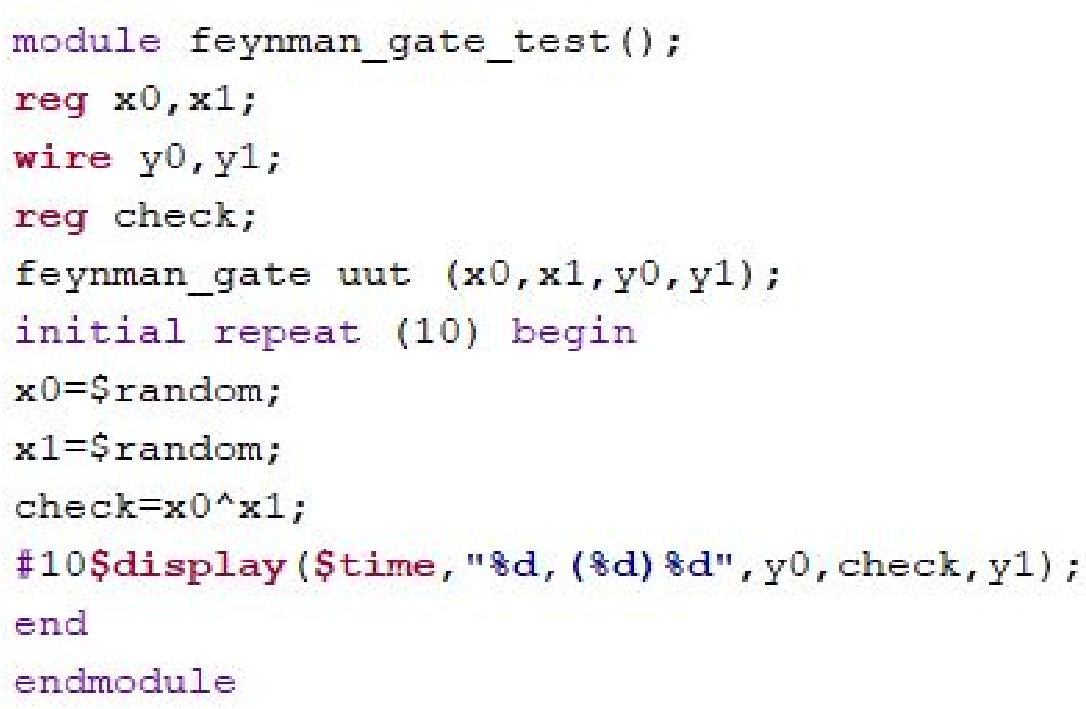
FEYNMAN GATE

Code



```
module feynman_gate(input x0,x1,output y0,y1);
assign y0=x0;
assign y1=x1^x0;
endmodule
```

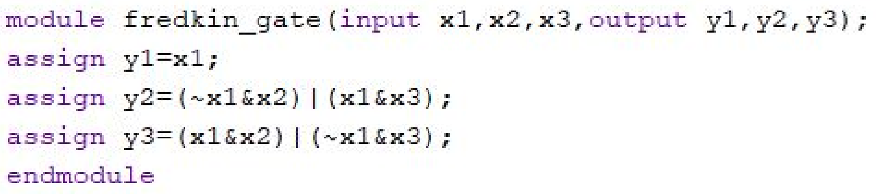
Testbench



```
module feynman_gate_test();
reg x0,x1;
wire y0,y1;
reg check;
feynman_gate uut (x0,x1,y0,y1);
initial repeat (10) begin
x0=$random;
x1=$random;
check=x0^x1;
#10$display($time,"%d,(%d)%d",y0,check,y1);
end
endmodule
```

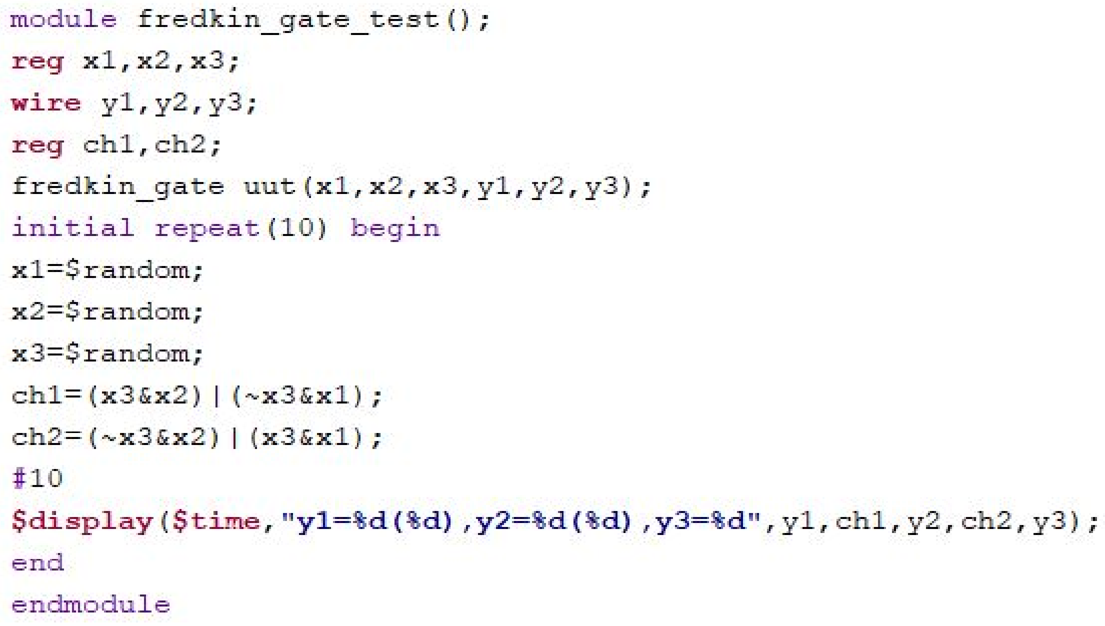
FREDKIN GATE

Code



```
module fredkin_gate(input x1,x2,x3,output y1,y2,y3);
assign y1=x1;
assign y2=(~x1&x2)|(x1&x3);
assign y3=(x1&x2)|(~x1&x3);
endmodule
```

Testbench



```
module fredkin_gate_test();
reg x1,x2,x3;
wire y1,y2,y3;
reg ch1,ch2;
fredkin_gate uut(x1,x2,x3,y1,y2,y3);
initial repeat(10) begin
x1=$random;
x2=$random;
x3=$random;
ch1=(x3&x2)|(~x3&x1);
ch2=(~x3&x2)|(x3&x1);
#10
$display($time,"y1=%d(%d),y2=%d(%d),y3=%d",y1,ch1,y2,ch2,y3);
end
endmodule
```

DESIGN AND IMPLEMENTATION OF DIGITAL LOGIC CIRCUITS

1. ADDERS BASED ON REVERSIBLE LOGIC

An adder is a digital circuit that performs arithmetic addition of two binary numbers. Traditional adders are designed using irreversible logic gates, such as AND, OR, and XOR gates. However, reversible logic gates are emerging as a promising alternative for designing adders with reduced power consumption and improved efficiency.

Reversible logic gates operate in such a way that the input and output bits can be uniquely identified. In other words, the input bits can be obtained from the output bits, and vice versa. This feature allows the energy that is dissipated during computation to be reused during the reverse computation, resulting in a reduction in power consumption.

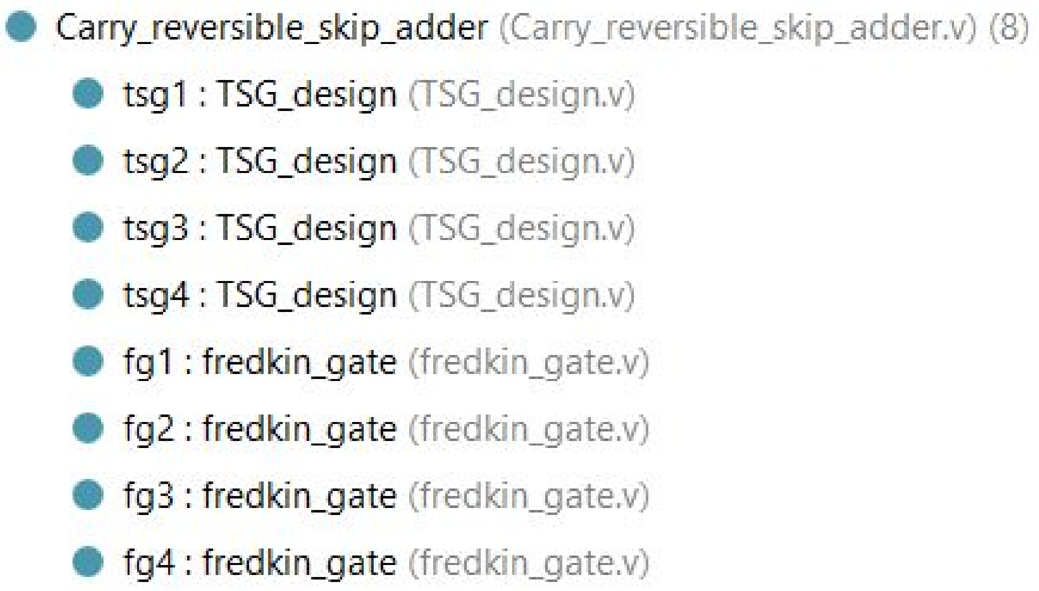
Reversible adders can be designed using a variety of reversible logic gates, such as Feynman gates, Toffoli gates, and Fredkin gates. In reversible adders, the carry bit is implemented using reversible logic, allowing the addition process to be performed without the loss of information. The carry bit is an essential component of adders and determines the accuracy of the addition operation.

Advantages of reversible adders include reduced power consumption, improved efficiency, and the potential for reduced heat generation. However, reversible adders have some disadvantages as well, including the higher gate count and the increased complexity of the design. Additionally, reversible logic gates require more area to implement than traditional irreversible logic gates, which can limit their applicability in some cases.

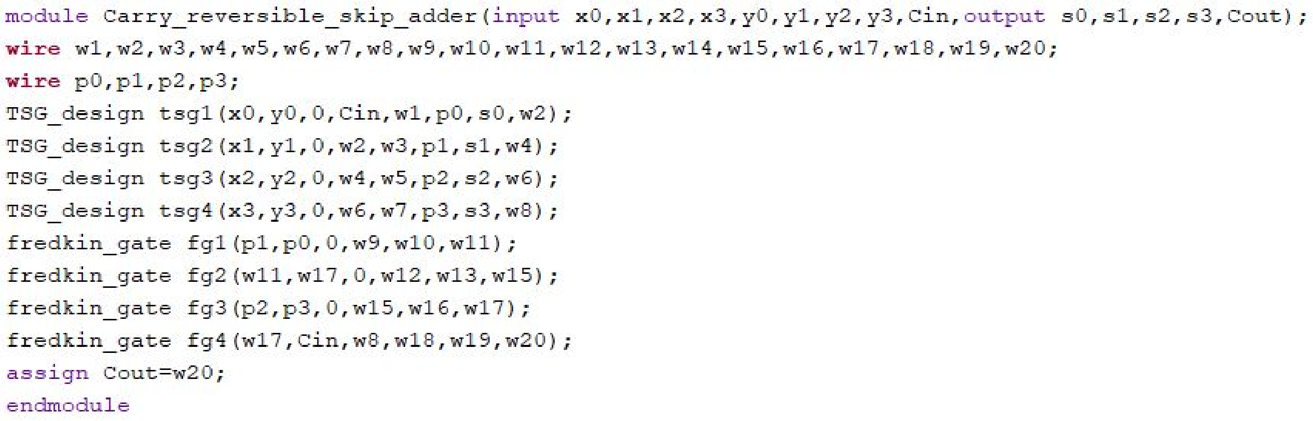
In conclusion, reversible adders offer several advantages over traditional irreversible adders, including reduced power consumption and improved efficiency. However, the design of reversible adders can be more complex, and their implementation may require more area. Further research is needed to explore the full potential of reversible adders in digital circuits and to optimize their performance for specific applications.

a. Reversible Carry Skip Adder A Reversible Carry Skip Adder (RCSA) is a type of digital circuit used for adding two or more binary numbers together. Unlike traditional adders, the RCSA is designed to minimize the number of carry propagation stages, which helps to reduce the delay and power consumption of the circuit. The RCSA achieves this by skipping over groups of bits that do not require a carry, using a carry-skip logic.

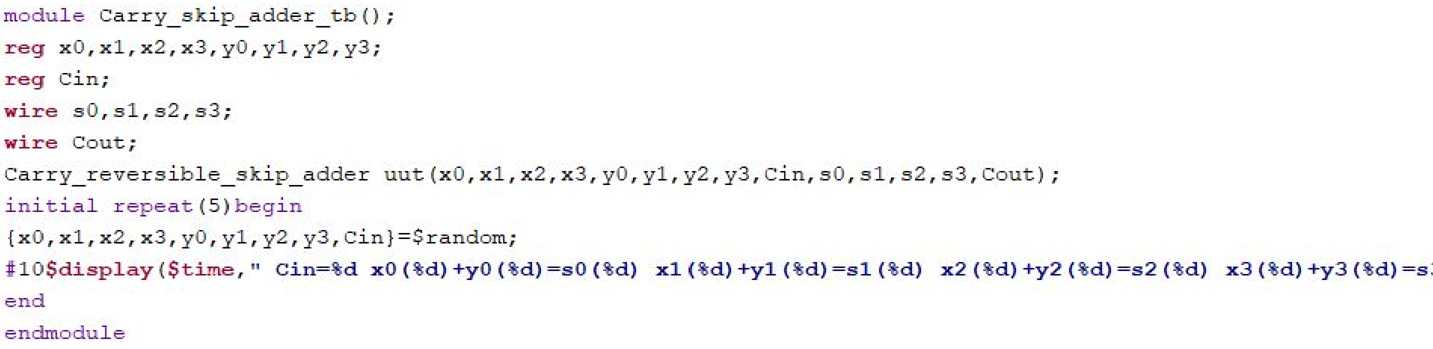
The RCSA is a reversible circuit, which means that it can be run in reverse to perform subtraction as well as addition. This is a useful feature for certain applications in quantum computing and cryptography, where reversible circuits are often preferred because they allow for more efficient computation and better security. Overall, the RCSA is a powerful and efficient tool for performing binary addition and subtraction, particularly in applications where low delay and power consumption are critical.



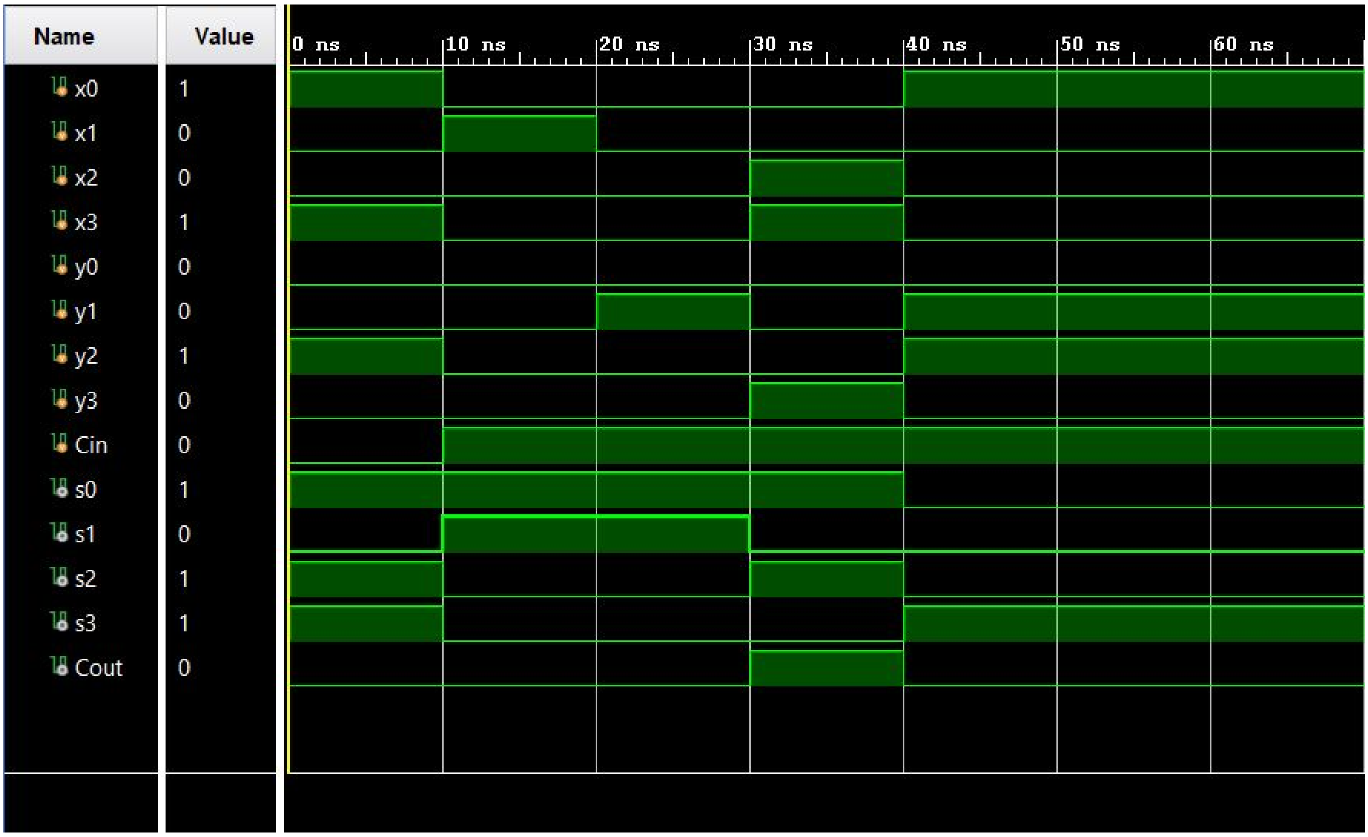
- ● Carry_reversible_skip_adder (Carry_reversible_skip_adder.v) (8)
  - ● tsg1 : TSG_design (TSG_design.v)
  - ● tsg2 : TSG_design (TSG_design.v)
  - ● tsg3 : TSG_design (TSG_design.v)
  - ● tsg4 : TSG_design (TSG_design.v)
  - ● fg1 : fredkin_gate (fredkin_gate.v)
  - ● fg2 : fredkin_gate (fredkin_gate.v)
  - ● fg3 : fredkin_gate (fredkin_gate.v)
  - ● fg4 : fredkin_gate (fredkin_gate.v)

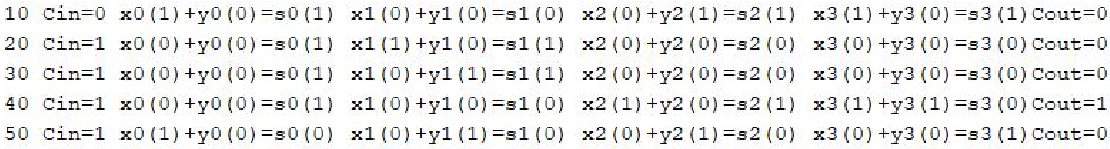


```
module Carry_reversible_skip_adder(input x0,x1,x2,x3,y0,y1,y2,y3,Cin,output s0,s1,s2,s3,Cout);
wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15,w16,w17,w18,w19,w20;
wire p0,p1,p2,p3;
TSG_design tsg1(x0,y0,0,Cin,w1,p0,s0,w2);
TSG_design tsg2(x1,y1,0,w2,w3,p1,s1,w4);
TSG_design tsg3(x2,y2,0,w4,w5,p2,s2,w6);
TSG_design tsg4(x3,y3,0,w6,w7,p3,s3,w8);
fredkin_gate fg1(p1,p0,0,w9,w10,w11);
fredkin_gate fg2(w11,w17,0,w12,w13,w15);
fredkin_gate fg3(p2,p3,0,w15,w16,w17);
fredkin_gate fg4(w17,Cin,w8,w18,w19,w20);
assign Cout=w20;
endmodule
```



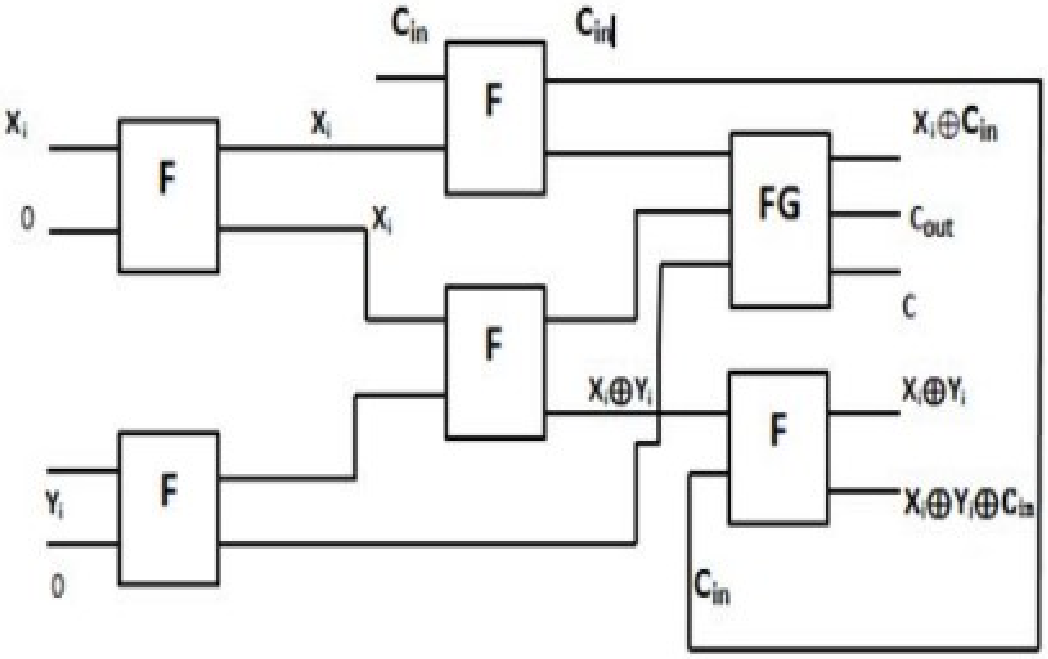
```
module Carry_skip_adder_tb();
reg x0,x1,x2,x3,y0,y1,y2,y3;
reg Cin;
wire s0,s1,s2,s3;
wire Cout;
Carry_reversible_skip_adder uut(x0,x1,x2,x3,y0,y1,y2,y3,Cin,s0,s1,s2,s3,Cout);
initial repeat(5)begin
{x0,x1,x2,x3,y0,y1,y2,y3,Cin}=$random;
#10$display($time," Cin=%d x0(%d)+y0(%d)=s0(%d)  x1(%d)+y1(%d)=s1(%d)  x2(%d)+y2(%d)=s2(%d)  x3(%d)+y3(%d)=s
end
endmodule
```





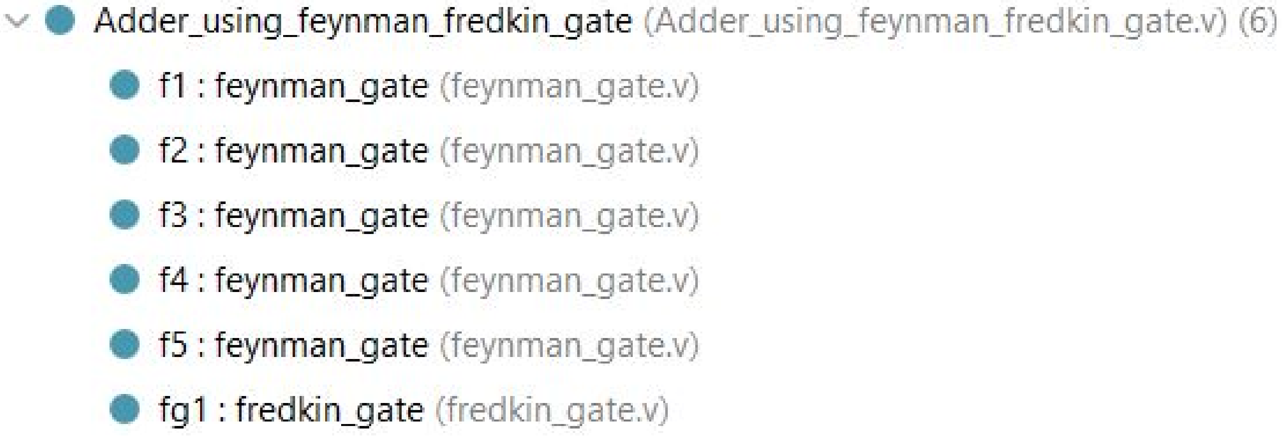
```
10 Cin=0  x0(1)+y0(0)=s0(1)  x1(0)+y1(0)=s1(0)  x2(0)+y2(1)=s2(1)  x3(1)+y3(0)=s3(1)Cout=0
20 Cin=1  x0(0)+y0(0)=s0(1)  x1(1)+y1(0)=s1(1)  x2(0)+y2(0)=s2(0)  x3(0)+y3(0)=s3(0)Cout=0
30 Cin=1  x0(0)+y0(0)=s0(1)  x1(0)+y1(1)=s1(1)  x2(0)+y2(0)=s2(0)  x3(0)+y3(0)=s3(0)Cout=0
40 Cin=1  x0(0)+y0(0)=s0(1)  x1(0)+y1(0)=s1(0)  x2(1)+y2(0)=s2(1)  x3(1)+y3(1)=s3(0)Cout=1
50 Cin=1  x0(1)+y0(0)=s0(0)  x1(0)+y1(1)=s1(0)  x2(0)+y2(1)=s2(0)  x3(0)+y3(0)=s3(1)Cout=0
```

b. Implementation of reversible adder using feynman and fredkin gate : Model-1

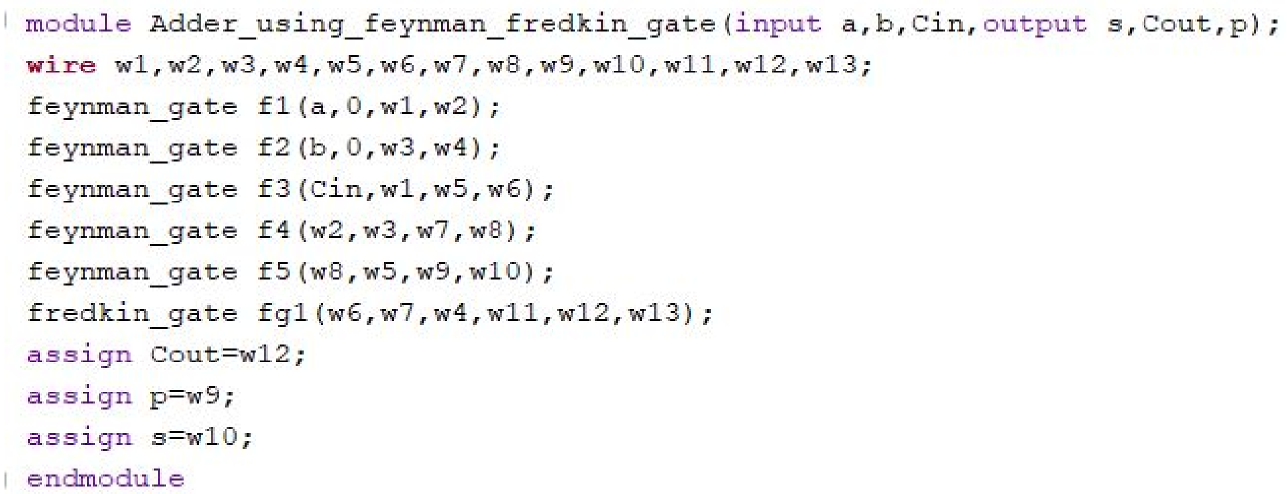


The adder Design as shown in Fig, consist of 1 Fredkin gate and 5 Feynman gates. Two out of five Feynman gates are used to generate fan-out signal remaining three Feynman gates are used to generate carry propagate signal .Fredkin gate is used for making either Xi or Yi as carry out signal on the basis of signal Xi xor Cin.

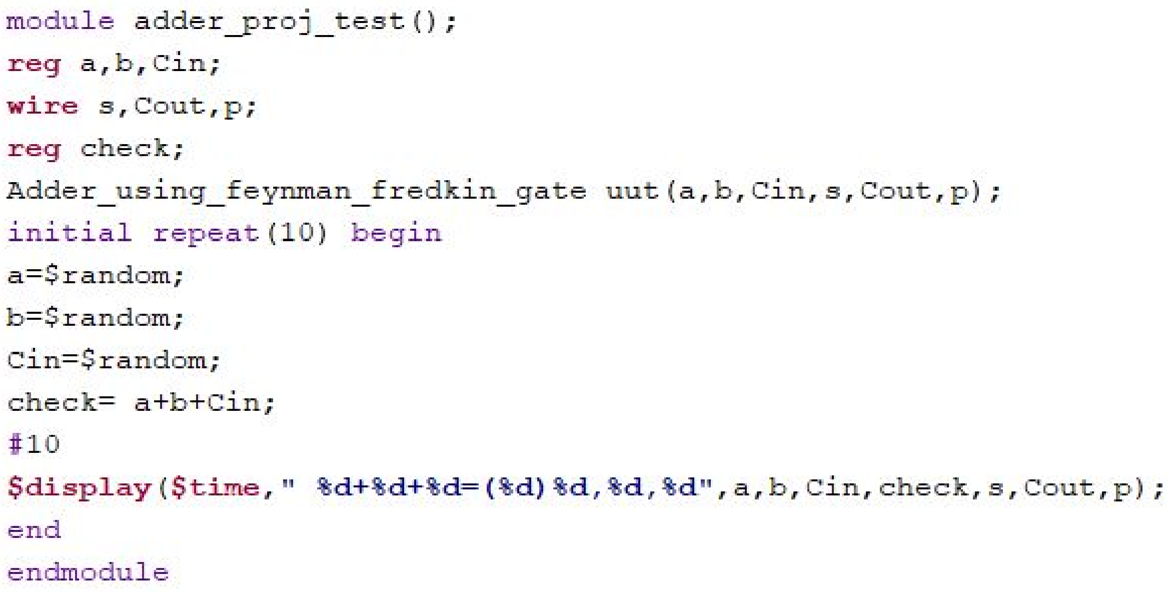
The adder shown in Fig. 9 has quantum cost of 10, 2 constant inputs and 2 garbage outputs. Delay in sum signal generation is 3F and delay in carry output signal generation is 1FG+2F. Carry propagate signal is generated after the delay of 3F. In design I Cout signal and Sum signal are generated simultaneously.



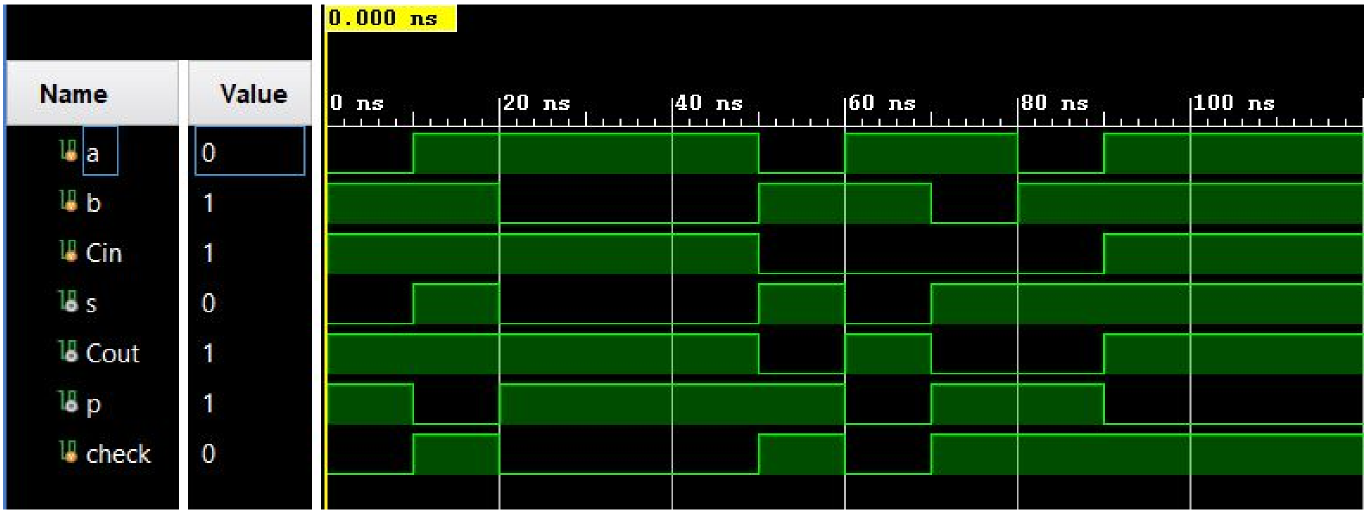
```
⌄ ● Adder_using_feynman_fredkin_gate (Adder_using_feynman_fredkin_gate.v) (6)
    ● f1 : feynman_gate (feynman_gate.v)
    ● f2 : feynman_gate (feynman_gate.v)
    ● f3 : feynman_gate (feynman_gate.v)
    ● f4 : feynman_gate (feynman_gate.v)
    ● f5 : feynman_gate (feynman_gate.v)
    ● fg1 : fredkin_gate (fredkin_gate.v)
```



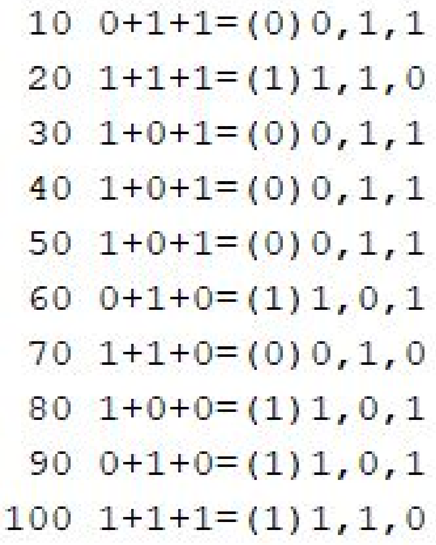
```
module Adder_using_feynman_fredkin_gate(input a,b,Cin,output s,Cout,p);
wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13;
feynman_gate f1(a,0,w1,w2);
feynman_gate f2(b,0,w3,w4);
feynman_gate f3(Cin,w1,w5,w6);
feynman_gate f4(w2,w3,w7,w8);
feynman_gate f5(w8,w5,w9,w10);
fredkin_gate fg1(w6,w7,w4,w11,w12,w13);
assign Cout=w12;
assign p=w9;
assign s=w10;
endmodule
```



```
module adder_proj_test();
reg a,b,Cin;
wire s,Cout,p;
reg check;
Adder_using_feynman_fredkin_gate uut(a,b,Cin,s,Cout,p);
initial repeat(10) begin
a=$random;
b=$random;
Cin=$random;
check= a+b+Cin;
#10
$display($time," %d+%d+%d=(%d)%d,%d,%d",a,b,Cin,check,s,Cout,p);
end
endmodule
```

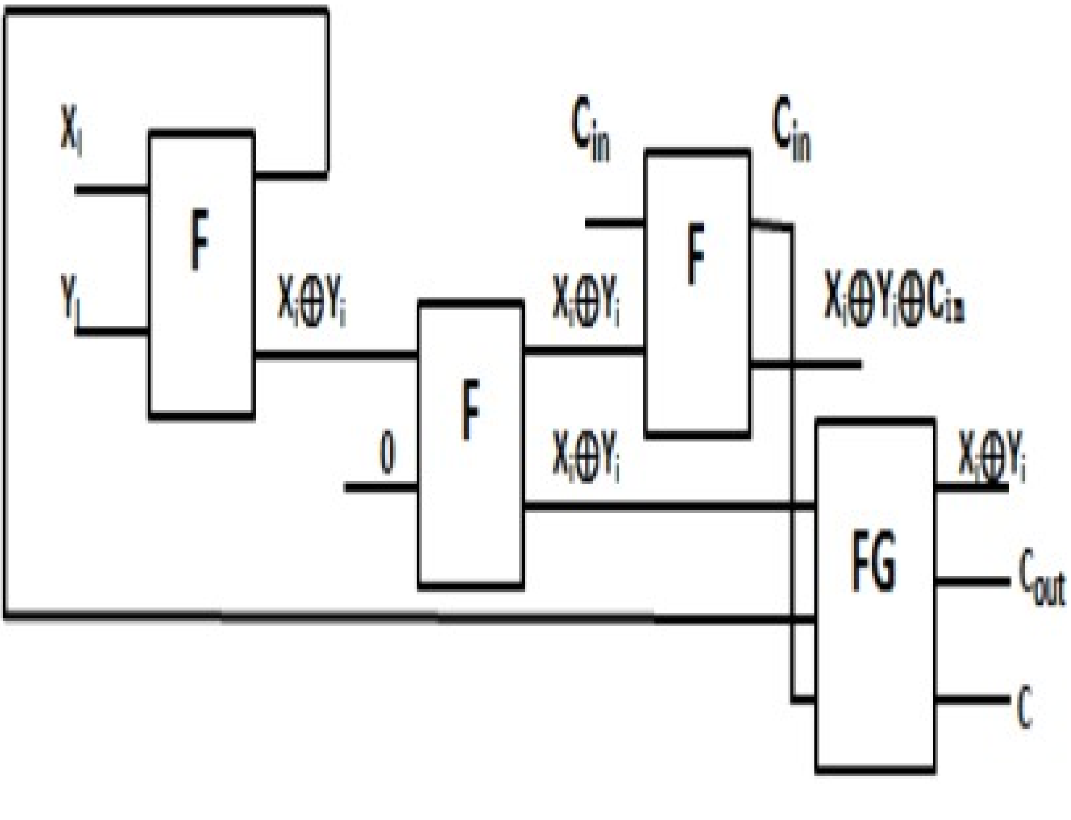


Name	Value
a	0
b	1
Cin	1
s	0
Cout	1
p	1
check	0



```
 10 0+1+1=(0)0,1,1
 20 1+1+1=(1)1,1,0
 30 1+0+1=(0)0,1,1
 40 1+0+1=(0)0,1,1
 50 1+0+1=(0)0,1,1
 60 0+1+0=(1)1,0,1
 70 1+1+0=(0)0,1,0
 80 1+0+0=(1)1,0,1
 90 0+1+0=(1)1,0,1
100 1+1+1=(1)1,1,0
```

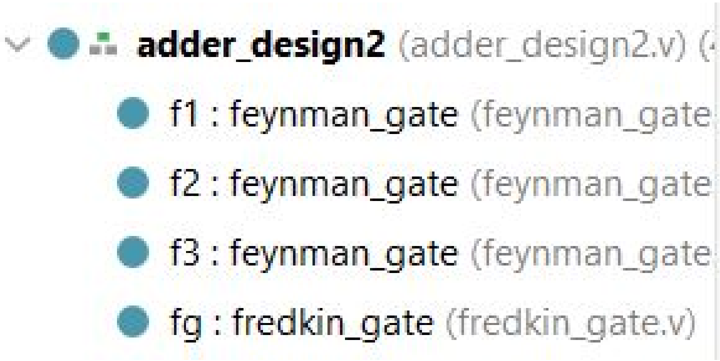
c. Implementation of reversible adder using feynman and fredkin gate : Model-2



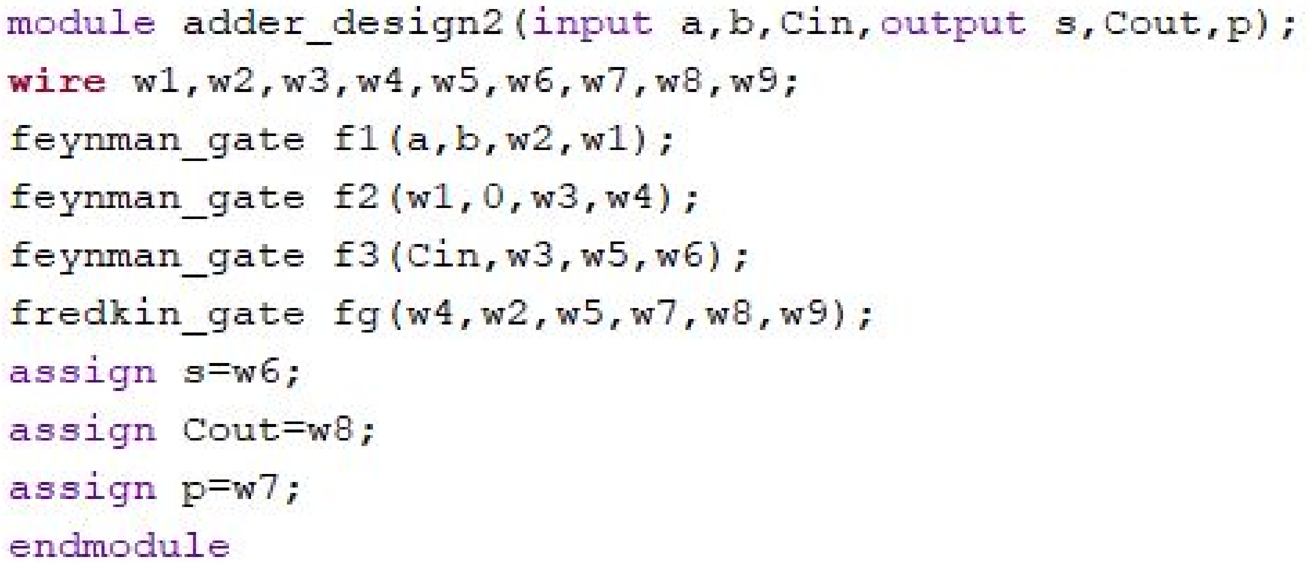
The proposed adder design II, consist of 1 Fredkin gate and 3 Feynman gates. One out of three Feynman gates are used to generate fanout signal remaining two Feynman gates are used to generate carry propogate signal. Fredkin gate is used to select either Xi or Cin as carry out signal on the basis of Xi xor Yi signal.

This Adder shown in Fig.10 has quantum cost of 8, one constant input and one garbage output. Delay in sum signal generation is 3F and delay in carry output signal generation is 1FG+2F.

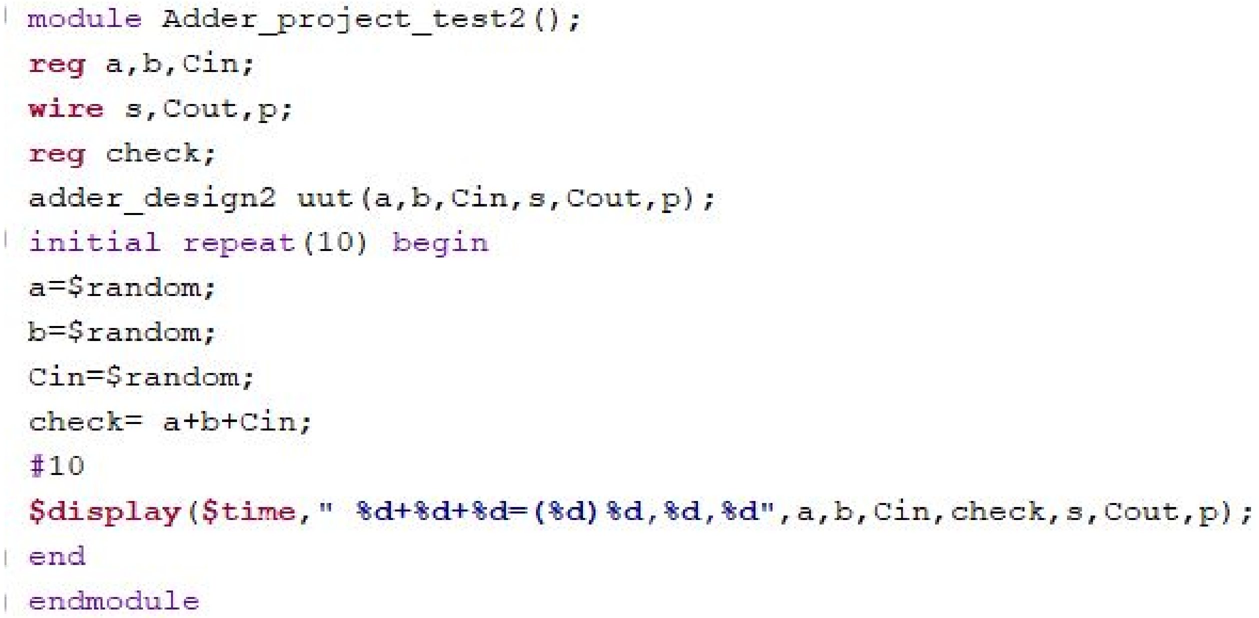
Carry propagate signal is generated after the delay of 2F. In design II Cout signal is generated after Sum signal.



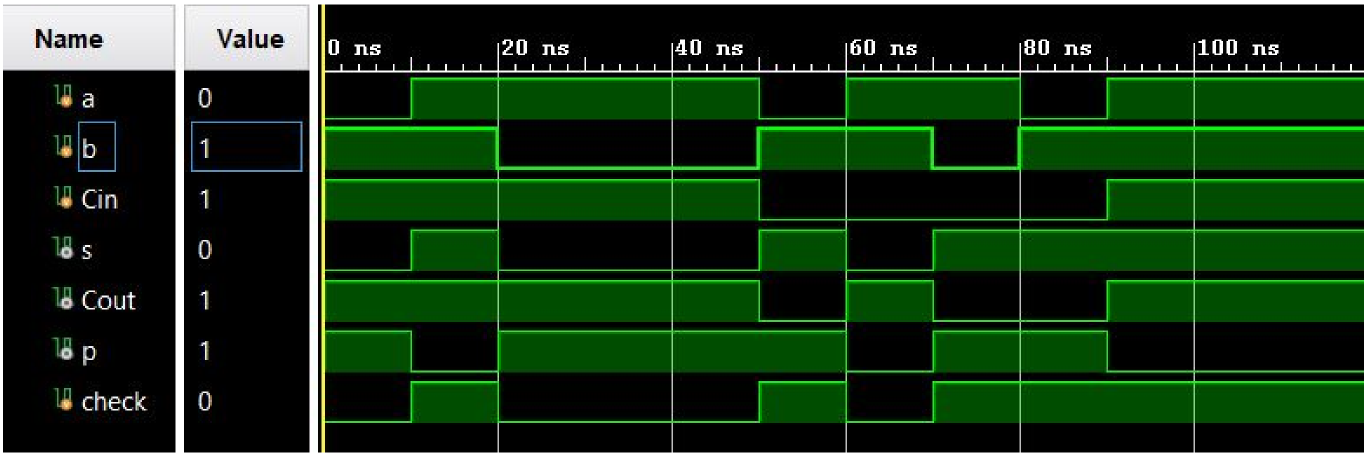
```
adder_design2 (adder_design2.v) (
    f1 : feynman_gate (feynman_gate
    f2 : feynman_gate (feynman_gate
    f3 : feynman_gate (feynman_gate
    fg : fredkin_gate (fredkin_gate.v)
```



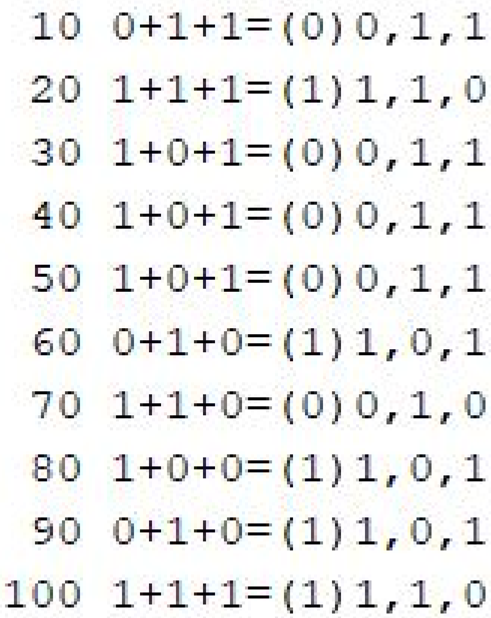
```
module adder_design2(input a,b,Cin,output s,Cout,p);
wire w1,w2,w3,w4,w5,w6,w7,w8,w9;
feynman_gate f1(a,b,w2,w1);
feynman_gate f2(w1,0,w3,w4);
feynman_gate f3(Cin,w3,w5,w6);
fredkin_gate fg(w4,w2,w5,w7,w8,w9);
assign s=w6;
assign Cout=w8;
assign p=w7;
endmodule
```



```
module Adder_project_test2();
reg a,b,Cin;
wire s,Cout,p;
reg check;
adder_design2 uut(a,b,Cin,s,Cout,p);
initial repeat(10) begin
a=$random;
b=$random;
Cin=$random;
check= a+b+Cin;
#10
$display($time," %d+%d+%d=(%d)%d,%d,%d",a,b,Cin,check,s,Cout,p);
end
endmodule
```



Name	Value
a	0
b	1
Cin	1
s	0
Cout	1
p	1
check	0



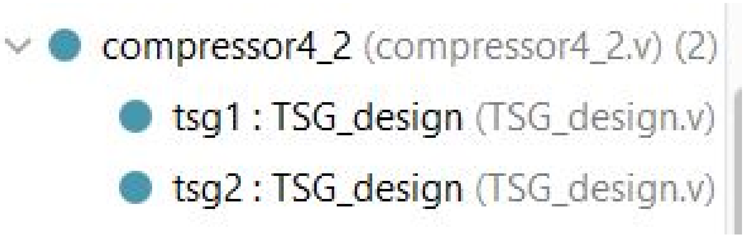
```
 10  0+1+1=(0)0,1,1
 20  1+1+1=(1)1,1,0
 30  1+0+1=(0)0,1,1
 40  1+0+1=(0)0,1,1
 50  1+0+1=(0)0,1,1
 60  0+1+0=(1)1,0,1
 70  1+1+0=(0)0,1,0
 80  1+0+0=(1)1,0,1
 90  0+1+0=(1)1,0,1
100  1+1+1=(1)1,1,0
```

2. COMPRESSORS BASED ON REVERSIBLE LOGIC Compressors based on reversible logic are a type of digital circuit that is designed to compress binary data by reducing the number of bits required to represent it. These circuits use reversible logic gates, which allow them to perform operations that can be reversed without any loss of information.

One of the most common types of compressors based on reversible logic is the carry-select adder (CSA). The CSA works by splitting the input data into two groups, and then performing separate additions on each group. The results of these additions are then combined using a multiplexer, which selects the correct output based on the value of a carry bit.

Another type of compressor based on reversible logic is the carry-lookahead adder (CLA). The CLA works by precomputing the carry bits for each bit position, using a set of logic gates that can be implemented using reversible logic. By precomputing the carry bits in this way, the CLA is able to perform additions more quickly than a traditional adder, while also consuming less power.

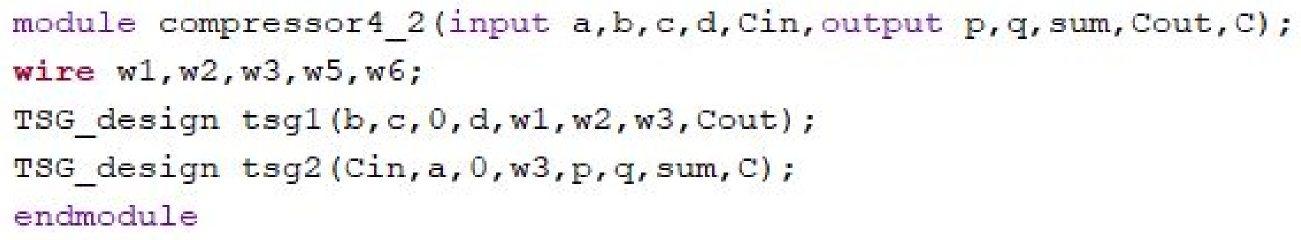
Compressors based on reversible logic are becoming increasingly important in the field of quantum computing, where efficient data compression is essential due to the limited number of qubits available. They are also being used in other areas of computing, such as cryptography and signal processing, where efficient data compression is required for low-latency operation. Overall, compressors based on reversible logic are a powerful tool for reducing the size of binary data, while also reducing the delay and power consumption of digital circuits.



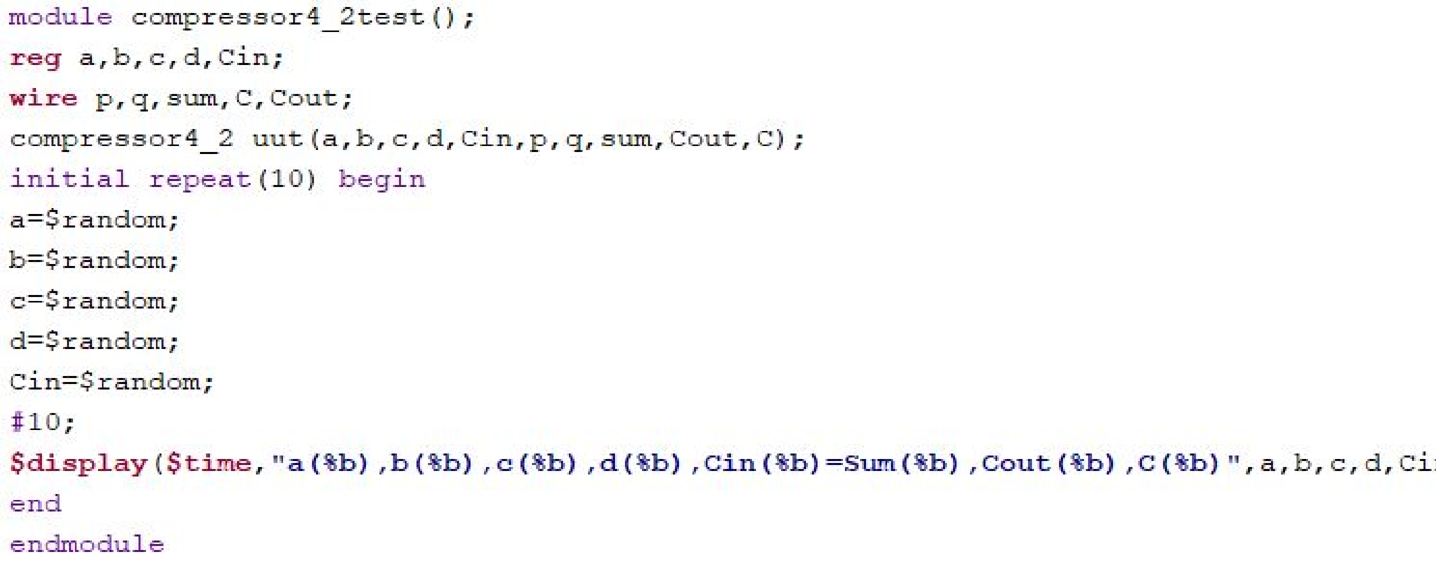
compressor4_2 (compressor4_2.v) (2)

tsg1 : TSG_design (TSG_design.v)

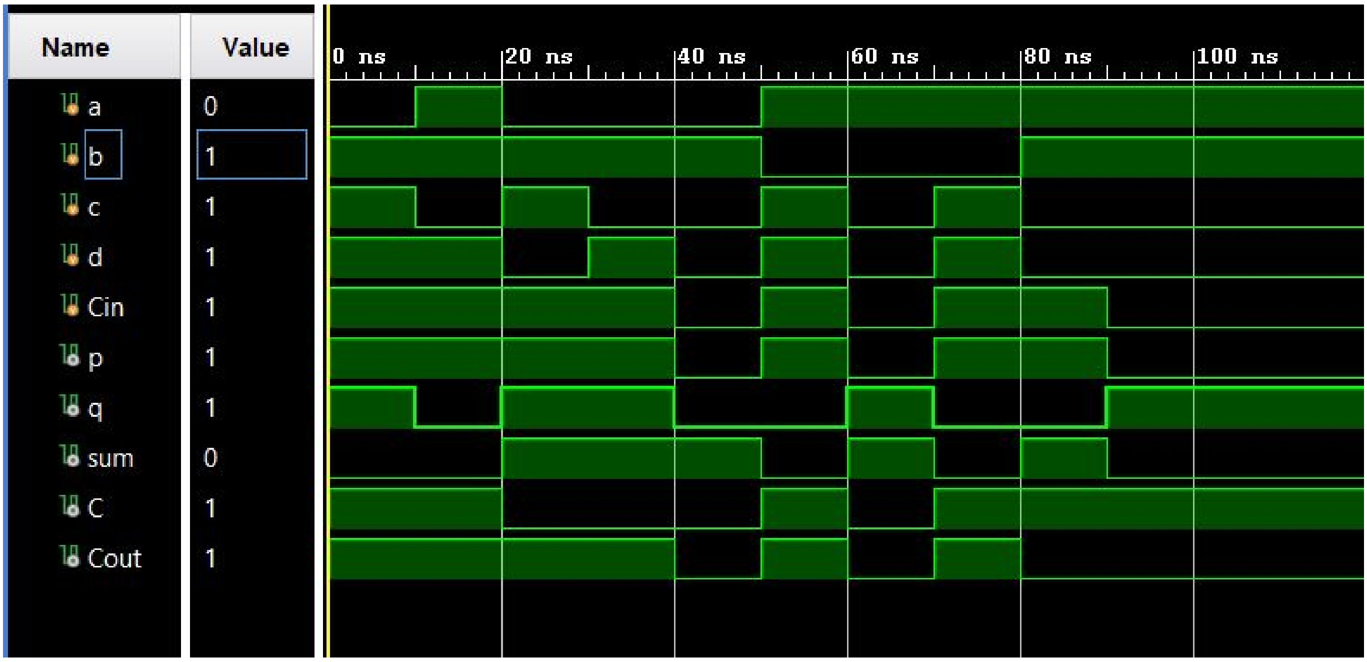
tsg2 : TSG_design (TSG_design.v)

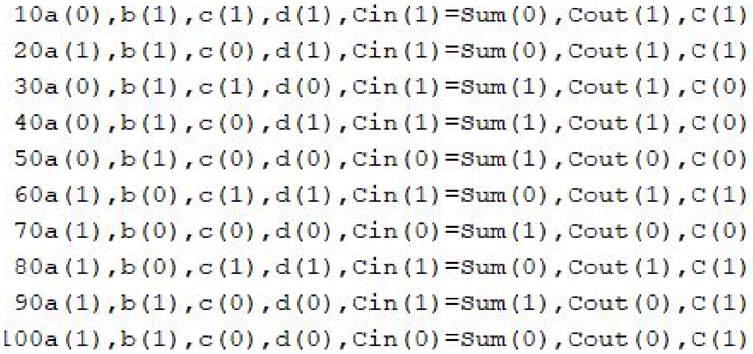


```
module compressor4_2(input a,b,c,d,Cin,output p,q,sum,Cout,C);
wire w1,w2,w3,w5,w6;
TSG_design tsg1(b,c,0,d,w1,w2,w3,Cout);
TSG_design tsg2(Cin,a,0,w3,p,q,sum,C);
endmodule
```



```
module compressor4_2test();
reg a,b,c,d,Cin;
wire p,q,sum,C,Cout;
compressor4_2 uut(a,b,c,d,Cin,p,q,sum,Cout,C);
initial repeat(10) begin
a=$random;
b=$random;
c=$random;
d=$random;
Cin=$random;
#10;
$display($time,"a(%b),b(%b),c(%b),d(%b),Cin(%b)=Sum(%b),Cout(%b),C(%b)",a,b,c,d,Ci
end
endmodule
```





```
10a(0),b(1),c(1),d(1),Cin(1)=Sum(0),Cout(1),C(1)
20a(1),b(1),c(0),d(1),Cin(1)=Sum(0),Cout(1),C(1)
30a(0),b(1),c(1),d(0),Cin(1)=Sum(1),Cout(1),C(0)
40a(0),b(1),c(0),d(1),Cin(1)=Sum(1),Cout(1),C(0)
50a(0),b(1),c(0),d(0),Cin(0)=Sum(1),Cout(0),C(0)
60a(1),b(0),c(1),d(1),Cin(1)=Sum(0),Cout(1),C(1)
70a(1),b(0),c(0),d(0),Cin(0)=Sum(1),Cout(0),C(0)
80a(1),b(0),c(1),d(1),Cin(1)=Sum(0),Cout(1),C(1)
90a(1),b(1),c(0),d(0),Cin(1)=Sum(1),Cout(0),C(1)
100a(1),b(1),c(0),d(0),Cin(0)=Sum(0),Cout(0),C(1)
```

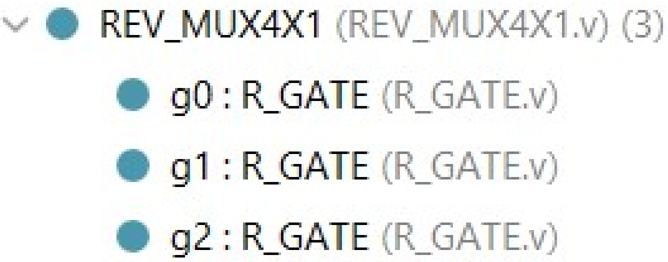
3. 4:1 MULTIPLEXER BASED ON REVERSIBLE LOGIC A 4:1 multiplexer based on reversible logic is a digital circuit that allows a single output to select one of four possible input values. In a traditional 4:1 multiplexer, this operation is typically performed using a set of logic gates, such as AND and OR gates. However, in a reversible 4:1 multiplexer, the operation is performed using reversible logic gates, which allows the circuit to be run in reverse without losing any information.

One common implementation of a reversible 4:1 multiplexer is based on the Fredkin gate, which is a three-qubit gate that can perform reversible logic operations. To construct a 4:1 multiplexer using Fredkin gates, Fredkin gates are connected together in a specific configuration, with each gate selecting one of the four input values based on the values of two control bits.

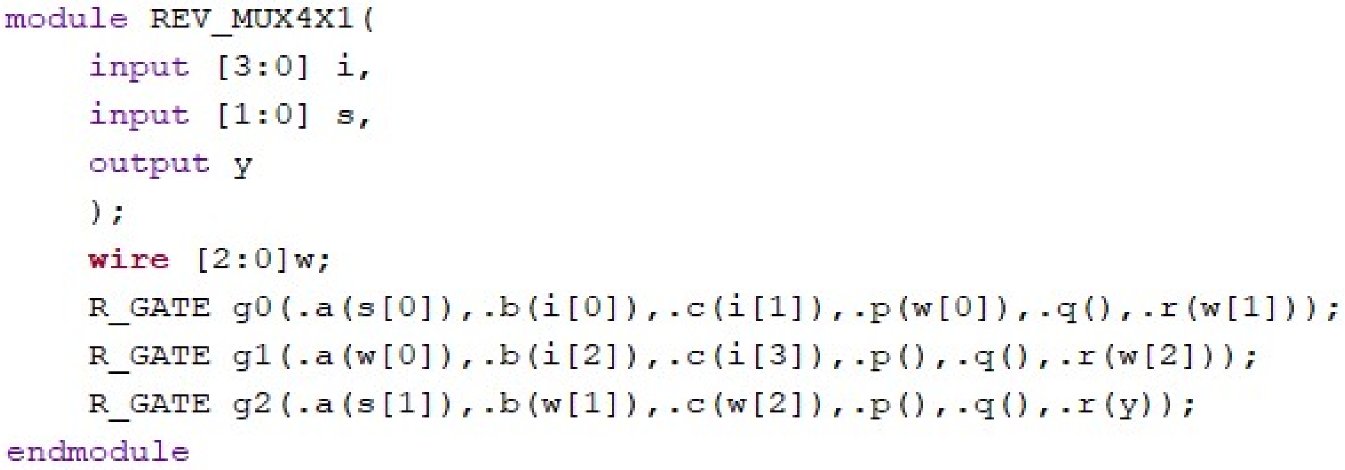
Another approach to implementing a reversible 4:1 multiplexer is based on the Peres gate, which is a two-qubit gate that can perform reversible logic operations. To construct a 4:1 multiplexer using Peres gates, two Peres gates are connected together in a specific configuration, with each gate selecting one of two possible input values based on the value of a control bit.

The advantage of using reversible logic gates to implement a 4:1 multiplexer is that it allows the circuit to be run in reverse without losing any information. This is important in applications such as quantum computing, where reversible circuits are required for efficient computation. Additionally, reversible circuits have the potential to reduce the power consumption and delay of digital circuits, which is an important consideration in many applications.

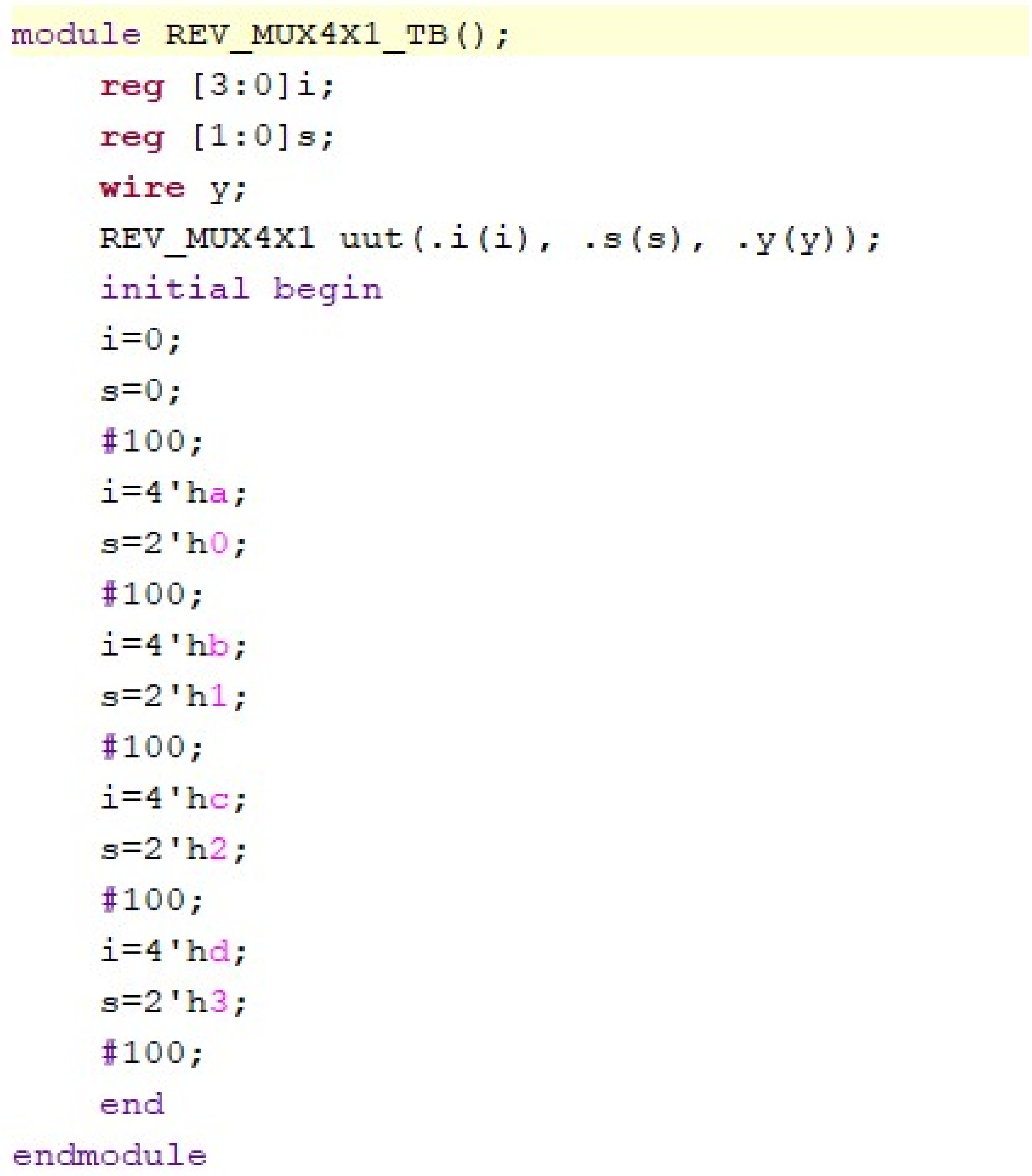
In summary, a 4:1 multiplexer based on reversible logic is a powerful tool for selecting one of four input values, while also allowing the circuit to be run in reverse without losing any information. This makes it a valuable component in many digital circuits, particularly in the field of quantum computing.



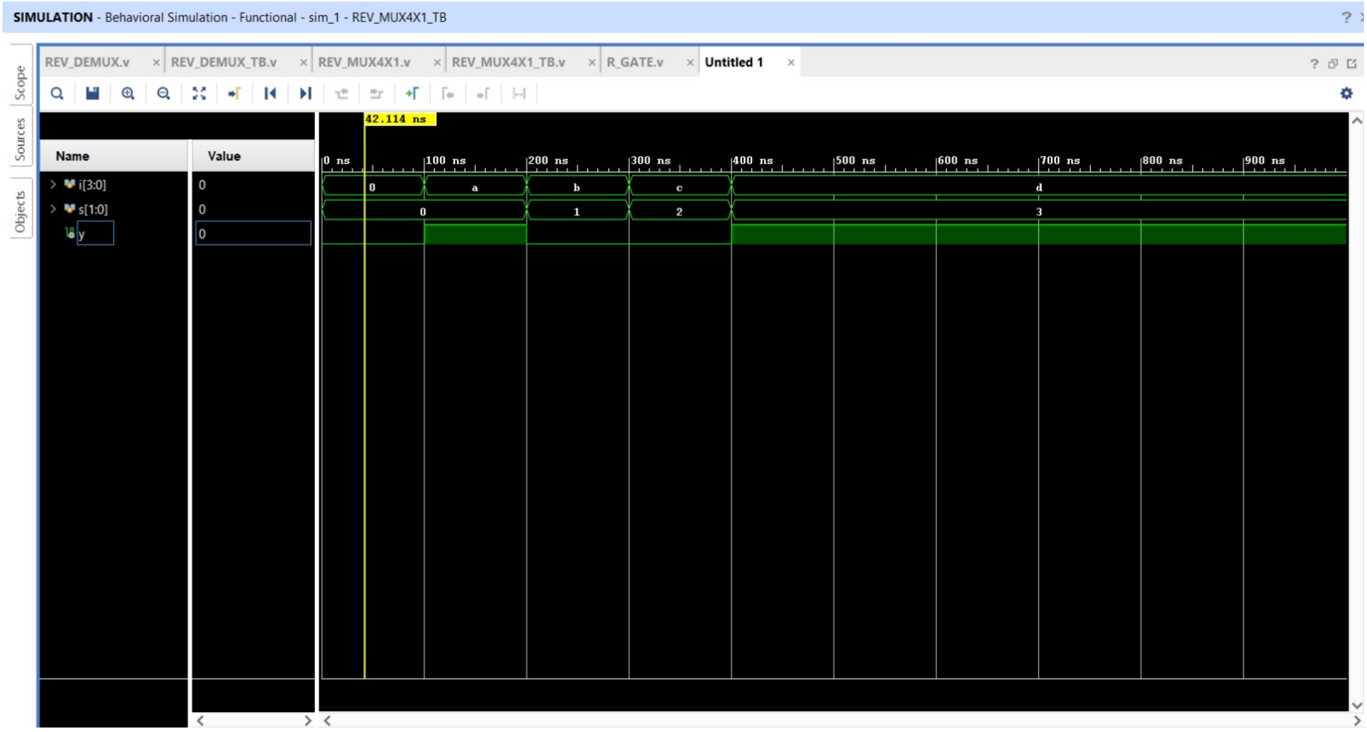
● REV_MUX4X1 (REV_MUX4X1.v) (3)
● g0 : R_GATE (R_GATE.v)
● g1 : R_GATE (R_GATE.v)
● g2 : R_GATE (R_GATE.v)



```
module REV_MUX4X1(
    input [3:0] i,
    input [1:0] s,
    output y
    );
    wire [2:0]w;
    R_GATE g0(.a(s[0]),.b(i[0]),.c(i[1]),.p(w[0]),.q(),.r(w[1]));
    R_GATE g1(.a(w[0]),.b(i[2]),.c(i[3]),.p(),.q(),.r(w[2]));
    R_GATE g2(.a(s[1]),.b(w[1]),.c(w[2]),.p(),.q(),.r(y));
endmodule
```



```verilog
module REV_MUX4X1_TB();
    reg [3:0]i;
    reg [1:0]s;
    wire y;
    REV_MUX4X1 uut(.i(i), .s(s), .y(y));
    initial begin
    i=0;
    s=0;
    #100;
    i=4'ha;
    s=2'h0;
    #100;
    i=4'hb;
    s=2'h1;
    #100;
    i=4'hc;
    s=2'h2;
    #100;
    i=4'hd;
    s=2'h3;
    #100;
    end
endmodule
```



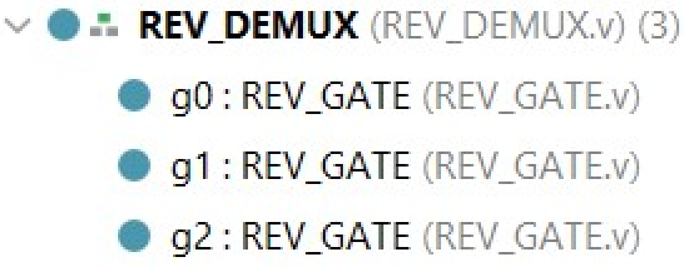
4. 1:4 DEMULTIPLEXER BASED ON REVERSIBLE LOGIC A 1:4 demultiplexer based on reversible logic is a digital circuit that takes a single input and directs it to one of four possible output channels. In a traditional 1:4 demultiplexer, this operation is typically performed using a set of logic gates, such as AND and NOT gates. However, in a reversible 1:4 demultiplexer, the operation is performed using reversible logic gates, which allows the circuit to be run in reverse without losing any information.

One common implementation of a reversible 1:4 demultiplexer is based on the Fredkin gate, which is a three-qubit gate that can perform reversible logic operations. To construct a 1:4 demultiplexer using Fredkin gates, four Fredkin gates are connected together in a specific configuration, with each gate directing the input to one of the four output channels based on the values of two control bits.

Another approach to implementing a reversible 1:4 demultiplexer is based on the controlled swap gate, which is a two-qubit gate that can perform reversible logic operations. To construct a 1:4 demultiplexer using controlled swap gates, two controlled swap gates are connected together in a specific configuration, with each gate directing the input to one of two possible output channels based on the value of a control bit.

The advantage of using reversible logic gates to implement a 1:4 demultiplexer is that it allows the circuit to be run in reverse without losing any information. This is important in applications such as quantum computing, where reversible circuits are required for efficient computation. Additionally, reversible circuits have the potential to reduce the power consumption and delay of digital circuits, which is an important consideration in many applications.

In summary, a 1:4 demultiplexer based on reversible logic is a powerful tool for directing a single input to one of four possible output channels, while also allowing the circuit to be run in reverse without losing any information. This makes it a valuable component in many digital circuits, particularly in the field of quantum computing.

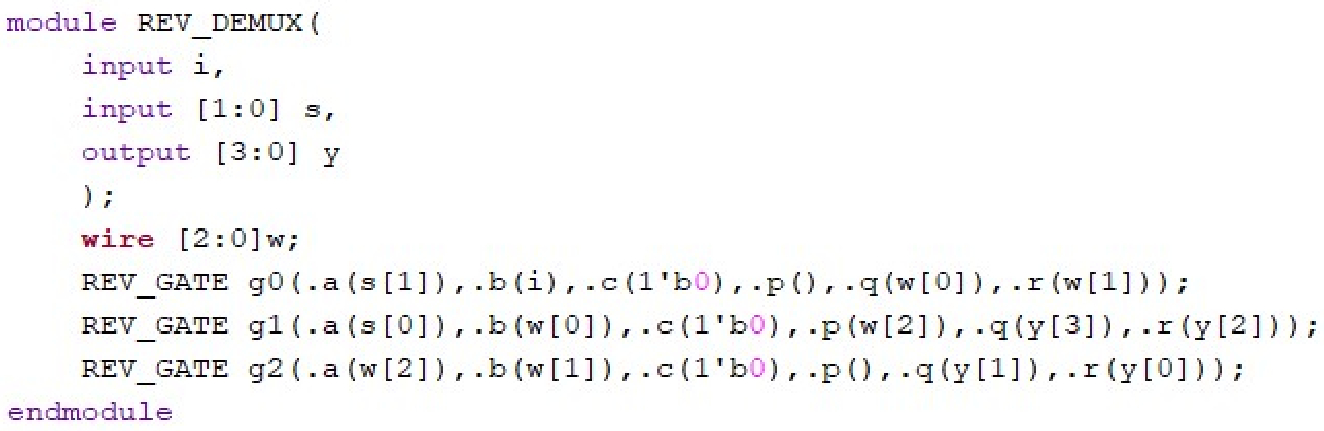


REV_DEMUX (REV_DEMUX.v) (3)

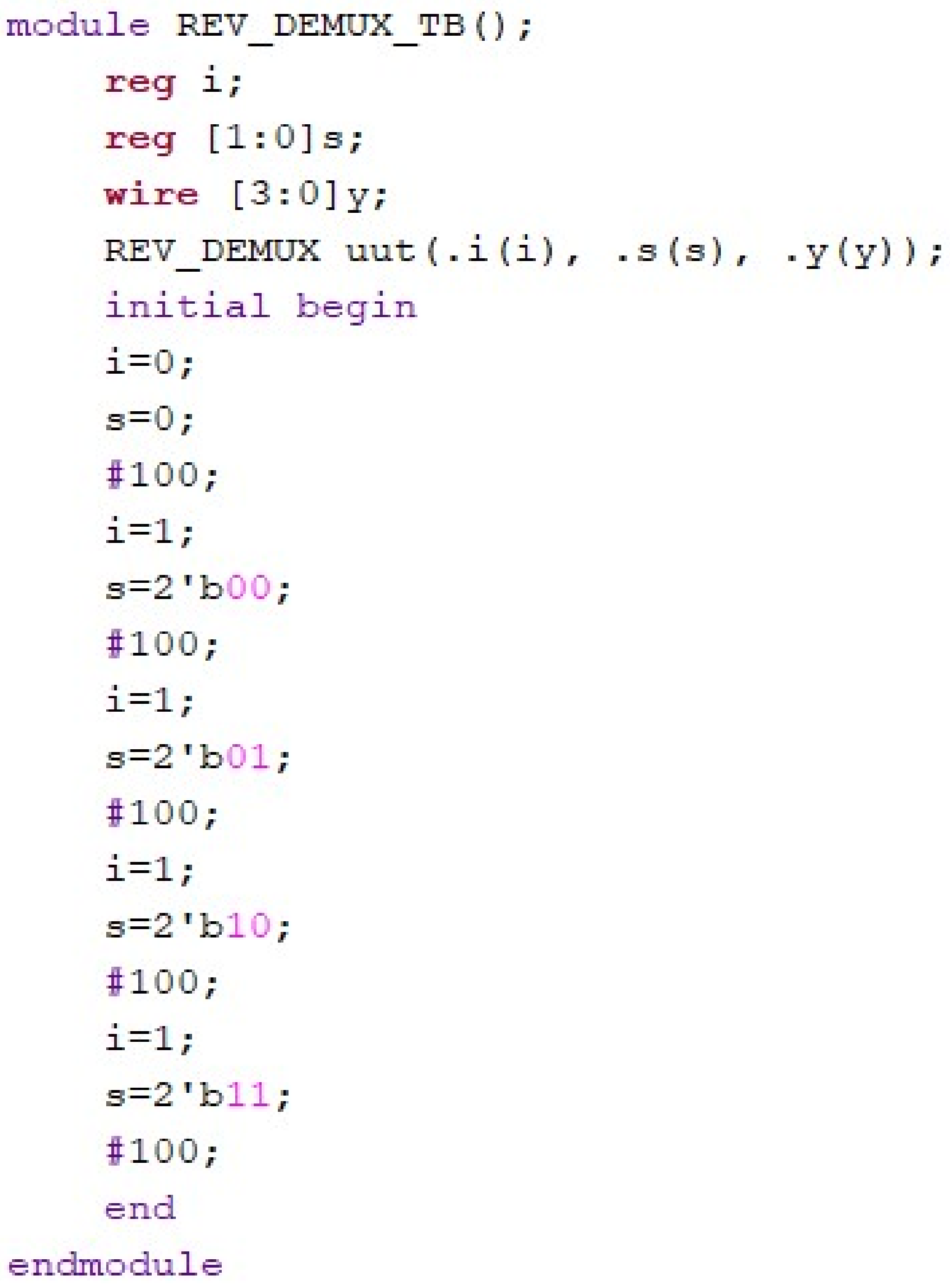
● g0 : REV_GATE (REV_GATE.v)

● g1 : REV_GATE (REV_GATE.v)

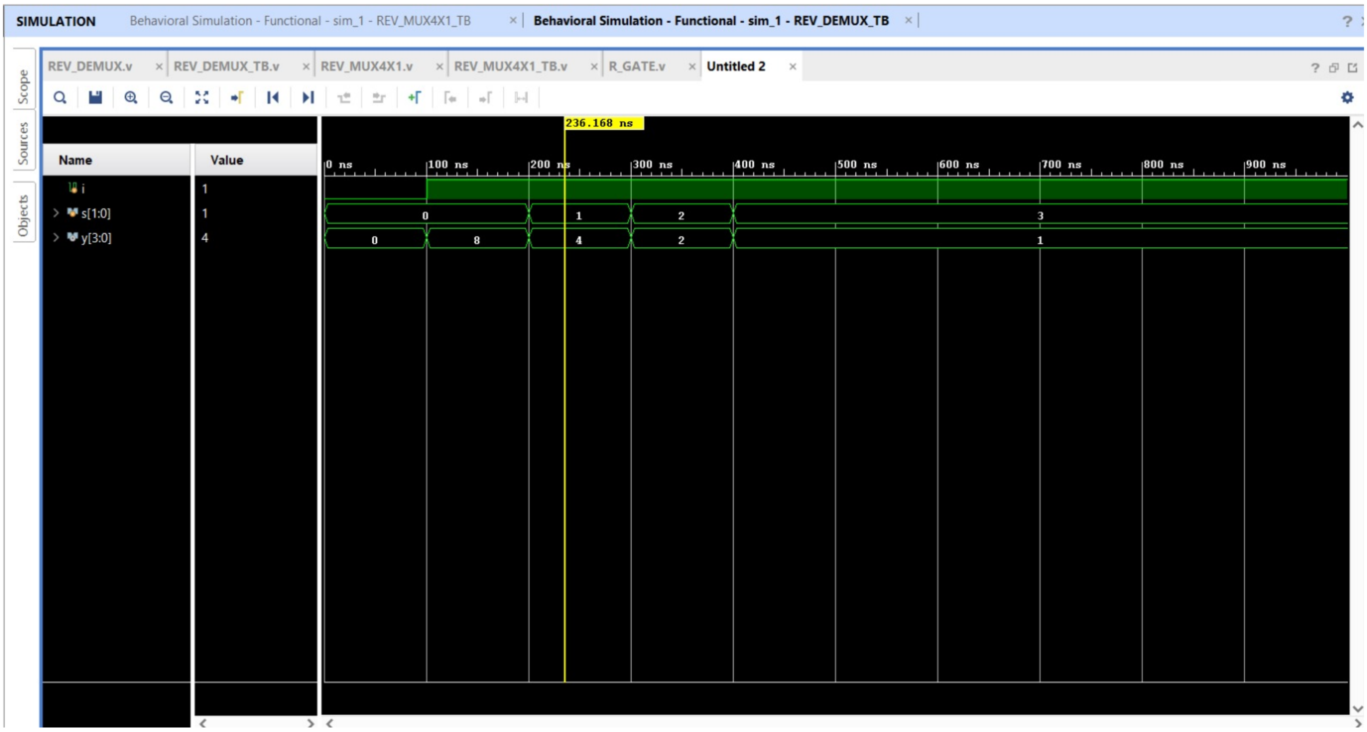
● g2 : REV_GATE (REV_GATE.v)



```
module REV_DEMUX(
    input i,
    input [1:0] s,
    output [3:0] y
    );
    wire [2:0]w;
    REV_GATE g0(.a(s[1]),.b(i),.c(1'b0),.p(),.q(w[0]),.r(w[1]));
    REV_GATE g1(.a(s[0]),.b(w[0]),.c(1'b0),.p(w[2]),.q(y[3]),.r(y[2]));
    REV_GATE g2(.a(w[2]),.b(w[1]),.c(1'b0),.p(),.q(y[1]),.r(y[0]));
endmodule
```



```
module REV_DEMUX_TB();
    reg i;
    reg [1:0]s;
    wire [3:0]y;
    REV_DEMUX uut(.i(i), .s(s), .y(y));
    initial begin
    i=0;
    s=0;
    #100;
    i=1;
    s=2'b00;
    #100;
    i=1;
    s=2'b01;
    #100;
    i=1;
    s=2'b10;
    #100;
    i=1;
    s=2'b11;
    #100;
    end
endmodule
```



CONCLUSION

In conclusion, this project aimed to explore the use of reversible logic in the implementation of digital logic circuits. The project involved the design and simulation of various digital logic circuits using reversible logic gates, including adders, compressors, demultiplexers, and multiplexers.

The results of this project demonstrate that reversible logic gates have the potential to reduce power consumption and improve the efficiency of digital logic circuits. Reversible logic gates also offer the ability to perform computations without the loss of information, which is essential for many applications.

The project also highlighted some of the challenges and limitations of reversible logic, such as the increased complexity of design and the higher gate count required for implementation. Further research is needed to optimize the performance of reversible logic circuits for specific applications and to explore the full potential of reversible logic in digital circuits.

Overall, this project contributes to the growing body of research on reversible logic and its potential applications in digital logic circuits. The use of reversible logic has the potential to make significant contributions to the development of energy-efficient and high-performance digital systems in the future.

FUTURE SCOPE

The implementation of digital logic circuits using reversible logic has enormous potential for future research and development. The following are some of the possible future scope of this project:

1. Optimization of reversible logic circuits: There is a need for further research to optimize the performance of reversible logic circuits, specifically to reduce the number of gates required and improve their speed of operation.

2. Reversible computing architectures: Reversible computing architectures can offer significant energy savings in applications where data is frequently accessed and modified. The development of such architectures can be an exciting future scope for this project.

3. Integration with existing systems: The integration of reversible logic circuits with existing systems and architectures can offer significant performance improvements. The development of methods for seamless integration of reversible logic circuits with traditional irreversible circuits can be an exciting future direction for this project.

4. Quantum computing: Reversible logic is a key component of quantum computing, and further research in this area can be an exciting future scope for this project.

5. Design automation tools: There is a need for developing design automation tools that can simplify the design process of reversible logic circuits. The development of such tools can improve the adoption of reversible logic circuits in practical applications.

In summary, there are many exciting future directions for research and development in the field of reversible logic circuits. This project has laid the foundation for further exploration and development of reversible logic circuits, and the future scope of this project can contribute significantly to the development of efficient and high-performance digital systems.

References

[1] R. Landauer, "Irreversibility and heat generation in the computing process," IBM Journal of Research and Development, vol. 5, no. 3, pp. 183-191, Jul. 1961.

[2] C. H. Bennett, "Logical reversibility of computation," IBM Journal of Research and Development, vol. 17, no. 6, pp. 525-532, Nov. 1973.

[3] M. Perkowski, R. Drechsler, A. Mishchenko, and O. Ciftcioglu, "On synthesis of reversible sequential machines," Proceedings of the 2002 IEEE International Symposium on Circuits and Systems, vol. 5, pp. V-729-V-732, May 2002.

[4] M. B. Tahoori and F. Lombardi, "Compressor Trees for High-Speed Arithmetic," Proceedings of the 36th Conference on Design Automation, pp. 539-544, Jun. 1999.

[5] A. Kumar, S. Mukhopadhyay, and B. Sen, "A novel high speed and low power reversible logic based 4:2 compressor," Proceedings of the 2012 IEEE International Conference on Advances in Computing, Communications and Informatics, pp. 499-502, Aug. 2012.

[6] M. Haghparast, M. A. Ebrahimi, and M. Khademolhosseini, "Design and implementation of a reversible multiplexer and demultiplexer," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, no. 5, pp. 6-10, May 2016.

[7] N. B. Karayiannis, C. Efstathiou, G. Nikiforos, and D. Soudris, ”Low-power reversible MUX and DEMUX designs for nanometer CMOS technology,” Microprocessors and Microsystems, vol. 36, no. 2-3, pp. 119-126, Mar. 2012.

[8] M. B. Tahoori and F. Lombardi, ”Reversible Logic Synthesis for Large Boolean Functions,” Proceedings of the 15th IEEE International Conference on Computer Design, pp. 501-506, Oct. 1997.

[9] M. A. Thornton and K. S. Brace, ”Reversible logic synthesis using a genetic algorithm,” Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol. 2, pp. 537-540, May 1998.

[10] H. Thapliyal and N. Ranganathan, ”Reversible logic synthesis using evolutionary algorithms,” International Journal of Electronics and Communications, vol. 66, no. 3, pp. 178-184, Mar. 2012.

[11] R. Wille, R. Drechsler, and G. W. Dueck, ”Reversible logic synthesis 101: Fundamentals, recent developments, and future directions,” Proceedings of the 2018 International Conference on ReConFigurable Computing and FPGAs, pp. 1-8, Dec. 2018.

[12] N. B. Karayiannis and C. Efstathiou, ”Reversible logic synthesis methodologies: A survey,” ACM Transactions on Design Automation of Electronic Systems, vol. 21, no. 3, article no. 44

[13] M. B. Srinivas, S. R. Pattanaik, and S. K. Satpathy, “Design of High-Speed Reversible Comparator Using New Reversible Gate,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 23, no. 5, pp. 939-947, May 2015.

[14] N. N. Jha, K. K. Singh, and S. C. De Sarkar, “Novel Reversible Gate Based Design of MUX and DEMUX,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 1, pp. 109-117, Jan. 2015.

[15] S. M. Shamsul Alam and M. M. Islam, “A Novel Design of Compressor and Decompressor Circuit Using Reversible Logic,” in Proceedings of the 2017 4th International Conference on Advances in Electrical Engineering (ICAEE), Dhaka, Bangladesh, 2017, pp. 122-125.

[16] M. Perkowski, “Reversible Logic Synthesis,” in Proceedings of the 1994 IEEE International Symposium on Multiple-Valued Logic (ISMVL), May 1994, pp. 162-169.

[17] G. N. Frederickson, “Conservative Logic,” Information and Control, vol. 28, no. 3, pp. 257-270, March 1975.

[18] E. Fredkin and T. Toffoli, “Conservative Logic,” International Journal of Theoretical Physics, vol. 21, no. 3/4, pp. 219-253, March/April 1982.

[19] R. P. Feynman, “Quantum mechanical computers,” Foundations of Physics, vol. 16, no. 6, pp. 507-531, June 1986.

[20] M. B. Srinivas and S. K. Satpathy, “Design of Reversible Sequential Circuits Using Reversible Decoder and D-Flip-Flop,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 10, pp. 2058-2066, Oct. 2014.

[21] M. B. Srinivas and S. K. Satpathy, “Design of Low-Power High-Speed 4-bit Reversible ALU,” Microelectronics Journal, vol. 45, no. 6, pp. 720-729, June 2014.

[22] S. M. Shamsul Alam and M. M. Islam, “Design of a Reversible Full Adder Circuit,” in Proceedings of the 2016 2nd International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2016, pp. 1-5.

[23] S. M. Shamsul Alam, M. M. Islam, and M. R. Hasan, “Design of a Reversible Full Subtractor Circuit,” in Proceedings of the 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Dhaka, Bangladesh, 2016, pp. 18-21.

[24] M. H. Anvarkhan and M. N. S. Swamy, “Quantum realization of reversible full adder,” Int. J. Quantum Inf., vol. 2, no. 4, pp. 257–267, 2004.

[25] M. Haghparast and H. Mohammadi, “A new reversible design of binary comparator,” J. Comput. Theor. Nanosci., vol. 11, no. 7, pp. 1988–1992, 2014.

[26] M. Perkowski and K. K. Bhattacharyya, “Realization of digital circuits with reversible logic,” Int. J. Theor. Phys., vol. 33, no. 12, pp. 2317–2348, 1994.

[27] S. Mandal, P. P. Sarkar, S. Sengupta, and A. Sarkar, “Design and analysis of a new reversible 4: 1 multiplexer using quantum-dot cellular automata,” J. Comput. Electron., vol. 16, no. 4, pp. 1379–1391, 2017.

[28] S. Sengupta and S. Mandal, “Design and analysis of a novel reversible fault-tolerant universal shift register,” J. Circuits, Syst. Comput., vol. 26, no. 11, pp. 1–24, 2017.

[29] A. R. Rezaei and A. Ejlali, “Design and implementation of a novel reversible ALU using GDI cell,” in 2012 20th Iranian Conference on Electrical Engineering, 2012, pp. 1578–1581.

[30] S. Sengupta and S. Mandal, “Design and analysis of a new reversible multiplier circuit using quantum-dot cellular automata,” Int. J. Electron. Lett., vol. 5, no. 3, pp. 219–227, 2017.

[31] F. F. Yuan, Z. Y. He, X. Y. Niu, and X. S. Zhang, “Design and optimization of reversible 8× 8 multiplier using modified reversible CNT gate,” J. Comput. Theor. Nanosci., vol. 12, no. 4, pp. 678–682, 2015.

[32] S. Sengupta, P. P. Sarkar, and S. Mandal, “Design and analysis of a novel reversible programmable logic array,” J. Electron. Test., vol. 34, no. 2, pp. 131–143, 2018.

[33] S. Sengupta and S. Mandal, “Design and analysis of a novel reversible binary to gray code converter,” in 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2017, pp. 1022–1026.