

Design of A Controlled Multi-Logic Function Generator and ALU by Using COG Reversible Logic Gates

**K.Rajasekhar 2.Devi Prasanna 3 Yamuna 4.Pujitha Sai 5.Pranay Kumar
6.Sanasi Naidu**

1 (Assistant Professor , ECE Department , Sri Sivani College of Engineering , Srikakulam ,
Andhra Pradesh , India)

2, 3,4,5 (B.tech Student , ECE Department , Sri Sivani College of Engineering , Srikakulam
, Andhra Pradesh , India)

ABSTRACT—

Reversible logic is one of the most essential issues at present time due to its power reduction capability in circuit designing. It finds application in various fields including quantum computing , optical computing , nano technology , computer graphic , cryptography . Dissipation of a significant amount of energy is achieved in the conventional digital circuits because bits of information are erased during the logic operations. Thus, if logic gates are designed in such a way that the information bits are not destroyed, then it is possible to reduce the power consumption dramatically. The information bits are not lost in case of a reversible computation. This has led to the development of reversible gates. The reversible circuits do not lose information and can generate unique outputs from the specified inputs. The main purposes of designing reversible logic are to decrease quantum cost and the number of garbage outputs. This paper represents the realization of different MULTIPLEXERS and a MULTI LOGIC FUNCTION GENERATOR circuit for generating multiple logical functions simultaneously using COG gates. And a CONTROLLED MULTI LOGIC FUNCTION GENERATOR circuit for generating any specified output in a controlled way. This paper also proposes the design of ALU which have better performance in terms of quantum cost. The proposed work leads to an improvement of 20% and 17% in terms of gate count and quantum cost. The simulation and verification of these are done in XILINX software.

INDEX TERMS : reversible gates, reversible multiplexer, irreversible multiplexer, garbage output, constant input, ALU.

I. INTRODUCTION

Energy dissipation is one of the major issues in present day technology . Energy dissipation due to information loss in high technology circuits and systems constructed using irreversible hardware was demonstrated by R. Landauer in the year 1960. According to Landauer principle, the loss of one bit of information lost, will dissipate $kT \ln 2$ joules of energy where, k is the Boltzmann's constant, T is the absolute temperature. In 1973, Bennett showed that in order to avoid $kT \ln 2$ joules of energy dissipation

in a circuit it must be built from reversible circuits. According to Moore's law the numbers of transistors will double every 18 months. Thus energy conservative devices are the need of the day. The amount of energy dissipated in a system bears a direct relationship to the number of bits erased during computation. Reversible circuits are those circuits that do not lose information. The current irreversible technologies will dissipate a lot of heat and can reduce the life of the circuit. The reversible logic operations do not erase (lose) information and dissipate less heat. Synthesis of reversible logic circuit differs from combinational one in many ways. Firstly, in reversible circuit there should be no fan-out, that is, each output will be used only once. Secondly for each input pattern there should be unique output pattern. Finally, the resulting circuit must be acyclic. Any reversible circuit design includes only the gates that are the number of gates, quantum cost and the number of garbage outputs.

II. NEED OF REVERSIBLE LOGIC GATES

Since, the present technologies are reaching their limits in terms of power and area, a new paradigm is needed. Reversible computing provides low power design. improves the performance of the system. It increases portability of the device by reducing the element size to atomic size. It is an emerging trend in nanotechnology and QCA domain. This clearly shows that the reversible logic is future in VLSI.

III. DEFINITIONS

1. Reversible logic function:

A reversible logic function is a function which maps each input vector to a unique output vector. A function is said reversible if, from its given output, it is always possible to determine back its input, because there is a one - to- one relationship between input and output states.

2. Reversible logic gate:

A reversible logic gate is a device which performs reversible computation maintaining one to one mapping between the inputs and outputs. If a reversible logic gate has N inputs, then to perfo

to one mapping, the number of outputs should also be N. Then this device may be called an NxN reversible logic gate whose inputs are denoted by $I_1 I_2 I_3 \dots I_N$ and the outputs are denoted by $O_1 O_2 O_3 \dots O_N$.

3. Garbage output:

These are the outputs that are not used in the synthesis of a function. These may appear to be redundant but are very essential to preserve the reversibility of a gate. It is denoted by GO.

4. Constant inputs:

These are the inputs that have to be maintained at either a constant 0 or at constant 1 in order to generate a given logical expression by using the reversible logic gates. It is abbreviated as CI.

5. Quantum cost:

This refers to the cost of the circuit in terms of the cost of a primitive gate. It is computed knowing the number of primitive reversible logic gates (1×1 or 2×2) required to realize the circuit. It is denoted as QC.

6. Gate count:

This refers to the number of gates that are required to implement a reversible logic circuit. It is denoted by GC. Another parameter that can be defined in relation to the gate count is the flexibility, which can be defined as the ability of a reversible logic gate in realizing more functions. Higher the flexibility of a gate, lesser is the number of gates that are needed to implement a given function.

7. Hardware complexity:

The hardware complexity [7] is measured by counting the number of EX-OR operations, number of AND operations and number of NOT operations. Let

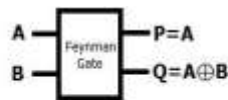
α = No. of EX-OR operations

β = No. of AND operations

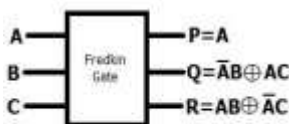
δ = No. of NOT operations

Then the total hardware complexity is given as sum of EX-OR, AND and NOT operations.

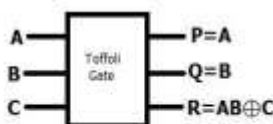
BASIC REVERSIBLE LOGIC GATES



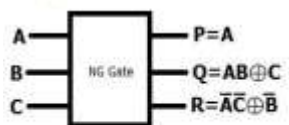
(a) Feynman Gate



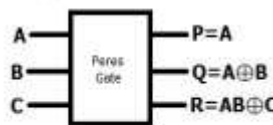
(b) Fredkin Gate



(c) Toffoli Gate



(d) New Gate



(e) Peres Gate

B. Reversible Logic Gates

The important basic reversible logic gate is Feynman gate which is the only 2×2 reversible gate used most popularly by the designers for fan-out purposes. There is also a double Feynman gate, Fredkin gate and Toffoli gate, New Gate, Peres gate, all of which can be used to realize important combinational functions and all are 3×3 reversible gates. Some basic reversible gates are shown in Fig. 1.

C. COG Reversible Logic Gates

A 3×3 reversible gate COG (Controlled Operation Gate) already had been proposed [14] shown in Fig. 2. The Truth table for the corresponding gate is shown in Table I also. The closer looking at the truth table reveals that the input pattern corresponding to a specific output pattern can be uniquely determined and thereby maintaining that there is a one-to-one correspondence between the input vectors and the output vectors. In this gate the input vector is given by $I_V = (A, B, C)$ and the corresponding output vector is $O_V = (P, Q, R)$.



Figure 2. COG reversible gate.

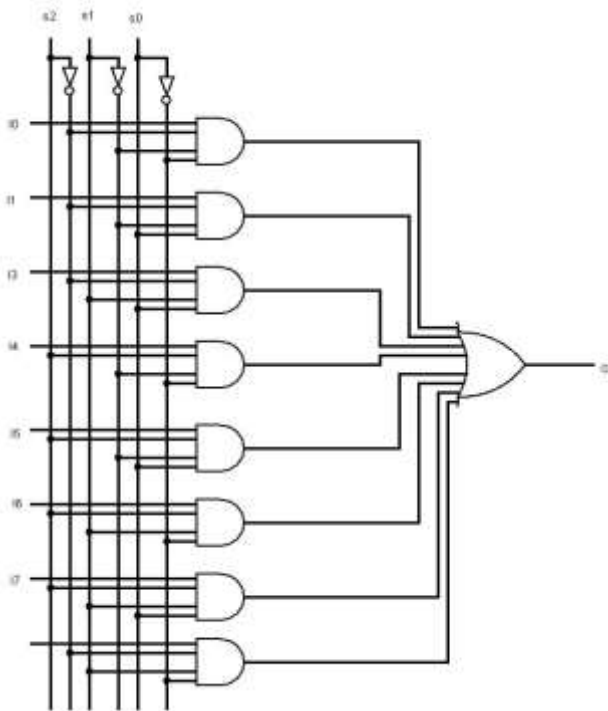
TABLE I. TRUTH TABLE OF COG GATE

Inputs			Outputs		
A	B	C	P	Q	R
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	0	1
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	0	1
1	1	0	1	0	0
1	1	1	1	1	1

III. CONVENTIONAL MULTIPLEXER

Multiplexer is a combinational circuit that selects binary information from one of the input lines and directs it to a single output line. Usually there are 2^n input lines and n selection lines whose bit combinations determine which input line is to be selected. It is also called a "Data Selector". Common sizes are 2:1, 4:1, 8:1, and 16:1. Since digital logic uses binary values, powers of 2 are used (4, 8, and 16) to maximally control a number of inputs for the given number of selector inputs. For 8:1mux, the number of selector inputs are 3 and inputs are 8 and output is 1. An 8 to 1 multiplexer of basic gate is shown below.

Irreversible 8 to 1 multiplexer



IV. REALIZATION OF MULTIPLEXERS BY COG GATE

Different multiplexers have been realized in this paper by using COG reversible gates.

Design of 2:1 Reversible Multiplexer Using COG Gate

we can design many combinational circuits by using the COG reversible gates, one of these circuit is multiplexer. Generally we use multiplexer for multiplexing one channel out of many channels. A 3x3 reversible COG gate is used in order to act as the 2:1 reversible multiplexer producing two garbage bits. The inputs are S_0 , I_0 and I_1 . Based on the selection input S_0 , the corresponding message bits are passed on to the output Y .

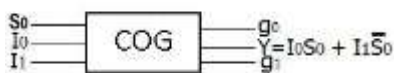


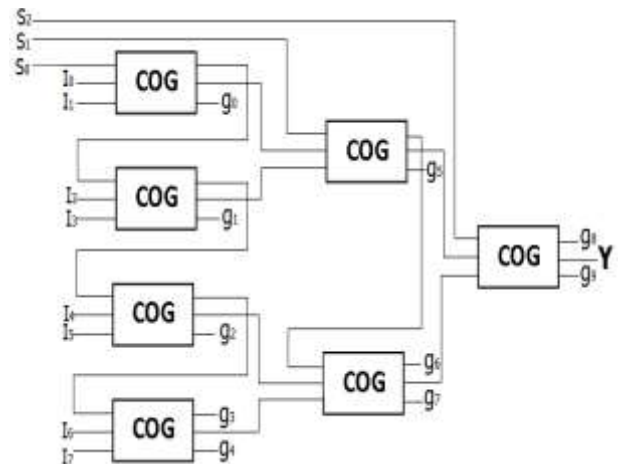
Figure . Reversible 2 to 1 multiplexer

Design of 8:1 Reversible Multiplexer Using COG Gate

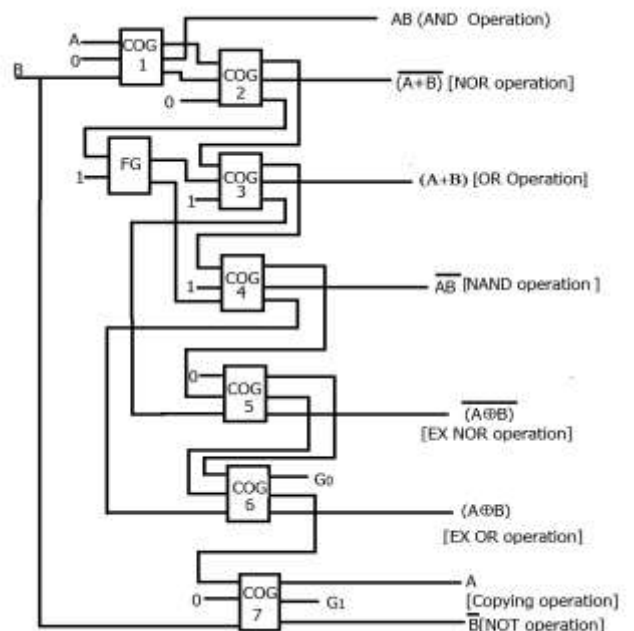
For the design of 8:1 multiplexer, we have used seven COG reversible gates. There are three selection lines S_3, S_2, S_1 which can be used for channelizing the required input lines. There are total eight input lines $I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7$. Out of which we will get any one at the output depending on the selection lines value. There are 10 garbage output.

In general designing of $2^n:1$ reversible multiplexer can be possible where n is 1, 2, 3... n . For $2^n:1$ reversible multiplexer (2^n-1) COG gates are required producing (2^n+n-1) number of garbage outputs.

Reversible 8 to 1 multiplexer



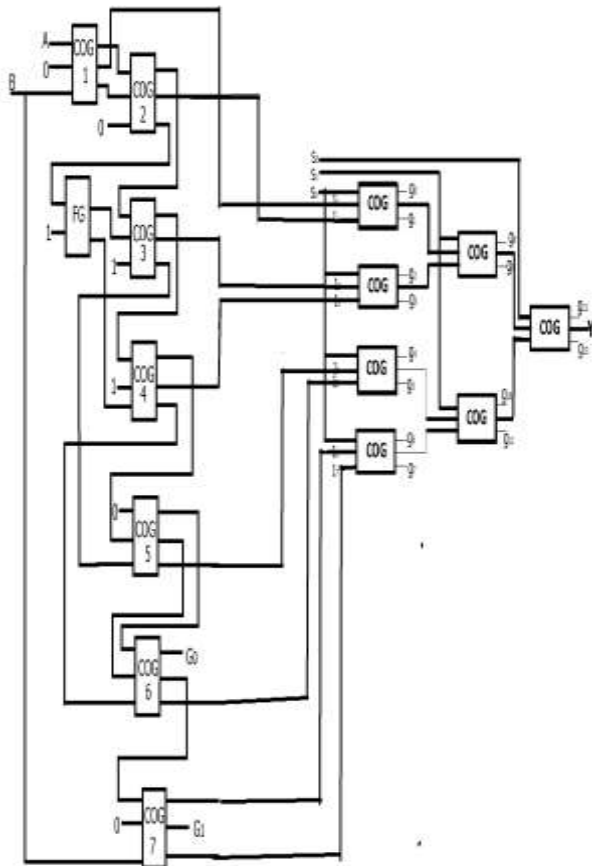
V. IMPLEMENTATION OF MULTI-LOGIC FUNCTION GENERATOR FOR GENERATING 8 FUNCTIONS



We have implemented a Multi-Logic Function Generator by using COG gates. For this we have used 7 COG reversible gates and 1 FEYMAN gate. There are eight operations at the output. In this way we can generate multiple logical functions from this function generator. If two variables A and B are taken then the functions which available at output are AND operation (AB) , NOR operation $(A+B)'$, OR operation $(A+B)$, NAND operation $(A.B)'$, EX-NOR operation $(AB+ A'B)'$,

EX-OR operation ($A'B+AB'$), Copying operation (A), NOT operation (B').

IV. Controlled Multi Logic Function Generator using 8:1 multiplexer by using COG gates



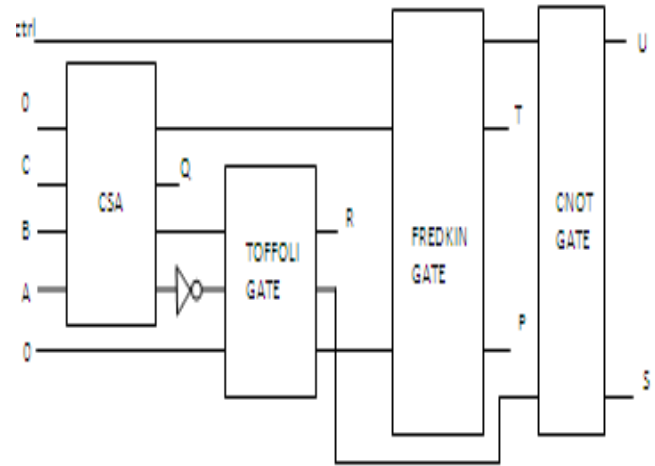
We have cascaded the function generator with the 8x1 multiplexer to get the controlled output. Depending upon the value of the selection lines we will get the logical function at the output of the multiplexer which has already been specified. So we can get the controlled behavior at the output of the circuit depending upon the values of S3, S2, and S1.

s3	s2	s1	Boolean function	Function Name	Output Y Of 8:1 mux
0	0	0	$(A.B)$	AND	I0
0	0	1	$(A+B)'$	NOR	I1
0	1	0	$(A+B)$	OR	I2
0	1	1	$(A.B)'$	NAND	I3
1	0	0	$(AB+A'B')$	EX NOR	I4
1	0	1	$(A'B+AB')$	EX OR	I5
1	1	0	A	COPYING	I6
1	1	1	B'	NOT	I7

V. Design of 1bit ALU using Reversible logic gates

The proposed 1-bit ALU is designed using adder

block, Toffoli gate, Fredkin gate, Cnot gate and Not gate. There are multiple techniques to perform addition operation such as carry save, carry skip, ripple carry, carry look ahead etc. In this approach carry save technique is applied.



This carry save adder (CSA) is comparatively lower in quantum cost and number of gates. This adder can be used to design an ALU with lower quantum cost, constant inputs and garbage output. The variable 'c' acts as an input as well as control and variable 'ctrl' is utilized as a control line.

ALU OPERATIONS :

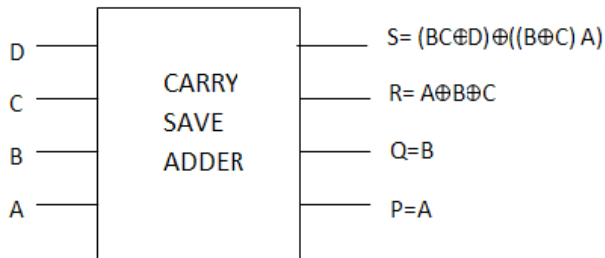
A. ARTHEMATICAL OPERATIONS

C	CNTRL	P	Q	R
0	0	Sum	Carry	-
0	1	Borrow	Difference	-
X	1	-	-	Buffer
X	0	-	-	complement

B. LOGICAL OPERATIONS

C	CNTRL	P	Q	R
0	0	A and B	A xor B	-
1	0	A or B	A xnor B	-

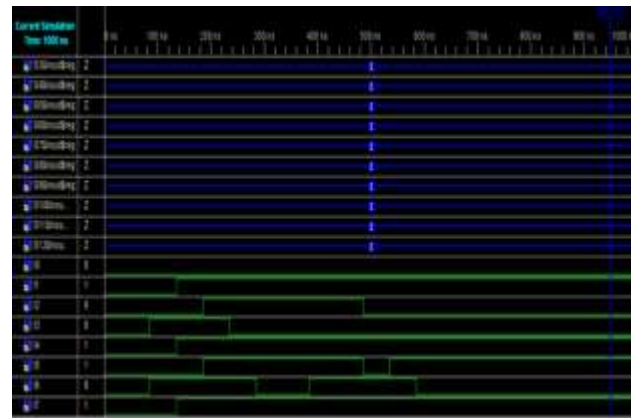
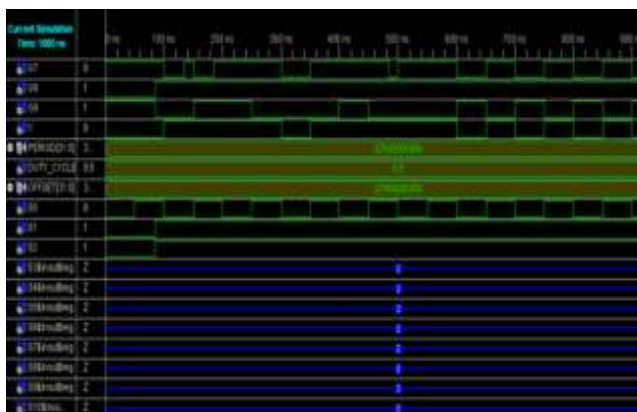
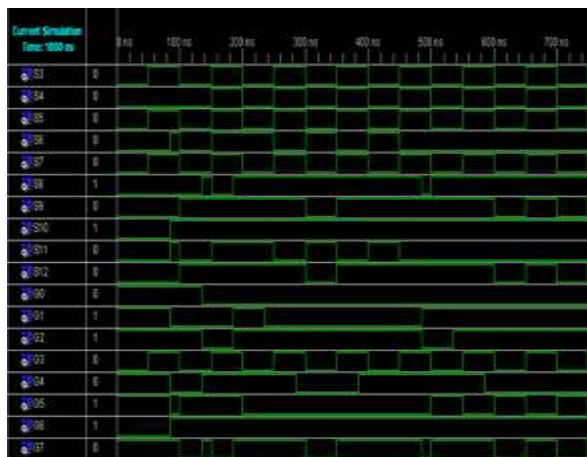
CARRY SAVE ADDER



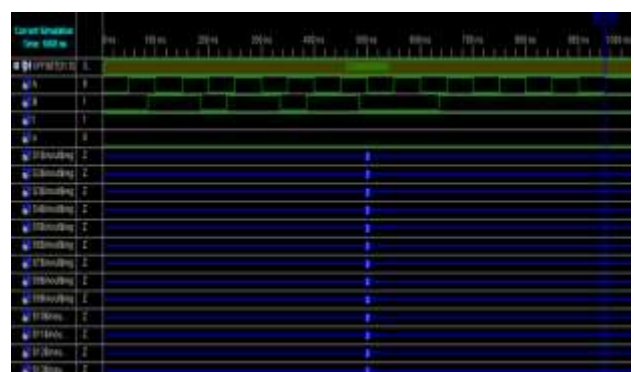
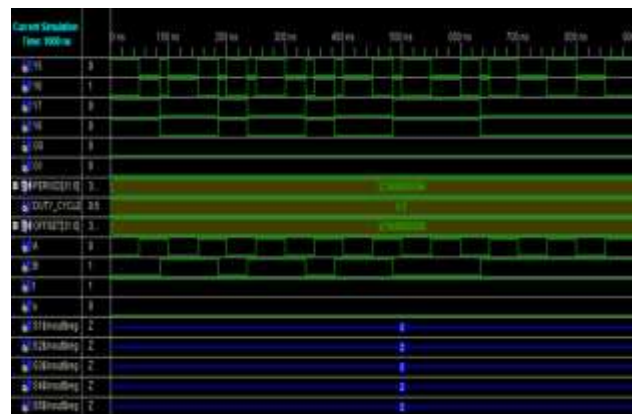
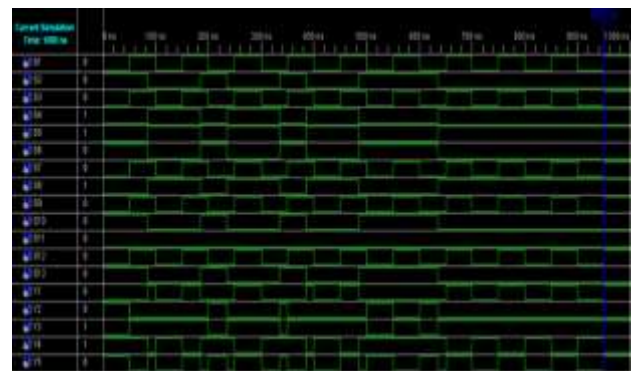
VI. SIMULATION RESULTS

This work also includes the simulation of 8:1 multiplexer, multi logic function generator which generates 8 Boolean functions, controlled multi logic function generator using 8:1 multiplexer and ALU using this Reversible Logic gates. The simulated snapshot output waveforms of the proposed circuits are shown below. The simulation has been done by XILINX ISE 10.1 tool.

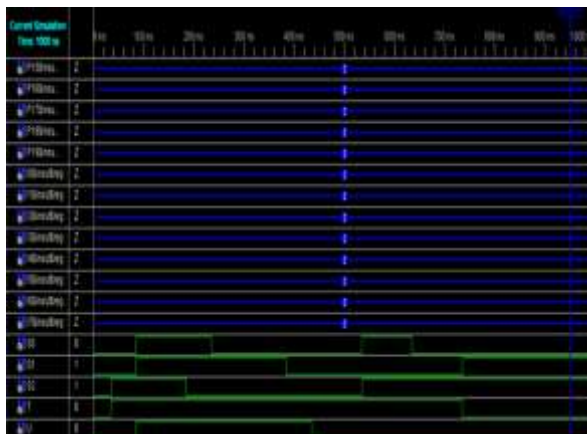
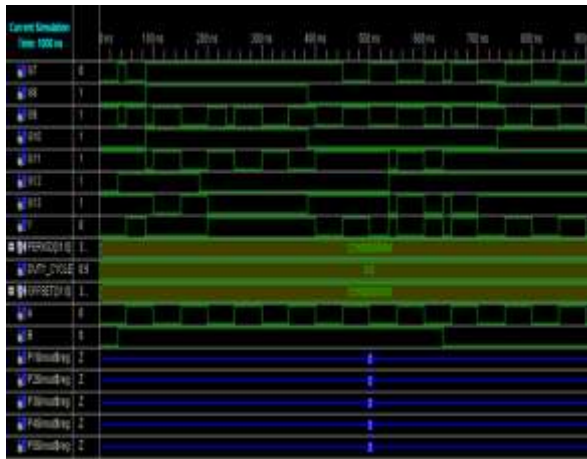
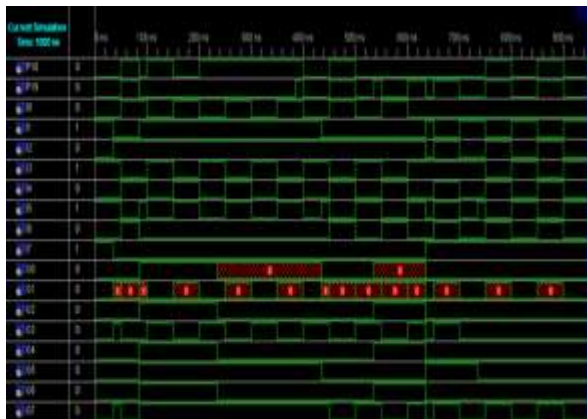
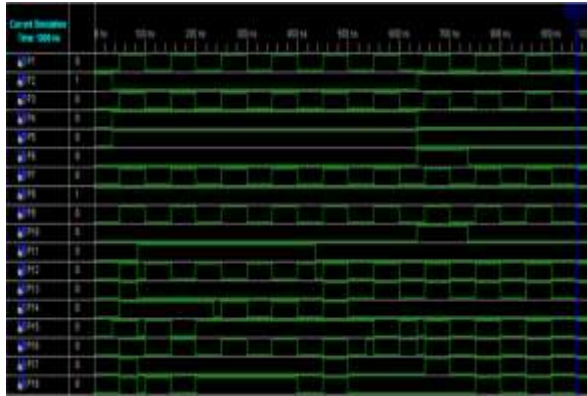
A. Simulation result 8:1 Multiplexer



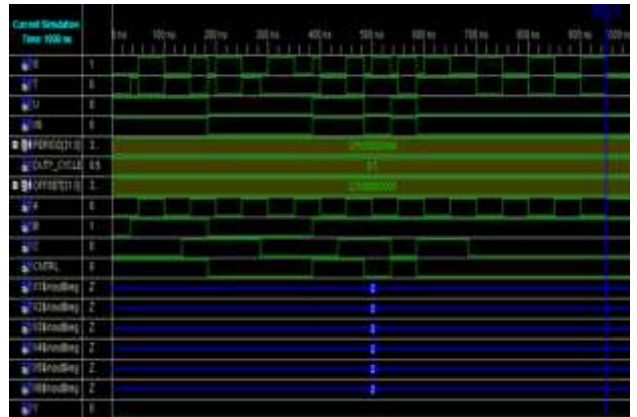
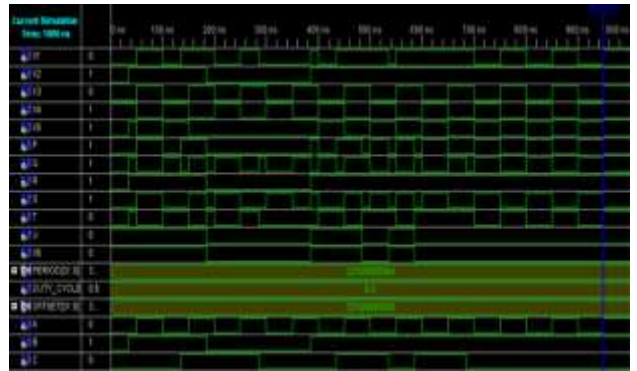
B. Simulation result for Multi Logic Function Generator



C. Simulation result for controlled multi logic function generator using 8:1 multiplexer



D. Simulation result for ALU



VII. COMPARISONS

Name of the circuit	Gate count	Garbage output	Constant Input	Hardware Complexity
Irreversible 8:1 MUX	6	10	4	$11\alpha + 10\beta + 2\delta$
Reversible 8:1 MUX	3	6	0	$6\alpha + 8\beta + 3\delta$
$2^n:1$	$2^n - 1$	$2^n + n - 1$	-	$(2^n - 1)(2\alpha + 2\beta + 2\delta)$

Parameters to be compared	Function Generator of 8 Functions	
	Without Controlled Circuit	With Controlled Circuit
Gate count	15	8
Constant input	7	17
Garbage output	12	2
Hardware complexity	$29\alpha + 28\beta + 28\delta$	$15\alpha + 14\beta + 14\delta$

Parameters to be Compared	Basic 1bit ALU	Proposed 1 bit ALU
Gate Count	11	8
Quantum Cost	31	24
Garbage Output	2	1
Constant Input	1	2

VIII. CONCLUSION

In this paper a reversible multiplexer using cog gate proposed and described. One of the major constraints in reversible logic is to minimize the number of reversible gates used, garbage outputs produced and usage of number of constant inputs. A comparison is made between irreversible and reversible multiplexers and function generator and ALU in terms of gate count, garbage output and hardware complexity, constant inputs.

In this paper, we have also proposed a novel design of an ALU . Thus, it can be clearly seen that there is a considerable improvement in the quantum cost and gate count compared to the previous works

X. REFERENCES

- [1] R. Landauer, "Irreversibility and heat generation in the computational process," IBM Journal of Research and Development, vol. 5, pp. 183-191, 1961.
- [2] C. H. Bennett, "Logical reversibility of computation," IBM J. Research and Development, vol. 17, no. 6, pp. 525-532, Nov. 1973.
- [3] P. Singla and N. K. Malik, "A cost effective design of reversible programmable logic array," IJCA, vol. 41, no. 15, Mar. 2012.
- [4] S. Younis and T. Knight, "Asymptotically zero energy split level charge recovery logic," in Proc. Workshop on Low Power Design, Jun. 1994.
- [5] M. Perkowski and P. Kerntopf, "Reversible logic. Invited tutorial," in Proc. EURO-MICRO, Warsaw, Poland, Sep. 2001.
- [6] R. Saligram and T. R. Rakshith, "Novel code converter employing reversible logic," IJCA, vol. 52, no. 18, Aug. 2012.
- [7] M. S. Islam, et al., "Synthesis of fault tolerant reversible logic," in Proc. IEEE Circuits and Systems International Conference on Testing and Diagnosis, 2009.
- [8] R. Feynman, "Quantum mechanical computers," Optical News, vol. 11, no. 2, pp. 11-20, 1985.
- [9] B. Parhami, "Fault tolerant reversible circuits," in Proc. 40th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Oct. 2006.
- [10] E. Fredkin and T. Toffoli, "Conservative logic," International Journal of Theoretical Physics, vol. 21, pp. 219-253, 1982.

cellc

I